

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Cognitive analyses of machine learning systems

Permalink

<https://escholarship.org/uc/item/09r411mq>

Author

Grant, Erin Marie

Publication Date

2022

Peer reviewed|Thesis/dissertation

Cognitive analyses of machine learning systems

by

Erin Marie Grant

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael I. Jordan, Chair

Professor Thomas L. Griffiths

Professor Anca D. Dragan

Professor Steven T. Piantadosi

Summer 2022

Cognitive analyses of machine learning systems

Copyright 2022
by
Erin Marie Grant

Abstract

Cognitive analyses of machine learning systems

by

Erin Marie Grant

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Michael I. Jordan, Chair

Machine learning systems are increasingly a part of human lives, and so it is increasingly important to understand the similarities and differences between human intelligence and machine intelligence. However, as machine learning systems are applied to more complex problem settings, understanding them becomes more challenging, and their performance, correctness, and reliability become increasingly difficult to guarantee. Moreover, "human-level performance" in such settings is often itself not well-defined, as many of the cognitive mechanisms underlying human behavior remain opaque. This dissertation bridges gaps in our understanding of human and machine intelligence using cross-disciplinary insights from cognitive science and machine learning.

First, I develop two frameworks that borrow methodologically from cognitive science to identify deviations in the expected behavior of machine learning systems. Second, I forge a connection between a classical approach to building computational models of human cognition, hierarchical modeling, and a recent technique for small-sample learning in machine learning, meta-learning. I use this connection to develop algorithmic improvements to machine learning systems on established benchmarks and in new settings that highlight their inability to come close to human standards. Finally, I argue that machine learning should borrow methodologically from cognitive science, as both are now tasked with studying opaque learning and decision-making systems. I use this perspective to construct a computational model of machine learning systems that allows us to formalize and test hypotheses about how these systems operate.

To my mother.

Contents

Contents	ii
List of Figures	v
List of Tables	xi
1 Introduction	1
I Behavioral studies of neural networks	8
2 Theory of mind and the <i>false belief</i> task	9
2.1 Introduction	10
2.2 Theory of mind and the <i>false-belief</i> task	11
2.3 Memory networks	12
2.4 Simulation 1: MemN2N model	13
2.5 Simulation 2: Multiple-observer model	18
2.6 Conclusions	20
3 Rule- and exemplar-based generalization	21
3.1 Introduction	22
3.2 Inductive biases in category learning	23
3.3 A protocol for measuring inductive bias	25
3.4 2-D classification example	28
3.5 IMDb text classification	31
3.6 CelebA image classification	32
3.7 Related work and future directions	34
3.8 Conclusions	35
II Hierarchical modeling	36
4 Recasting meta-learning as hierarchical Bayes	37

4.1	Introduction	38
4.2	Meta-learning formulation	38
4.3	Linking gradient-based meta-learning & hierarchical Bayes	41
4.4	Improving model-agnostic meta-learning	44
4.5	Experimental evaluation	47
4.6	Related work	49
4.7	Conclusion	50
5	Concept learning from few positive examples	51
5.1	Introduction	52
5.2	Background	53
5.3	Modeling approach	55
5.4	Behavioral experiment	57
5.5	Meta-learning simulations	58
5.6	Discussion	62
6	Nonparametric priors for non-stationarity	63
6.1	Introduction	64
6.2	Gradient-based meta-learning as hierarchical Bayes	65
6.3	Improving meta-learning by modeling latent task structure	65
6.4	Experiment: <i>miniImageNet</i> few-shot classification	68
6.5	Scalable online mixtures for task-agnostic continual learning	69
6.6	Experiments: <i>Task-agnostic</i> continual few-shot regression & classification	70
6.7	Related work	75
6.8	Conclusion	75
III	Computational modeling of neural networks	76
7	Gaussian process surrogate models	77
7.1	Introduction	78
7.2	Background	79
7.3	Learning a Gaussian process surrogate model from neural network predictions	81
7.4	Experiments	85
7.5	Discussion	95
IV		96
8	Conclusion	97
	Bibliography	100

A	Mathematical derivations	118
A.1	Recasting meta-learning as hierarchical Bayes	118
B	Additional experimental results	124
B.1	Nonparametric priors for non-stationarity	124
B.2	Rule- and exemplar-based generalization	132
B.3	Gaussian process surrogate models	140

List of Figures

2.1	An example task from the bAbi dataset.	12
2.2	Examples of the training data, with the predicates of interest underlined. Note that the <i>true-belief</i> (TB) and <i>false-belief</i> (FB) test tasks are of the same form as the top and bottom items, respectively, in the last column.	13
2.3	Accuracy in Simulation 1. Test accuracies for the <i>true-belief</i> (TB) and <i>false-belief</i> (FB) tests across training conditions in Simulation 1. We report results for $p(\text{false belief}) = 0.5$, since varying this parameter did not affect results except in the few cases discussed in the text.	16
2.4	Accuracy in Simulation 2. Test accuracies for the <i>true-belief</i> (TB) and <i>false-belief</i> (FB) tests across training conditions in Simulation 2. As in Fig. 2.3, we report results only for $p(\text{false belief}) = 0.5$	16
2.5	From Simulation 1. The test accuracy in the AB condition is dependent on the value of $p(\text{false belief})$, but not in the BA condition.	18
2.6	Attention in Simulation 2. Visualisation of the attention weighting over memory caches for the <i>true-belief</i> (TB) and <i>false-belief</i> (FB) tests. We omit the visualization for the BA+AB and BA+AB+A(B)A training conditions because the test accuracy distribution in Simulation 2 for these conditions is very similar to the A(B)A training condition (see Fig. 2.4).	19
3.1	Example of a data condition: Data often underdetermines a decision boundary; here, it is unclear whether shape or color determines object label (“dax” vs “fep”). How a learner extrapolates to new stimuli reveals inductive bias. . . .	22
3.2	Illustrative category learning experiment: Training examples from the 3 independent training conditions, the extrapolation test, and characteristic behavior for learners with different inductive biases. We formalize the training conditions in Fig. 3.3.	24
3.3	Formalizing the illustrative experiment: The experiment from Fig. 3.2 expressed in terms of the formalism in Section 3.3 with color as \mathbf{z}_{dist} and shape as \mathbf{z}_{disc} . Background colors indicate the true category.	26
3.4	Spurious correlation (Eq. (3.3)).	27
3.5	Simple 2-D classification (Section 3.4) The specific model used in (a) are bolded in (b).	29

3.6	Example stimuli from the IMDb dataset.	31
3.7	CelebA results. Stimuli and results on various feature pairings from the CelebA domain (Section 3.6). Error bars represent 95% confidence intervals across ResNets of various sizes. See figure sub-captions and main text for details.	32
4.1	(Left) The computational graph of the model-agnostic meta-learning (MAML) algorithm covered in Section 4.2. Straight arrows denote deterministic computations and crooked arrows denote sampling operations. (Right) The probabilistic graphical model for which MAML provides a parameter estimation procedure as described in Section 4.3. In each figure, plates denote repeated computations (left) or factorization (right) across independent and identically distributed samples.	40
4.2	Model-agnostic meta-learning as hierarchical Bayesian inference. The choices of the subroutine <code>ML-...</code> that we consider are defined in Subroutine 4.3 and Subroutine 4.4.	41
4.3	Subroutine for computing a point estimate $\hat{\phi}$ using truncated gradient descent to approximate the marginal negative log likelihood (NLL).	42
4.4	Subroutine for computing a Laplace approximation of the marginal likelihood.	45
4.5	Our method is able to meta-learn a model that can quickly adapt to sinusoids with varying phases and amplitudes, and the interpretation of the method as hierarchical Bayes makes it practical to directly sample models from the posterior. In this figure, we illustrate various samples from the posterior of a model that is meta-trained on different sinusoids, when presented with a few datapoints (in red) from a new, previously unseen sinusoid. Note that the random samples from the posterior predictive describe a distribution of functions that are all sinusoidal and that there is increased uncertainty when the datapoints are less informative (<i>i.e.</i> , when the datapoints are sampled only from the lower part of the range input, shown in the bottom-right example).	47
5.1	The <i>word learning</i> paradigm from Xu and Tenenbaum (2007). In each trial, participants see a few instances exemplifying a novel word such as “dax” and are asked to select other instances that fall under the same word from a test array. The training conditions vary by the levels of the underlying image taxonomy from which the instances are drawn, <i>e.g.</i> , Dalmatians (subordinate) vs. dogs (basic) vs. animals (superordinate).	54
5.2	Examples of training stimuli for the (a) subordinate, (b) basic-level, and (c) superordinate level training conditions, as well as (d) a subset of the stimuli from the test array for a specific concept learning task (here, learning the concept <i>black currant</i> (a), <i>currant</i> (b) or <i>fruit</i> (c)). The test array (d) displays, from left to right, a subordinate match, a basic-level match and a superordinate match.	59

5.3	Human behavioral data (a), <code>flat</code> (b), and <code>hier</code> (c) modeling results on the concept generalization task. The horizontal axis identifies the training condition (<i>i.e.</i> , the level of taxonomic abstraction from which the few-shot examples are drawn). The vertical axis identifies, for each type of match in {subordinate, basic-level, superordinate}, the proportion of selections from the test array (a), or the average probability of generalization (b, c).	60
6.1	Algorithm 6.1: Stochastic gradient-based expectation maximization (EM) for probabilistic clustering of task-specific parameters in a meta-learning setting.	66
6.2	Subroutine 6.2: The E-STEP and M-STEP for a finite mixture of hierarchical Bayesian models implemented as gradient-based meta-learners.	66
6.3	The E-STEP and M-STEP for an infinite mixture of hierarchical Bayesian models.	69
6.4	The diverse set of periodic functions used for few-shot regression in Section 6.6.	71
6.5	Artistic filters (b-d) applied to <code>miniImageNet</code> (a) to ensure non-homogeneity of tasks in Section 6.6.	71
6.6	Results on the evolving dataset of few-shot regression tasks (lower is better). Each panel (row) presents, for a specific task type (polynomial, sinusoid or sawtooth), the average meta-test set accuracy of each method over cumulative number of few-shot episodes. We additionally report the degree of loss in backward transfer (<i>i.e.</i> , catastrophic forgetting) to the tasks in each meta-test set in the legend; all methods but the nonparametric method experience a large degree of catastrophic forgetting during an inactive phase.	72
6.7	Task-specific per-cluster meta-test responsibilities $\gamma^{(\ell)}$ for both active and unspawned clusters. Higher responsibility implies greater specialization of a particular cluster (color) to a particular task distribution (row).	72
6.8	Results on the evolving dataset of filtered <code>miniImageNet</code> few-shot classification tasks (higher is better). Each panel (row) presents, for a specific task type (filter), the average meta-test set accuracy over cumulative number of few-shot episodes. We additionally report the degree of loss in backward transfer (catastrophic forgetting, CF) in the legend. This is calculated for each method as the average drop in accuracy on the first two tasks at the end of training (lower is better; U.B.: upper bound).	73
6.9	Task-specific per-cluster meta-test responsibilities $\gamma^{(\ell)}$ for both active and unspawned clusters. Higher responsibility implies greater specialization of a particular cluster (color) to a particular task distribution (row).	74

7.1	Outline of the surrogate modeling approach. We learn a Gaussian process surrogate model for a neural network family applied to a task family by learning kernel hyperparameters from aggregated neural network predictions across datasets. We interpret the learned kernel to derive insights into the properties of the neural network family; for example, biases towards particular frequencies (see Section 7.4), or expected generalization behavior on a new dataset (see Section 7.4).	79
7.2	Training and evaluation of the Gaussian process (GP) surrogate model described in Section 7.3.	82
7.3	Demonstration: Comparing learned GP priors with neural network (NN) priors. Samples from GP prior (right) with kernel hyperparameters inferred from the predictions of NN families (left). GPs are flexible enough to capture properties of each NN family; for example, the samples from the learned GP prior reflect the quickly varying behavior of the 32-layer sinusoidal NNs and the increasing-decreasing behavior of rectifier NNs.	83
7.4	Capturing spectral bias in neural networks. (Top) Neural network predictions as training progresses on the sum-of-sines target function described in Section 7.4. (Bottom) Spectral mixture kernel fit to neural network predictions as training progresses. The kernel reveals a spectral bias for this neural network family, with the range of spectral frequencies expressed in the kernel increasing with the number of iterations of training.	85
7.5	Depth pathologies in randomly initialized neural networks. Predictions of neural networks (left) from neural network families of different activations (rows) and varying depths (columns); mean and standard error of the covariance of the corresponding surrogate model kernels (right). The covariance is aggregated across 10 kernels learned from 10 different 50-member neural network ensembles from a given family. Greater depth results in kernels with shorter lengthscales, with this pathology emerging earlier in rectifier neural networks (rectifier NNs); this result is consistent with prior work on pathologies of deep neural networks.	87
7.6	Ranking generalization from MLL in large-width NNs. Mean and standard error of the test MSE of large-width sinusoidal and erf NNs trained with learning rates $\eta = 0.01$ (left) and $\eta = 0.1$ (right) on the target function of Section 7.4. The MLL of the target function under the surrogate model corresponding to the limiting kernel for each model family is shown in the legend. Consistent with expectations, the model family whose surrogate assigns higher MLL to the target function achieves lower test error for both values of η .	89

- 7.7 Ranking generalization from MLL in small-width NNs. Mean and standard error of test MSE (left) of small-width sinusoidal and rectifier NN ensembles on sine (sin) (top) and rectified linear unit (ReLU) (bottom) target function families, with the target function MLL under the surrogate learned from each model family in the legend. Covariance (right) of surrogate kernels alongside data kernels learned from the sin (top) and ReLU (bottom) target function families. Even in the small-width regime and when the kernel is learned, the model family whose surrogate assigns a higher MLL to the target function attains lower error (left); the surrogate kernel learned from the better-performing model family better matches the data kernel (right). 90
- 7.8 Ranking generalization performance from MLL across different learning algorithms and architectures. Each panel displays mean and standard error of test MSE of an NN family trained on the target function $\sin(0.5x)$ with noise; legend displays MLL of the training data under the surrogate for one of two NN families: 1-layer (256 hidden units) sinusoidal or rectifier NNs (top)); 3-layer (256 hidden units) sinusoidal or rectifier NNs (bottom). NNs are trained with batch gradient descent with Adam (learning rates $\eta = 0.0003$, $\eta = 0.0003$) or vanilla batch gradient descent ($\eta = 0.01$). Across architectures and learning algorithms, the NN family whose surrogate assigns higher MLL to the target function achieves lower test error. 91
- 7.9 Qualitative connection between lengthscale profile discrepancy and generalization gap. Each subfigure compares normalized lengthscales learned from neural network predictions on validation set (*i.e.*, surrogate lengthscales) after training and normalized lengthscales learned from training data (*i.e.*, data lengthscales). A lengthscale greater than 1 indicates an “unimportant” feature. The title indicates the UCI dataset and generalization gap defined in Fig. 7.10. Data and surrogate lengthscales for some features are different (*e.g.*, features 1, 4, 6), reflected in a high generalization gap (left). Data and surrogate lengthscales for the same features are generally similar, reflected in a low generalization gap (right). This suggests a connection between the generalization gap and discrepancy between surrogate and data lengthscales. 92
- 7.10 Inverse relationship between generalization error and lengthscale correlation on UCI datasets. Each point represents the lengthscale correlation (between surrogate and data lengthscales) and the generalization gap for a neural network ensemble to which the surrogate model is fit, on a single UCI dataset. Each panel corresponds to a particular neural family; see Section 7.4 for details about hyperparameters of these families, including architectures. Across datasets and architectures, a larger lengthscale correlation (*i.e.*, higher similarity between the data and surrogate representations) corresponds to a lower generalization gap (*i.e.*, better extrapolation). 93

B.1	We expand on Fig. 3.3 from the main text by including a realization of the abstract training conditions in the simple 2D points-in-a-plane setting. (Top) Formalizing the illustrative experiment: The experiment from Fig. 3.2 expressed in terms of the formalism in Section 3.3 with $\mathbf{z}_{\text{dist}} = \text{color}$ and $\mathbf{z}_{\text{disc}} = \text{shape}$. Background colors indicate true category boundary. (Bottom) The conditions realized via a binarization of continuous feature values. Here, the discriminant is binarized as $x_1 > 0$ and the distractor as $x_2 > 0$; this setting is further investigated in Section 3.4. Color here depicts the label but is not part of the input.	133
B.2	CelebA exemplar vs. rule propensity (EvR) and feature-level bias (FLB) across feature pairs, averaged across 30 runs, split by depth and width of ResNet. . .	134
B.3	We visualize several of the interpolants used for the interpolation analyses. . .	136
B.4	Interpolations away from the PE: changes in extrapolation behavior under data distribution with the same spurious correlation as in PE, as well as different ways to change spurious correlation.	137
B.5	Illustrating the effect of GP kernel hyperparameters on the GP prior. (Left) Samples from a GP prior with spectral mixture kernel (SMK) with varying mixture weights ω , mixture scale τ , and mixture means μ . (Right) Samples from a GP prior with Matern kernel with varying ν and ℓ (lengthscale). GPs are flexible models whose properties can be controlled through hyperparameters.	140
B.6	Sensitivity analysis of generalization gap and lengthscale profile relationship. Each panel a histogram and mean (red line) of correlations obtained by recomputing the correlation between lengthscale profile correlation and generalization gap after removing each UCI dataset. Across datasets and architectures, even when a single dataset is removed, there remains a negative correlation between generalization gap and lengthscale profile correlation. Therefore, the inverse relationship between generalization gap and lengthscale profile correlation demonstrated in Fig. 7.10 is robust to outlier datasets.	141

List of Tables

4.1	One-shot classification performance on the <i>miniImageNet</i> test set, with comparison methods ordered by one-shot performance. All results are averaged over 600 test episodes, and we report 95% confidence intervals.	48
5.1	The eight taxonomies of Rosch et al. (1976) used to define ImageNet concepts that our images are sampled from.	56
6.1	Meta-test set accuracy on the <i>miniImageNet</i> 5-way, 1- and 5-shot classification benchmarks from Vinyals et al. (2016) among methods using a comparable architecture (the 4-layer convolutional network from Vinyals et al. (2016)). For methods on which we report results in later experiments, we additionally report the total number of parameters optimized by the meta-learning algorithm. . .	68

Acknowledgments

This thesis would not have been possible without the support of my advisor, Tom Griffiths. I am grateful to have learned from his ability to formulate clear and deeply interesting research questions, and I admire that he genuinely cares about the happiness and success of his students. Thank you to my dissertation and qualifying exam committee members, Anca Dragan, Steve Piantadosi, Sergey Levine, and especially Mike Jordan, for serving as chair of my committee and for supporting me at Berkeley as Tom's lab moved to Princeton. Thank you to all my collaborators, from whom I have learned innumerable lessons, in the process of working on the content of this thesis and beyond.

To the current and former students and postdocs with whom I've had the good fortune to overlap at Berkeley and Princeton—Devin Guillory, Seth Park, Allan Jabri, Thanard Kurutach, Dibya Ghosh, Ashvin Nair, Young Geng, Kelvin Xu, Aviral Kumar, Anusha Nagabandi, Kate Rakelly, Parsa Mahmoudieh, Michael Li, Smitha Milli, Ishita Dasgupta, Lisa Hendricks, Kaylee Burns, Chelsea Finn, Devin Guillory, Rudy Corona, Ignasi Clavera, Maria Eckstein, Carlos Florensa, Rachit Dubey, Mayank Agrawal, Evan Shelhamer, Deepak Pathak, Sayna Ebrahimi, Igor Mordatch, Rowan McAllister, Michael Janner, Sreejan Kumar, Daniel Fried, Ilia Sucholutsky, Qiong Zhang, and many others—I look forward to seeing you continue to do great things. I would especially like to thank Michael Chang, for the value of deep dives; Abhishek Gupta, for resistance to the commoditization of academic mentorship; and Brian Cheung, for finding the humor hidden in everything. I am also grateful to my first-year cohort, Justin Fu, Avi Singh, Vitthay Pong, JD Co-Reyes, Dequan Wang, Harry Xu, and Marvin Zhang, for fun escapades near and far. Thank you to Jean Nguyen, Shirley Salanio, and Angie Abbatecola for creating order amid chaos, and to Devin Guillory, Nathan Lambert, Giulia Guidi, Angjoo Kanazawa, and all who have helped two inclusion programs at Berkeley—the BAIR REU and EAAA—live on and help others.

To my roommates in Berkeley and San Francisco—Ludwig Schubert, Nikhil Mishra, Ignasi Clavera, Tenzin Kunsel, Mostafa Rohaninejad, Esther Rolf, Victoria Cheng, and Carolyn Chen—thanks for making home *home*. I would especially like to thank Katherine Lee for demos of the value of effort in all things, and Caroline Lemieux and Abhishek Gupta, for wisely answering endless anxious questions about my recent job search. Thank you to colleagues and friends from faraway places—Utku Evci, Laura Graesser, Cyril Zhang, Samira Abnar, Eleni Triantafillou, and many more—for enriching my life. Thank you to Vincent Dumoulin, Hugo Larochelle, Hanie Sedghi, Yan Wu, and Tim Lillicrap for hosting me for internships and for giving me the excuse to spend summers and falls in la belle ville de Montréal. And, looking all the way back, thank you to Aida Nematzadeh and Suzanne Stevenson, who made all the difference in my becoming a scientist at all.

Merci à Laurent, for sharing this time with me. I admire your integrity and your compassion, and I have learned so much from you during our time together. Thank you to my family, for encouraging me to go into unfamiliar territory, and especially to my mother, who has supported and will support me endlessly. I dedicate this thesis to her.

Chapter 1

Introduction

Machine learning has the potential to revolutionize scientific and application domains alike. However, current machine learning systems—primarily built of deep neural networks (LeCun, Bengio, et al. 2015)—are unpredictable and uninterpretable, posing challenges for designing reliable systems and deriving robust scientific insights (D’Amour et al. 2020; Kapoor and Narayanan 2022). The cause for this dilemma is that we lack *design principles* to provide guarantees on the relevant behavior of machine learning systems. Though progress has been made in theory and practice to identify and ameliorate cases of unexpected behaviors, this progress in many cases relies on simplifying assumptions—such as narrowing the model class, or assuming a particular structure of data—and as such cannot provide prescriptions in naturalistic settings. Indeed, the downstream consequences of a change in design—for example, tweaking an architecture or changing the value of a hyperparameter—are often not known prior to the deployment of the system itself.

Where do we go from here? We must first contend with the fact that a single set of principles governing machine learning systems is unlikely to exist—indeed, the result of a data-dependent, iterative optimization procedure like those used to train neural networks is unlikely to be describable in a simple mathematical form, and explanatory gaps have already been identified for promising frameworks (Arora et al. 2019; Razin and Cohen 2020; Nagarajan and Kolter 2019; Dziugaite et al. 2020). However, this does not leave us at a loss—instead of despairing of the lack of a single unified theory, we can find hope in the multiplicity of ways in which we could study machine learning systems in order to make these systems more reliable wherever they are used.

In this dissertation, I take that opportunity and revisit a discipline classically connected to artificial intelligence and machine learning—the study of human cognition. The earliest visions of artificial intelligence were motivated as a way to recreate human capacities for thinking and reasoning (Turing 1950), and the precursors of modern deep neural networks, connectionist models, were motivated as explicit models of cognitive processes (Hinton and Anderson 1981; Rumelhart and McClelland 1986; Rumelhart, McClelland, Group, et al. 1986). Even the image classification benchmark that can be seen as the catalyst to this era of explosive interest in deep learning, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC, Deng et al. 2009), is underlain by a conceptual organization of the *human* lexicon (Miller 1995; Fellbaum 1998), and so is fundamentally tied to human cognition. Within the disciplinary connections between cognitive science and machine learning, I focus on two aspects that are particularly relevant in understanding the present era of large-scale machine learning systems: the question of *what* should be studied of these systems and the question of *how* to go about doing so. Before doing so, in the next section, I introduce the notion of *inductive bias*, which allows us to make reference to mechanisms governing the behavior of machine learning systems even if we may not be able to exactly and precisely describe how they operate. In addition, in a later section, I introduce hierarchical models, which allow us to formalize the *acquisition* or *automatic discovery* of inductive bias.

Inductive bias

As machine learning systems are applied beyond idealized research environments and benchmarks, where ground-truth datasets are easy to gather and the correctness of a behavior is easy to evaluate, to high-stakes decision-making in medicine, finance, and policing, failure cases become more evident. The fundamental link between many of these failure cases is the *problem of underdetermination*—that is, the insufficiency of data alone to provide evidence for any particular solution. In the simplest terms, there are, in almost all problems of interest in machine learning, a multitude of mechanisms or decision rules that a learner could posit that are consistent with observed data, but nonetheless produce different behaviors on unseen data.

The underdetermination problem has been fundamental to the understanding of intelligence as long as intelligence has been studied: Plato (editor, 1976) wondered how an uneducated boy could be taught geometric principles. Russell (1948) wondered: “How comes it that human beings, whose contacts with the world are brief and personal and limited, are nevertheless able to know as much as they do know?” Chomsky (1980) considers the “poverty of the stimulus” in the acquisition of grammatical structure and lexical items by an abstract learner. Mitchell (1980) was the first to call out underdetermination in the context of machine learning by highlighting the inability of a statistical learner to generalize without additional information—termed *inductive bias*—that guides decisions on unseen data. Inductive bias also includes design choices that *implicitly* affect extrapolation behavior without the explicit intent of the system designer. This setting more closely characterizes the situation of modern machine learning systems, for which any and all components could give rise to implicit effects on extrapolation behavior. Indeed, many components have been shown to produce systematic effects on generalization, including model architecture (Golubeva et al. 2020), parameter initialization procedures (Mehta et al. 2020), gradient-based optimization (Smith, Dherin, et al. 2021), and overparametrization (Neyshabur, Li, et al. 2019).

Inductive bias is thus a catch-all term for everything that is consequential for a learner’s extrapolation behavior—behavior on unseen data—including explicit constraints, regularizers, or prior distributions in the case of Bayesian models, and this notion allows us to speak concretely about what it is that makes learning systems behave differently in extrapolation regimes, even though they may make the same decisions on observed data. This notion will be foundational for the cognitive analyses developed in this dissertation.

Phenomena in machine learning systems: *What should we analyze?*

Until now, this description has assumed that the relevant extrapolation behavior is clear. While in some settings, it is obvious what notions of extrapolation to guarantee—for example, studies of *robustness* often make specific distributional assumptions to test (Hendrycks, Basart, et al. 2021)—in general, there is no way to determine *a priori* what extrapolation behavior is relevant for a scientific or applied machine learning use case. Indeed, though

there has been no end of interest in out-of-distribution generalization, there can be no notion of what it means to be out-of-distribution or extrapolation without assumptions (Le Lan and Dinh 2021; Ye et al. 2021; Adebayo et al. 2022).

How do we identify salient extrapolation behaviors of a given system? Many tasks to which we apply machine learning systems are tasks that humans solve or that involve interaction with humans. Consequently, in many cases, we would like to understand deviations of these systems *from human-like behavior*. The first part of this dissertation, Part I, contributes behavioral evaluation protocols that investigate such deviations. The key contribution in these sections is to identify a hallmark of how a system solves a given problem that is more nuanced than a summary statistic such as accuracy by appealing to a human standard in which particular modes of behavior have already been characterized. In both case studies, comparing the behavior of machine learning systems to these standards requires work on the side of evaluation—that is, conceptualizing a phenomenon of interest, and defining a protocol to measure it.

Chapter 2 studies *theory of mind* in a language model termed a “memory network” that can solve reasoning tasks. We propose a new set of tasks to demonstrate that question-answering models fail in social reasoning scenarios. Humans find these tasks easy, and cognitive scientists have attested that this is because humans possess an inductive bias known as “theory of mind,” the propensity to model the latent mental states of other agents—for example, their desires, beliefs, or intentions. In this chapter, we introduce a model design that explicitly simulates the mental representations of different participants in a narrative reasoning task, demonstrating that this simulated component is sufficient for good performance.

Chapter 3 relates the memorization-systematicity trade-off in neural networks to a classical perspective on category learning models, the trade-off between *rule- and exemplar-based generalization*. A characteristic of neural networks that reliably deviates from human learning is their tendency to memorize datapoints instead of learning systematic predictors (Lake and Baroni 2018a). This chapter demonstrates such failures of systematicity in a study of combinatorial generalization using formulations of rule- and exemplar-based generalization appropriate for general category learning models, including neural networks. This formulation reveals that, contrary to the simple maxim of “more data is better,” exposure to new feature values can *worsen* systematic generalization to new combinations involving these features. This finding contradicts the common intuition that more coverage in training data necessarily leads to better generalization performance, and has implications for the incautious use of neural networks in data settings that have this partial exposure structure.

Hierarchical models

As described above, Part I of this dissertation introduces behavioral protocols to examine specific extrapolation behaviors in machine learning systems. In Part II, I introduce the framework of hierarchical models in order to be able to reason about inductive bias by

making reference to explicit inductive bias in the form of a parameterized prior distribution. We will see in Part III of this dissertation that this allows us to respond to a problem left open in the previous section—rather than investigating specific inductive biases in machine learning models by constructing independent behavioral protocols, is there a way to more systematically investigate them, and perhaps even *discover them from data*?

The acquisition of abstract knowledge can be formalized via a *hierarchical* or *multilevel* model. Hierarchical models exist in a variety of frameworks, but *hierarchical Bayesian models* bring the benefits of working within the framework of probabilistic models, including an explicit declaration of the assumed prior knowledge in the form of the prior distribution and the distributional parameterizations. To formalize how people acquire domain-general, abstract knowledge, hierarchical Bayesian models can describe how individual learning experiences can be consolidated into domain-general knowledge in the process of learning to solve a variety of individual problems. Machine learning systems enact an analogous capability—*meta-learning*—as the solution to a two-stage learning problem in which a system learns to solve a set of independent tasks from the limited experience available for each task, then consolidates this experience in the form of general principles for learning to solve any new task (Thrun and Pratt 1998). Meta-learning can be implemented in a framework compatible with modern pattern recognition systems as the estimation of a common parameter initialization of a set of models that are independently adapted to different tasks (Vinyals et al. 2016; Ravi and Larochelle 2017; Finn, Abbeel, et al. 2017). The key idea is that the initialization shared among task-specific models serves as a useful domain-general bias for solving the kinds of tasks that each learner may be faced with, thereby increasing learning efficiency to be closer to the level of a human learner rather than a learner with no prior knowledge.

Despite the algorithmic similarities between these concepts—hierarchical Bayesian modeling in cognitive science and meta-learning in machine learning—it was not known how to relate two methodologies at a level that would be prescriptively useful to either machine learning or cognitive science. Chapter 4 provides this connection by showing how a formalism for expressing the solution to inference problems about domain-general knowledge—posed in the language of hierarchical Bayesian models—as a multi-level optimization problem—which can be solved with the tools from meta-learning in machine learning. A key feature of the meta-learning approach is that it provides a framework to investigate constraints on learning simply on the basis of the observable outcomes of learning; these constraints can be interpreted via tailored analysis datasets.

Chapter 5 solidifies the connection between meta-learning and hierarchical Bayes by revisiting a classical study in computational models of cognition—how taxonomic structure influences concept generalization. Human concept learning is surprisingly robust, allowing for precise generalizations given only a few positive examples. Bayesian formulations that account for this behavior require elaborate, pre-specified priors, leaving much of the learning process unexplained. More recent models of concept learning bootstrap from deep representations, but the deep neural networks are themselves trained using millions of positive and negative examples. In machine learning, recent progress in

meta-learning has provided large-scale learning algorithms that can learn new concepts from a few examples, but these approaches still assume access to implicit negative evidence. In this chapter, we formulate a training paradigm that allows a meta-learning algorithm to solve a problem of concept learning from few positive examples. The algorithm discovers a taxonomic prior useful for learning novel concepts even from held-out supercategories and mimics human generalization behavior—the first to do so without hand-specified domain knowledge or negative examples of a novel concept.

Chapter 6 uses the connection between meta-learning and hierarchical Bayes to implement the *latent cause inference* model of Gershman, Blei, and Niv (2010). This probabilistic meta-learner explicitly modulates the amount of transfer between tasks, as well as to adapt its parameter dimensionality when the underlying task distribution evolves. We formulate this as probabilistic inference in a mixture model that defines a clustering of task-specific parameters. and the connection between gradient-based meta-learning and hierarchical Bayes (Grant et al. 2018) from Part II allows scalable approximate *maximum a posteriori* (MAP) inference in both a finite and an infinite mixture model. This chapter is a first step towards more realistic settings of diverse task distributions, and crucially, *task-agnostic* continual learning, and is also an example of a model that leverages insights from cognitive science in order to propose improvements in the learning efficiency of machine learning systems.

Explanations of machine learning systems: *How* should we analyze them?

Prior work on implicit inductive biases in modern machine learning systems focuses on the effect of well-understood constraints, such as the use of architectures that impose invariances to particular properties of data or on regularizers that confer a simplicity bias (e.g., Golubeva et al. 2020; Mehta et al. 2020; Smith, Dherin, et al. 2021; Neyshabur, Li, et al. 2019). However, it is difficult to analyze the interaction of all the design decisions that make up a machine learning system, some of which may or may not contribute to inductive bias specification. Moreover, the behavioral approach I advocated for in Part I allows us to probe the existence or degree of specific inductive biases, but does not allow us to investigate inductive biases in all generality.

In the last part of this dissertation, I make use of the connection between hierarchical models and meta-learning to argue for a *computational modeling* approach to the study of machine learning systems. Computational models of *cognition* are tools of analysis that allow cognitive scientists to formalize and test hypotheses about how the human mind represents and processes information. Here, I argue analogous computational models can also be used to investigate the internal mechanisms of machine learning systems and, by doing so, allow us to investigate implicit inductive biases.

Using this framing, Part III approaches the problem of characterizing inductive biases as a hierarchical modeling problem in which the extrapolation *behavior* of a population of machine learning systems—rather than details of their internal workings—is viewed as evidence of a systematic inductive bias. The hierarchical modeling framing allows us to

learn a *representation* of the inductive bias implicit in a design specification of machine learning systems that can be analyzed in order to probe the content of this inductive bias. Chapter 7 implements this computational modeling framing by learning a representation of the inductive biases consistent with the observed behavior of a particular family of neural network models. The neural networks in this family share in design choices (for example, the architecture, training procedure, and random initialization scheme) but differ in quantities that are randomized prior to or during training. The machine learning methodology we employed to represent inductive biases, Gaussian processes, allows us to reveal systematic structure—including a preference for low-frequency signal and pathological behavior with depth—in the inductive bias underlying this particular family of neural networks. In two further practical case studies, we use the computational model to predict the generalization properties of neural networks.

Part I

Behavioral studies of neural networks

Chapter 2

Theory of mind and the *false belief* task

The work described in this chapter is published as Erin Grant, Aida Nematzadeh, and Thomas L. Griffiths (2017). "How Can Memory-Augmented Neural Networks Pass a False-Belief Task?" In: *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 427–432.

2.1 Introduction

Question answering poses difficulties to artificial intelligence systems because correctly answering a query often requires sophisticated reasoning and language understanding capacities, and so simply memorizing the answer or searching in a knowledge base is not enough. Despite this challenge, recent neural network models that make use of attention mechanisms in combination with an explicit external memory can successfully answer questions that require more complex forms of reasoning than before (*e.g.*, Sukhbaatar et al. 2015; Henaff et al. 2017). The benchmark dataset for such tasks has become the Facebook bAbi dataset (henceforth, bAbi) (Weston et al. 2016), which is a collection of question-answering tasks in the form of simple narrative episodes—termed *stories*—that are accompanied by questions about the state of the world described in the stories. (See Fig. 2.1 for an example story from this dataset.)

Although bAbi is a start towards enumerating the requirements for human-like reasoning capabilities, it lacks tasks for testing the ability to reason about mental states, which is also necessary for correctly answering questions of the sort that humans encounter regularly. Consider the following:

*Sally and Ann are in the kitchen.
Sally placed the milk in the pantry.
Sally exited the kitchen.
Ann moved the milk to the fridge.*

For a model to correctly answer questions such as *Where would Sally/Ann search for the milk?* it need not only recognize that Sally and Ann have mental representations of the state of the world but also that these representations are inconsistent: Sally believes that the milk is in the pantry while Ann thinks it is in the fridge.

Psychologists have used a task similar to this scenario—termed the *false-belief* task—to examine children’s development of *theory of mind*: the capacity to reason about the mental states of oneself and others (Premack and Woodruff 1978). Most 3-year-old children, after observing such a scenario, answer that Sally would search for the milk in the fridge because they cannot infer Sally’s belief about the location of the milk, which is inconsistent with their own knowledge (*e.g.*, Baron-Cohen 1989; Baron-Cohen et al. 1985). However, most older children are able to identify, correctly, that Sally’s belief is different from theirs in that she thinks that the milk is the pantry.

To answer questions about situations like those that occur in a *false-belief* task, a model needs to use the observed actions in the scenario to infer the mental states of Sally and Ann. In this work, we investigate whether the End-to-End Memory Network (henceforth MemN2N), a recent neural question-answering model (Sukhbaatar et al. 2015) that solves most of the bAbi tasks, is able to answer questions of the same structure as a *false-belief* task. We formulate scenarios to capture different possible causal relations among actions and beliefs, and examine the performance of the model therein. We find that the MemN2N

model performs well only in the presence of strong supervision—when the training and test data share the same casual structure. This result suggests that the model is able to memorize the training data but is unable to learn to reason about mental states and how they cause and are caused by actions.

Furthermore, to simulate the (perhaps inconsistent) beliefs of the participants in a story, we extend the MemN2N model to include a separate memory representation for each participant. We show that this extension improves model performance, suggesting that explicitly modeling agents' knowledge in a disentangled manner is in part sufficient for more human-like reasoning on a *false-belief* task.

2.2 Theory of mind and the *false-belief* task

A theory of mind is integral for an agent to predict and explain the behavior that is caused by the mental representations of other agents, and therefore succeed on tasks such as the *false-belief* task. For children, this capacity is acquired gradually over the course of development. In particular, children undergo several milestones before they develop an adult-like theory of mind: By age two, they can distinguish between external states of the world and internal mental states possessed by cognitive agents (*e.g.*, Meltzoff et al. 1999). By age four, they can distinguish between consistent and inconsistent mental states (*e.g.*, Perner et al. 1987), which allows them to identify a false belief.

Previous computational works have modeled human performance on the *false-belief* task. Some focus on modeling the development of theory of mind by instantiating a model that initially fails but eventually passes the *false-belief* test (Van Overwalle 2010), while others study the settings in which a model can succeed on the task by varying the input data or the model architecture (O'Laughlin and Thagard 2000; Triona et al. 2002; Goodman et al. 2006). However, none of these models use natural language sentences, despite the fact that the psychological *false-belief* task is usually administered verbally in the form of a natural language reasoning problem.

Furthermore, natural language is known to interact with the development of theory of mind. For example, use of mental state terms in child-directed speech (*e.g.*, Slaughter and Gopnik 1996), engagement in pretend play (Youngblade and Dunn 1995), storybook reading (Rosnay and Hughes 2006), and reflection on events in the child's past (Nelson 2007) serve to accelerate its developments, while, in turn, a greater grasp of theory of mind leads to increased linguistic ability (Milligan et al. 2007). In this work, we examine whether a model can learn from natural language about the causal relationship between actions and beliefs, in order to be able to answer questions that require reasoning about mental states.

Mary got the milk there.
 John moved to the bedroom.
 Sandra went back to the kitchen.
 Mary traveled to the hallway.
 Q: Where is the milk? A: hallway

Figure 2.1: An example task from the bAbi dataset.

2.3 Memory networks

The MemN2N model of Sukhbaatar et al. (2015) comprises an external memory cache and mechanisms to read and write to this memory. The model is trained to write a sequence of stories into its external memory and to answer questions about the stories by reading its memory and emitting the correct vocabulary item. At test time, the model is evaluated by the extent to which it can correctly answer questions about a held-out set of test stories.

Formally, the model ingests a sequence of input sentences (x_1, \dots, x_n) and produces, for each input item x_i , both a memory representation m_i and a context representation c_i , which are stored in memory. The model is then presented with a question q_k about the story, for which it produces an internal representation u_k . To answer the question, the model computes a normalized association score p_{ik} between the question representation and each of its stored memory representations:

$$p_{ik} = \frac{\exp\{u_k^T m_i\}}{\sum_j \exp\{u_k^T m_j\}}. \quad (2.1)$$

This weight can be interpreted as an attention mechanism that defines where in memory the model will look for information relevant to the given question.

The model then produces an output representation by way of a linear combination of its context representations, weighted by the attention computed in Eq. (2.1):

$$o_k = \sum_i p_{ik} c_i. \quad (2.2)$$

The output representation is combined with the query representation and decoded by some function f to produce the predicted answer \hat{a} :

$$\hat{a} = f(o_k + u_k). \quad (2.3)$$

Learning model parameters at training time is done by way of stochastic gradient descent in cross entropy error.

	BA	AB	A(B)A
TB	Anne moved the milk to the fridge.	Sally <u>placed</u> the milk in the pantry.	Sally <u>placed</u> the milk in the pantry.
	Sally <u>believes</u> the milk is in the fridge.	Anne moved the milk to the fridge.	Anne moved the milk to the fridge.
	Q: Where did Sally <u>search</u> for the milk?	Q: Where does Sally <u>believe</u> the milk is?	Q: Where did Sally <u>search</u> for the milk?
	A: fridge	A: fridge	A: fridge
FB	Sally <u>believes</u> the milk is in the pantry.	Sally <u>placed</u> the milk in the pantry.	Sally <u>placed</u> the milk in the pantry.
	Sally exited the kitchen.	Sally exited the kitchen.	Sally exited the kitchen.
	Anne moved the milk to the fridge.	Anne moved the milk to the fridge.	Anne moved the milk to the fridge.
	Sally entered the kitchen.	Sally entered the kitchen.	Sally entered the kitchen.
	Q: Where did Sally <u>search</u> for the milk?	Q: Where does Sally <u>believe</u> the milk is?	Q: Where did Sally <u>search</u> for the milk?
	A: pantry	A: pantry	A: pantry

Figure 2.2: Examples of the training data, with the predicates of interest underlined. Note that the *true-belief* (TB) and *false-belief* (FB) test tasks are of the same form as the top and bottom items, respectively, in the last column.

2.4 Simulation 1: MemN2N model

We evaluate the model introduced in the previous section on a set of novel textual reasoning tasks inspired by the *false-belief* task. Our tasks take the form of a sequence of natural language sentences—termed a *story*—and an associated question about the story.

Since we aim to create tasks that, for humans to solve, involve reasoning about other agents’ beliefs, we design various story templates that simulate how different actions give rise to different beliefs, and conversely how different beliefs result in different actions. These stories differ in whether or not the agent who is the subject of the question has observed a change in the state of the world (*i.e.*, the agent has a true belief), or has not (*i.e.*, has a false belief). The stories further differ in whether the belief is observable (*i.e.*, the story explicitly contains sentences such as *Sally believes the milk is in the pantry*) or whether only actions are observable. When the agent harbors a false belief, and the model is asked to predict the action of the agent without explicit reference to the beliefs of the agent in the story, we recover a simulation of the classic *false-belief* task.

With this experimental design, we aim to determine whether the MemN2N model can reason about how actions cause beliefs and vice versa, and how much information needs to be revealed to enable the model to succeed.

Data generation

To generate stories and corresponding questions, we emulate the bAbi (Weston et al. 2016) dataset generation procedure. We define a world of *entities*, which are the people and objects described in the stories, and possible *predicates* that take entities as subject and, optionally, object. Each entity has *properties* that define the predicates of which it can be subject or object. For example, a world may contain *Sally* with the property *is agent* and

apple with the property *is object*. Our rules permit *Sally* to perform the action *displace* on the *apple*.

In this work, we consider a restricted set of *action* and *belief* predicates. Our actions define simple interactions of an agent with the world (e.g., *place, move, enter, exit*) and our beliefs correspond to mental state terms (e.g., *believe, think*), inspired by the terms that children gradually learn to understand and use correctly over the course of development (e.g., Bretherton and Beeghly 1982; Johnson and Wellman 1980). Our templates manipulate the order of action and belief predicates to test how the model reasons about the causal relations between them.

Experimental conditions

Story template. We define a set of templates that correspond to the type of story that we wish to generate. Each template fixes a sequence of predicates and therefore puts constraints on the entities that may fill the template. For example, a template could be the sequence (*drop, pick up, exit*). Completion of the template entails sampling valid entities from the world to fill the subject and object positions of the predicates, producing, for example, the story (*Sally dropped the ball, Sally picked up the ball, Sally exited the room*).

We consider three different template types:

- **BA:** observable beliefs (e.g., *Sally believes the milk is in the pantry*) give rise to observable actions (e.g., *Sally searches the pantry*);
- **AB:** observable actions (e.g., *Sally places the milk in the pantry*) give rise to observable beliefs (e.g., *Sally believes milk is in the pantry*); and
- **A(B)A:** observable actions (e.g., *Sally places the milk in the pantry*) give rise to observable actions (e.g., *Sally searches the pantry*) by way of unobserved beliefs (e.g., *Sally believes the milk is in the pantry*).

Note that the **AB** and **A(B)A** conditions are different in that in **AB**, the question explicitly asks about Sally's belief; in **A(B)A**, on the other hand, the question is about Sally's action, which has been brought about by Sally's unobserved belief.

True vs. false belief. In addition to the type of template, for each story we manipulate whether the agent about whom the question is asked (*i.e.*, *Sally*) has a true belief or a false belief about the state of the world. In the case that the agent has a true belief, the agent observes all changes in the state of the world and thus their beliefs are consistent with the world. On the other hand, in the case that the agent has a false belief, the agent does not observe one or more changes in the state of the world (because, for instance, *Sally* may exit the room), and thus has a belief that is inconsistent with the world.

Training conditions. We have six possible story types as a results of crossing the template types with the true and false belief story types; examples of each of the story types are given in Fig. 2.2. We sample from these story types to produce our training conditions, in the following manner:

- When the training condition is such that $p(\text{false belief}) = 0$ or 1 , we sample only stories with true beliefs or false beliefs, respectively, and when $p(\text{false belief}) = 0.5$, we sample half of our stories with true beliefs and half with false beliefs.
- We sample stories from five different possible groups of templates: **BA**, **AB**, **AB+BA**, **A(B)A** and **AB+BA+A(B)A**.

The **AB+BA** and **AB+BA+A(B)A** conditions provide the model with training data that better approximates the variety of possible scenarios in the world. In these cases, the model observes more ways in which actions and beliefs interact, and thus we would expect it to be able to better generalize to new scenarios. Moreover, **AB+BA** provides the model with the opportunity to learn transitive inference—given that an action (*e.g.*, placing milk in the pantry) results in a belief (*e.g.*, the milk is in the pantry), and a belief (*e.g.*, the milk is in the pantry) can cause an action (*e.g.*, searching for milk in the pantry), a model that reasons about actions and beliefs could learn that an action (*e.g.*, searching for milk in the pantry) is a consequence of an unobservable belief brought about by a preceding action (*e.g.*, placing milk in the pantry).

Crossing template types **BA**, **AB**, **A(B)A**, **AB+BA**, **AB+BA+A(B)A** with $p(\text{false belief}) = \{0.0, 0.5, 1.0\}$ produces our 15 training conditions. We run 10 simulations for each training condition and for each configuration of parameter settings of the MemN2N model.*

Test conditions. We aim to evaluate the model on tasks that require reasoning about latent mental states, in analogy to the classic *false-belief* task; however, such a capacity should apply not only in cases when an agent has a belief that is inconsistent with the state of the world (*i.e.*, a false belief) but also when they have a true belief about the world. We therefore consider two test conditions: a *true-belief* (TB) and a *false-belief* (FB) task. All examples in both of these test conditions share the **A(B)A** template type, but the

*We vary the dimensionality of the memory and word embedding, the number of *computational hops* (accesses to the memory cache to answer a single question), the number of training and testing examples (1000 vs. 10000), and the size of the world from which the dataset of stories is generated (5 vs. 10 vs. 30 entities per entity type, which correspond to the objects, container, etc. in the story). Furthermore, we use the *adjacent* weight tying scheme as described in Sukhbaatar et al. (2015), an initial learning rate of 0.01, and initialize all weight matrices by sampling from a zero-centered normal distribution with a standard deviation of 0.1. As we found no effect of the number of training and testing examples nor of the world size, we collapse all results across these variables. Increasing the embedding size and the number of computational hops increases test accuracy in all conditions (likely due to increased model capacity), and increasing the memory size decreases performance in all conditions (likely because the model must search over more memories to retrieve an answer). However, the qualitative effects we report in this paper are preserved, so we also collapse results across these variables as well.

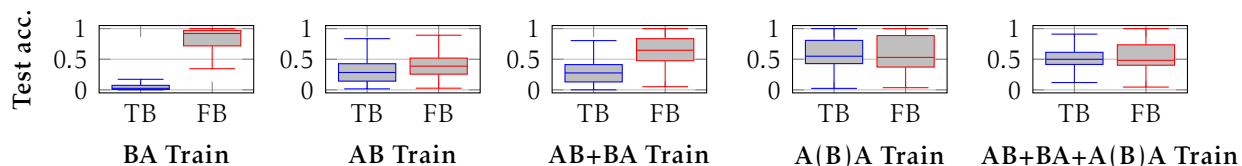


Figure 2.3: Accuracy in Simulation 1. Test accuracies for the *true-belief* (TB) and *false-belief* (FB) tests across training conditions in Simulation 1. We report results for $p(\text{false belief}) = 0.5$, since varying this parameter did not affect results except in the few cases discussed in the text.

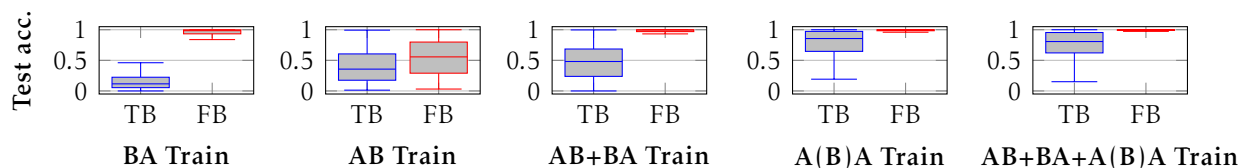


Figure 2.4: Accuracy in Simulation 2. Test accuracies for the *true-belief* (TB) and *false-belief* (FB) tests across training conditions in Simulation 2. As in Fig. 2.3, we report results only for $p(\text{false belief}) = 0.5$.

conditions differ in that the *true-belief* task contains only examples with true beliefs (*i.e.*, $p(\text{false belief}) = 0$), and the *false-belief* task contains only false belief examples (*i.e.*, $p(\text{false belief}) = 1$).

Results

As noted by Sukhbaatar et al. (2015), the MemN2N model exhibits large variance in performance across simulations, and so we show performance by plotting the distribution of test accuracies in boxplot format. In Fig. 2.3, we report accuracy on both test conditions (the *true-belief* (TB) and *false-belief* (FB) tasks) across the training conditions, for $p(\text{false belief}) = 0.5$. The results for $p(\text{false belief}) \in \{0, 1\}$ were similar except in the case of the **AB** story template; we compare this case with the **BA** condition in Fig. 2.5 and discuss in the following. Note that success at test time corresponds to achieving 1.0 accuracy in **both** the TB and FB test conditions.

Training condition BA: Beliefs to actions. The model fails on the TB task in the **BA** training condition, while succeeding on the FB task. This is true no matter the value of $p(\text{false belief})$ (as depicted in Fig. 2.5). To understand why this occurs, consider the following example of a **BA** training story when the false belief occurs:

Sally believes the milk is in the pantry. Sally exited the kitchen. Anne moved the milk to the fridge. Sally entered the kitchen.

Additionally, consider the **BA** training story when the false belief does not occur:

Anne moved the milk to the fridge. Sally believes the milk is in the fridge.

To answer the training question *Where did Sally search for the milk?* the model seems to learn that it should look for the sentence containing *Sally* and a container entity (*i.e.*, *Sally believes the milk is in the fridge*).

This strategy works for the *false-belief* test (see Fig. 2.2, last column, bottom row), because Sally believes that the milk is in the pantry—the location in which she originally placed it—and thus the sentence containing *Sally* and the identity of a container always provides the correct answer. However, this strategy fails on the *true-belief* test (again, see Fig. 2.2, last column, top row), because Sally observes that the milk has been moved, and so no longer believes that the milk is in fridge. This suggests that the model is unable to infer that an observable action changes the mental state of Sally.

Training condition AB: Actions to beliefs. The model is unable to achieve good performance on both the TB and FB tests in the **AB** condition. When the model performs better, it is in cases where the test is very similar to the training condition, *i.e.*, the *false-belief* test with $p(\text{false belief}) = 1$ in training and *true-belief* test with $p(\text{false belief}) = 0$ in training.

Training condition AB+BA: Transitive inference. The model fails on both test tasks in the **AB+BA** training condition. This is evidence that the model cannot reason about the causal relationships between actions and beliefs to perform transitive inference.

Training condition A(B)A: Equivalent to TB/FB test. The model achieves best performance on **A(B)A** in the $p(\text{false belief}) = 0.5$ condition. This again happens because the test and training conditions are similar: the model observes examples of both the FB and TB test tasks in training, and thus receives supervision to give the correct answer at test. However, the model performs well only on the TB task in the $p(\text{false belief}) = 0$, and on the FB task in the $p(\text{false belief}) = 1$ condition. This is because the model does not observe examples like one or the other test condition at training time.

Notably, the performance is not high even in the $p(\text{false belief}) = 0.5$ condition (the median is approximately 55% on both test tasks), despite the fact that the model is given test-like examples at training time. It is therefore not clear that the model is robustly able to solve a conditional reasoning task in which the correct answer is dependent on whether or not the observer sees the movement of the object and thus has a false or true belief. This, along with the model's failure in the other training scenarios, motivates an extension to the model, which we consider in the next section.

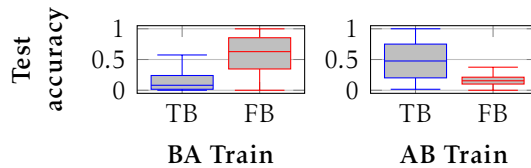
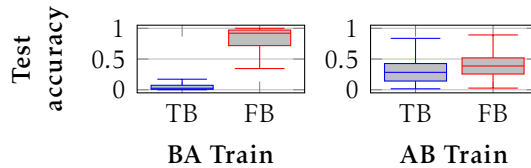
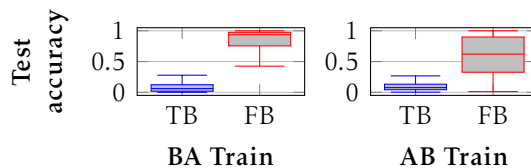
(a) $p(\text{false belief}) = 0$ in training.(b) $p(\text{false belief}) = 0.5$ in training.(c) $p(\text{false belief}) = 1$ in training.

Figure 2.5: From **Simulation 1**. The test accuracy in the **AB** condition is dependent on the value of $p(\text{false belief})$, but not in the **BA** condition.

2.5 Simulation 2: Multiple-observer model

We now propose a model that is given information about whether each agent in the story observes each sentence in the story. In general, this must also be inferred from context, but here we assume such annotations are available to the model as we simply attempt to investigate the effect of this information on the model’s predictions.

Formally, for a story of N input items that describes a situation with M agents, we provide the model with an N -by- $(M + 1)$ *observer annotation matrix* S such that $S_{ij} = 1$ if input item x_i is observable to agent j and 0 otherwise, where we assign the oracle observer (who observes all input items) to the first index. These annotations are used to mask the input such that $M + 1$ (possibly different) stories are produced, each of which corresponds to the story that a particular agent observes. Memory representations, attention over each memory cache, and output representations are computed separately for each observer, and so $M + 1$ output representations are computed, each corresponding to the output of a distinct observer’s memory.

The model then computes an attention weighting over each of the observer memory

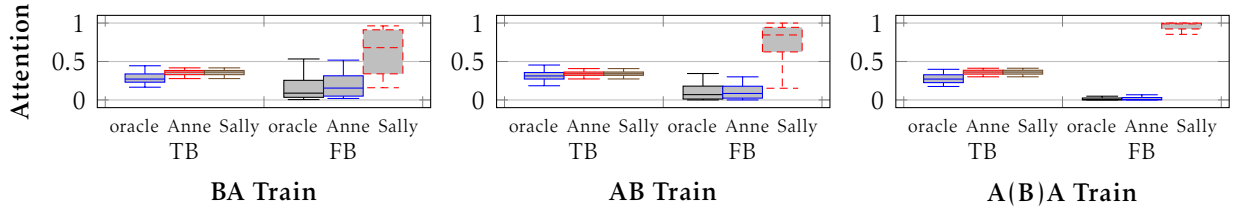


Figure 2.6: Attention in Simulation 2. Visualisation of the attention weighting over memory caches for the *true-belief* (TB) and *false-belief* (FB) tests. We omit the visualization for the **BA+AB** and **BA+AB+A(B)A** training conditions because the test accuracy distribution in Simulation 2 for these conditions is very similar to the **A(B)A** training condition (see Fig. 2.4).

caches (*c.f.*, Eq. (2.1)):

$$r_{k\ell} = \frac{\exp\{u_k^\top o_{k\ell}\}}{\sum_n \exp\{u_k^\top o_{kn}\}}. \quad (2.4)$$

This attention over memory caches is used to compute a weighted combination of the output representations that correspond to the memory cache for each agent (*c.f.*, Eq. (2.3)):

$$\hat{a} = f(u_k + \sum_{\ell} r_{k\ell} o_{k\ell}). \quad (2.5)$$

Note that the model considered in Simulation 1 is exactly this model extension with $r_{k0} = 1$ and $r_{km} = 0, \forall m \neq 0$ (*i.e.*, attention is given only to the oracle memory cache).

In this extension, the model is given explicit information about which observations in a story are available to each agent, by way of the annotation matrix S . However, it must *learn* to reason about this information in order to arrive at the correct answer, as before with how to write to memory and read from memory, and now with how to select over which observer’s knowledge of the story is relevant to answer the question.

Results

We report results of the model extension on the TB and FB tests in Fig. 2.4, as well as a visualization of the attention weights in Fig. 2.6. Our simulated data is composed of scenarios with only two agents, and therefore the extended model attends over three memory caches (one for the oracle that observes everything, one for Anne, and one for Sally, about whom the question is asked).

The extended model achieves higher accuracy across all training conditions. Notably, the model performs near perfectly (*i.e.*, both TB and FB are close to 1) in the **AB+BA+A(B)A** case, meaning that the model can learn to ignore irrelevant training stimuli. This suggests that awareness of agent’s knowledge about the state of the world helps in a task of reasoning about latent mental states.

Furthermore, the attention plots show that the model learns to attend to the memory representation of Sally in the FB test, which contains the information about how to answer questions about Sally’s actions and beliefs. On the other hand, in the TB test, the model does not attend differently to the different memory caches, because the observations stored in all caches are the same.

2.6 Conclusions

We investigated whether a recent language learning model that succeeds on a suite of textual reasoning tasks is able to succeed in a task that requires reasoning about latent mental states. We found that the model is unable to succeed in a set of simulated *true-belief* and *false-belief* tasks unless it has observed at training time situations that have the same structure as the test tasks, even if the diversity of the data is increased. This strongly suggests that the model is not reasoning about the state of the world, nor about mental representations thereof, but is simply memorizing its input. As a consequence, the model will not be able to succeed in a task of reasoning that differs greatly from the situations that it has observed at training time. This is in contrast to the novelty of situations that people encounter regularly, in which they must reason about the causal relationship between events in the world and latent mental states.

However, incorporating a simple mechanism that informs the model that there may be multiple observers with differing representations of the story allows the model to achieve higher performance on the simulated *false-belief* and *true-belief* tasks. Under this modification, the model does not simply memorize the training data but also learns to use knowledge that agents have (perhaps conflicting) observations about the story in order to answer the question. We could interpret this as analogous to the development of theory of mind in that, when a child is able to reason about others’ knowledge of and beliefs about the world, the child succeeds on tests of theory of mind such as the *false-belief* task. A further direction of research could investigate whether manipulating variables in the training data (e.g., frequency of mental state terms) affects the model’s performance in a manner similar to how a child’s developmental trajectory would be affected.

Chapter 3

Rule- and exemplar-based generalization

The work described in this chapter is published as Ishita Dasgupta*, Erin Grant*, and Thomas L Griffiths (2022). “Distinguishing rule-and exemplar-based generalization in learning systems”. In: *Proceedings of the International Conference on Machine Learning*.

3.1 Introduction

Extrapolation or generalization—decisions on unseen datapoints—is always underdetermined by data; which particular extrapolation behavior a machine learning system exhibits is determined by its inductive biases (Mitchell 1980). When those inductive biases are opaque—as is often the case with many modern machine learning systems (Geirhos, Jacobsen, et al. 2020; D’Amour et al. 2020)—we can instead turn to empirical investigation of the *behavior* of a system to reveal the system’s *implicit* inductive biases. Cognitive psychology provides a rich basis for experimental designs to study the often-opaque human cognitive system via its external behavior; these designs can be leveraged to distinguish between competing hypotheses about a machine learning system’s inductive biases as well (*e.g.*, Ritter, Barrett, et al. 2017; Lake, Ullman, et al. 2018; Dasgupta et al. 2020).

We draw on cognitive psychology to construct a protocol that isolates the inductive biases determining how a machine learning system generalizes feature-based categories such as those in Fig. 3.1. A key property of such categorization problems is the presence of a *distractor* dimension that does not play a causal role in the underlying category boundary; the ground truth categorization is determined by a *discriminant* dimension. Such problems are ubiquitous in machine learning applications (*e.g.*, Beery et al. 2018), where learned associations between the distractor and the categorization label are termed “spurious” (Arjovsky et al. 2019). The tendency to acquire (potentially harmful) spurious associations is an example of a downstream consequence of implicit inductive bias, and so characterizing such implicit inductive biases is of both theoretical and practical interest.

We use abstract problem settings such as that in Fig. 3.1 to identify and isolate two distinct inductive biases underlying feature-based category learning. The first, *feature-level bias*, expresses a preference for some features over others to support a decision boundary (*e.g.*, preferring shape over color). The second, *exemplar bias*—vs. *rule bias*—expresses a preference for feature-dense (vs. feature-sparse) decision boundaries (*e.g.*, a boundary informed by both shape and color, vs. only one of the two features). Our protocol presents data conditions that manipulate feature co-occurrences observed during training such that the resulting extrapolation behavior is diagnostic of these inductive biases in the learner.

The experimental setup underlying our training and testing conditions is similar to existing works in “combinatorial generalization” (Andreas et al. 2016; Johnson, Hariharan, et al. 2017) and “subgroup fairness” (Sagawa, Koh, et al. 2020; Sagawa, Raghunathan,

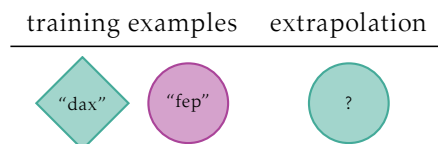


Figure 3.1: Example of a data condition: Data often underdetermines a decision boundary; here, it is unclear whether shape or color determines object label (“dax” vs “fep”). How a learner extrapolates to new stimuli reveals inductive bias.

et al. 2020). Our work also makes several independent contributions: We identify and isolate two distinct inductive biases that affect extrapolation of feature-based categories, and we examine these across models in an expository points-in-a-plane setting, as well as in more naturalistic text and image domains. We demonstrate that existing measures of feature co-occurrence and extrapolation behavior (“spurious correlation” and “worst-group accuracy,” Sagawa, Raghunathan, et al. 2020) are insufficient to characterize these inductive biases. Finally, we consider the normative question: *What extrapolation behavior is desirable for a given application?* We provide a preliminary answer by discussing the relevance of the inductive biases we identify to related work in systematic generalization, fairness, and data augmentation.

3.2 Inductive biases in category learning

We start by introducing the two inductive biases of interest. **Feature-level bias** characterizes *which* feature a system finds *easier* or *harder* to learn and thus which feature a system will utilize when both are associated with the category label. This kind of feature-level bias has been studied extensively in human cognition (Landau et al. 1988; Hudson Kam and Newport 2005), and specific feature-level biases—mostly notably the “shape-bias,” the tendency to generalize image category labels according to shape rather than according to color or texture—have been revisited in the context of recent neural network models (Ritter, Barrett, et al. 2017; Hermann et al. 2019; Geirhos, Rubisch, et al. 2019). We examine feature-level bias for arbitrary features, as well as demonstrate how this bias interacts with—but is distinct from—another kind of inductive bias, to be discussed next.

Exemplar (or rule) bias characterizes *how* a system uses features to inform decisions by trading off *exemplar- and rule-based generalization*. A rule-based decision is made on the basis of minimal features that support the category boundary (e.g., Ashby and Townsend 1986), while an exemplar-based decision-maker generalizes on the basis of similarity to category exemplars (e.g., Shepard and Chang 1963), invoking many or all features that underlie a category. Extensive empirical work in cognitive psychology has found evidence of both kinds of generalization in humans (Nosofsky et al. 1989; Rips 1989; Allen and Brooks 1991; Smith and Sloman 1994). This trade-off can be understood as a continuum that varies the number of features employed to discriminate between categories (Pothos 2005).

Feature-level bias and exemplar bias are **practically relevant** because they describe how a learning system uses features to extrapolate, and different problem settings call for different ways of doing so. An exemplar-based system that depends on all features, and is not invariant to any of them, suffers when not all feature combinations are observed and systematic generalization to unobserved combinations is expected (Lake, Ullman, et al. 2018; Marcus 2018; Arjovsky et al. 2019). On the other hand, a rule-based system that applies the same category decision rules across all data regions might over-generalize, which is undesirable in naturally occurring long-tailed distributions (Feldman and Zhang

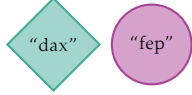

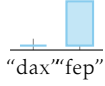
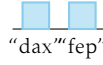
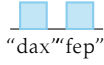
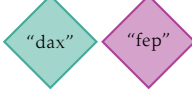

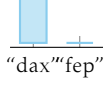
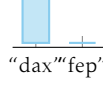
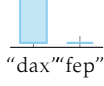


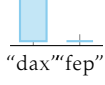
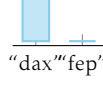
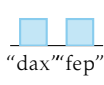
condition	observations		ratio of predictions		
	training examples	extrapolation	humans (<i>shape-biased</i>)	rule-based (<i>no feature bias</i>)	exemplar-based (<i>no feature bias</i>)
cue conflict					
zero shot					
partial exposure					

Figure 3.2: Illustrative category learning experiment: Training examples from the 3 independent training conditions, the extrapolation test, and characteristic behavior for learners with different inductive biases. We formalize the training conditions in Fig. 3.3.

2020; Feldman 2020; Brown et al. 2020). Diagnosing exemplar vs. rule bias is therefore of both theoretical and practical interest. In Section 3.6, we give a concrete example in a fairness setting—where certain regions of the data support is underrepresented but we want comparable accuracy on these regions nonetheless—in which understanding the inductive biases of the learning system allows for a data intervention that improves performance.

We now build intuitions for how **the category learning paradigm in Fig. 3.2 isolates feature-level bias and exemplar bias**. The stimuli Fig. 3.2 vary along two feature dimensions, shape and color. Color determines the label of an object (*i.e.*, green objects are “dax”; purple are “fep”, using arbitrary names to emphasize that the category is novel to humans as well as to machine learning systems). Shape is unrelated to the underlying category structure and acts as a distractor. Participants (either humans or artificial learning systems) are independently placed in three different conditions—**cue conflict**, **zero shot**, or **partial exposure**—that vary in coverage of the feature space. After observing the *training examples*, the participant is presented with an *extrapolation* test consisting of an example outside the support of feature combinations observed during training (*i.e.*, they must classify the green circle as a “dax” or a “fep”). We explain below how differences in classification behavior on this extrapolation test isolate feature-level bias as well as exemplar-vs-rule bias, but first: We encourage the reader to try the experiment themselves to examine their intuitions.

Cue conflict (CC, top row, Fig. 3.2). The data presented in this condition confound color and shape (*i.e.*, color and shape are equally predictive of the category boundary). How a system generalizes here directly measures its feature-level bias towards color or shape.

Characteristic behavior (right half of Fig. 3.2). Since humans have an established shape bias (Landau et al. 1988), we expect that humans will classify the test item according to the object that shares its shape, not its color; in this case, as a “fep.” However, this inductive bias is independent of whether a reasoner is rule- or exemplar-based; neither has an *a priori* propensity for features, both are equally likely to classify the test item as a “dax” or a “fep.”

Zero shot (ZS, middle row, Fig. 3.2). This condition requires extrapolation to a new feature value “zero-shot” (*i.e.*, without prior exposure). This setting is often used to examine out-of-domain (OOD) and compositional generalization in machine learning (Xian et al. 2018). Behavior in this condition reveals whether the model has learned the discriminating features and whether it can extrapolate to new feature values, and thus acts as a baseline.

Characteristic behavior (right half, Fig. 3.2). Rule- and exemplar-based behavior in this condition is confounded. A rule-based learner infers the minimal rule that color determines label, does not assign any predictive value to shape, and therefore classifies the extrapolation stimulus based on color as a “dax.” An exemplar-based learner categorizes based on the similarity along all feature dimensions of the extrapolation stimulus to category exemplars. Neither training exemplars have any overlap with the test stimulus along the shape dimension, but the “dax” overlaps along the color dimension, and the learner categorizes it as a “dax.”

Partial exposure (PE, bottom row, Fig. 3.2). Compared to zero shot, participants in this condition also receive “partial exposure” to a new feature value (*i.e.*, *circle*) along the shape dimension. The extrapolation test in this condition is most similar to *combinatorial zero-shot generalization* (*e.g.*, Lake and Baroni 2018a), where the learner is exposed independently to all feature values but has to generalize to a new combination.

Characteristic behavior (right half of Fig. 3.2). This setting meaningfully distinguishes rule- and exemplar-based generalization. To understand this distinction, we contrast this condition to the cue-conflict condition. The addition of the purple diamond-shaped “fep” means the learner has seen both a diamond and a circle labeled “fep”. A rule-based system takes this as direct evidence that shape is *not* predictive of category label and classifies the extrapolation stimulus on the basis of color as a “dax.” This is typically also how humans extrapolate. This additional training example, however, does not impact an exemplar-based system, since it does not share any features with the extrapolation stimulus. The exemplar-based reasoner classifies on the basis of feature-overlap with training exemplars and is therefore indifferent, exactly as in the cue-conflict condition.

3.3 A protocol for measuring inductive bias

We embed the structure of the category learning problem discussed in Section 3.2 into a statistical learning problem that can be applied across domains to test black-box learners.

Problem setting. We consider the *oracle* compositional setting of Andreas (2019) in which inputs are a composition of categorical attributes with two latent binary features,

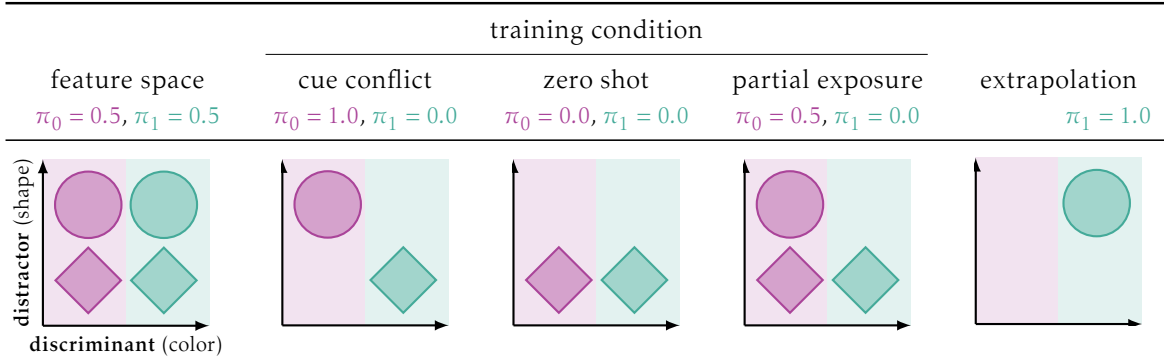


Figure 3.3: Formalizing the illustrative experiment: The experiment from Fig. 3.2 expressed in terms of the formalism in Section 3.3 with color as \mathbf{z}_{dist} and shape as \mathbf{z}_{disc} . Background colors indicate the true category.

$\mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}} \in \{0, 1\}$ that jointly determine the observation $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ via some mapping $g: \{0, 1\}^2 \rightarrow \mathcal{X}$; see Fig. 3.3. We consider the binary classification task of fitting a model $\hat{f}: \mathcal{X} \rightarrow \{0, 1\}$ from a given model family \mathcal{F} to predict a class for each observation. One of the underlying features, the *discriminant*, \mathbf{z}_{disc} , defines the decision boundary; the other one, the *distractor*, \mathbf{z}_{dist} , is not independently predictive of the label.

This specifies a generative process $\mathbf{x}, \mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}} \sim p(\mathbf{x} | \mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}}) p(\mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}})$. $p(\mathbf{x} | \mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}})$ is either generated (*e.g.*, in Section 3.4), or the empirical distribution of the subset of datapoints \mathbf{x} with the corresponding underlying feature values (assuming access to these annotations, *e.g.*, in Sections 3.5 and 3.6). $p(\mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}})$ is varied across training conditions, as outlined below.

Training conditions. The upper-right quadrant in all subfigures of Fig. 3.3, for which $p(\mathbf{z}_{\text{disc}} = 1, \mathbf{z}_{\text{dist}} = 1) = 1$, acts as a hold-out set on which we can evaluate generalization to an unseen combination of attribute values. We produce multiple training conditions with the remaining three quadrants of data by manipulating $p(\mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}})$. All the analyses in this work compare model extrapolation to the held-out test quadrant across various training conditions.

To equalize the class base rates we balance all training conditions across the discriminant; *i.e.*, we enforce $p(\mathbf{z}_{\text{disc}} = 0) = p(\mathbf{z}_{\text{disc}} = 1) = 0.5$. We also fix the number of datapoints across all conditions at N ; With these constraints, we can control $p(\mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}})$ via two degrees of freedom: $\pi_0 = p(\mathbf{z}_{\text{dist}} = 1 | \mathbf{z}_{\text{disc}} = 0)$ (this implicitly fixes $p(\mathbf{z}_{\text{dist}} = 0 | \mathbf{z}_{\text{disc}} = 0) = 1 - \pi_0$ to balance the dataset); and $\pi_1 = p(\mathbf{z}_{\text{dist}} = 1 | \mathbf{z}_{\text{disc}} = 1)$. The three conditions in Section 3.2, as well as the held-out test set, correspond to particular settings of π_0 and π_1 (shown in Fig. 3.3, more in Appendix B.2).

Measuring inductive bias. We measure *feature-level bias* as deviation from chance performance in the CC condition. *Exemplar bias* is measured as the difference between performance in the partial-exposure condition and zero-shot condition—no difference indicates rule-based generalization, the magnitude of the difference measures exemplar

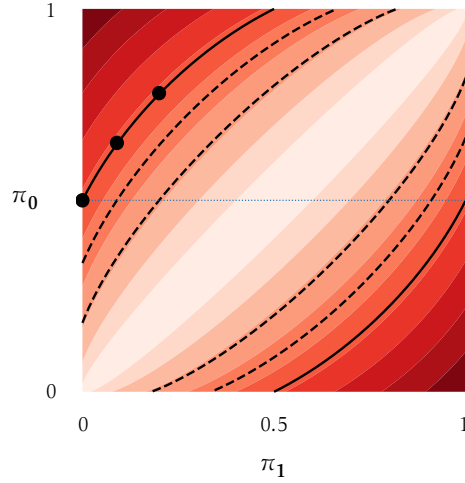


Figure 3.4: Spurious correlation (Eq. (3.3)).

bias. Formally, for a given model family \mathcal{F} , let \mathbf{g}^{ZS} denote the result of selecting a model from \mathcal{F} by training in the zero-shot condition, and similarly \mathbf{g}^{PE} and \mathbf{g}^{CC} . We define FLB and EvR as:

$$\text{FLB}(\mathcal{F}) = \mathbb{E}[A(y, \mathbf{g}^{\text{CC}}(\mathbf{x}))] - 0.5, \quad (3.1)$$

$$\text{EvR}(\mathcal{F}) = \mathbb{E}[A(y, \mathbf{g}^{\text{ZS}}(\mathbf{x}))] - \mathbb{E}[A(y, \mathbf{g}^{\text{PE}}(\mathbf{x}))] \quad (3.2)$$

where the expectation is taken with respect to the data distribution under the extrapolation region ($p(\mathbf{x}, y \mid \pi_0 = 1, \pi_1 = 1)$), and A is the 0-1 accuracy. FLB takes values between -0.5 and 0.5 (indicating bias toward \mathbf{z}_{dist} or \mathbf{z}_{disc} , respectively); 0 represents no feature bias. EvR takes values between 0 and 1 (indicating rule bias and exemplar bias, respectively).

Related formalisms and spurious correlation. This binary formulation of discriminant and distractor features has previously been studied in the context of spurious correlation (Sagawa, Raghunathan, et al. 2020). Rather than independently varying occupancy in the four quadrants, Sagawa, Raghunathan, et al. (2020) directly manipulate the (spurious) linear correlation between the distractor and the discriminant features (p_{maj}). In combinatorial feature spaces, a scalar spurious correlation insufficiently specifies the data distribution. The linear correlation coefficient ρ between \mathbf{z}_{disc} and \mathbf{z}_{dist} —henceforth *spurious correlation*—can be written in terms of π_0 and π_1 via $\alpha = \frac{\pi_0 - \pi_1}{2}$ and $\beta = \frac{\pi_0 + \pi_1}{2}$ as

$$\rho(\pi_0, \pi_1) = \frac{\alpha}{\sqrt{\beta(1 - \beta)}}. \quad (3.3)$$

Different combinations of π_0 and π_1 give equal ρ (see the contours in Fig. 3.4, with markings for points along the equi-correlation contour from partial exposure ($\pi_0 = 0.5, \pi_1 = 0.0, \rho = 0.58$)); while nonetheless producing qualitatively different extrapolation behavior, as we demonstrate in later sections. This indicates that sensitivity to spurious correlation

insufficiently specifies extrapolation behavior. We argue for a formulation like ours—based on manipulating feature *combinations*—that can tease apart distinct inductive biases at the level of what features a system finds easier to learn (FLB) as well as how to use these features to inform a decision boundary (EvR).

3.4 2-D classification example

To illustrate our framework in a simple statistical learning problem and quantitatively confirm the intuitions outlined in Section 3.2, we consider a two-dimensional classification problem. The feature dimensions are orthogonal bases in 2D space, and we define the data generating procedure as

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}}) &= \mathcal{N}(\mu, 1.0); \\ \mu &= \alpha \times [2\mathbf{z}_{\text{disc}} - 1, 2\mathbf{z}_{\text{dist}} - 1], \end{aligned} \tag{3.4}$$

where, as specified in Section 3.3, $\mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}} \in \{0, 1\}$, $p(\mathbf{z}_{\text{disc}}, \mathbf{z}_{\text{dist}})$ is determined by the training condition. \mathbf{z}_{disc} determines class labels, \mathbf{z}_{dist} is a distractor, α is fixed at 3, and $N = 300$ datapoints are in each class. The group with $\mathbf{z}_{\text{dist}} = \mathbf{z}_{\text{disc}} = 1$ is assigned the test set.

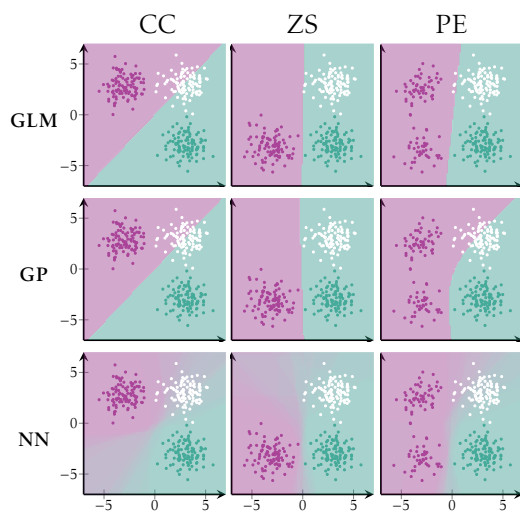
Model families and nomenclature.

NN: We train feedforward ReLU classifiers with varying numbers of hidden layers and hidden units. We use the scikit-learn implementation with default parameters, run 20 times for confidence intervals.

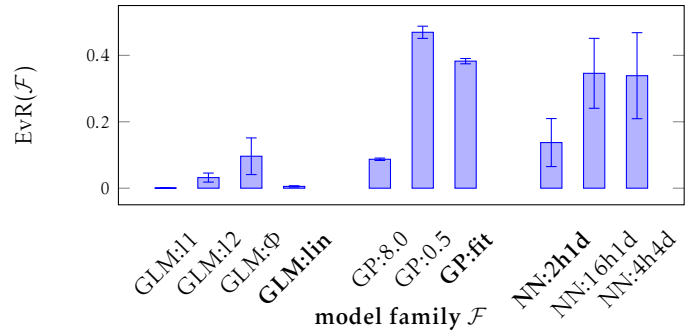
Generalized linear model (GLM): Parametric models allow us to formalize the feature-sparsity that characterizes rule-based learners. Linear logistic regression is sparse by definition (it has access to only linear features). We generalize this model by expanding the feature space to include a nonlinear interaction Φ and examine L1 and L2 regularization in a GLM over this altered feature space.

GP: Nonparametric kernel methods allow us to formalize exemplar-based generalization, where generalizations are made on the basis of feature-dense similarity to training data. We examine the performance of GPs with radial basis function (RBF) kernels. We fit the kernel lengthscale using gradient descent on the log marginal likelihood of the data (Rasmussen 2003) (giving 5.2) as well as vary it (adjusting “locality” in decision boundaries); GP:8.0 denotes a GP with lengthscale value of 8.0.

We can implement explicit rule- and exemplar-based models in the synthetic setting since we know the features over which to build parametric or similarity-based models respectively, so we use it to validate our measures. In most application domains (including those in Sections 3.5 and 3.6) feature learning is automated (Hinton and Salakhutdinov 2006), making it difficult to specify the corresponding GLM or GP.



(a) Decision boundaries averaged across 20 runs. Training datapoints are green or purple by label; test are white.



(b) EvR reflects exemplar-vs-rule propensity both within and across model families. The EvR across model families, computed across 20 runs, error bars represent 95% CIs. The GLMs are largely rule-based and show low EvR. Even within GLMs, sparsity regularization gives lower EvR. GPs are largely exemplar-based and show high EvR. Even within GPs, lower lengthscales give higher EvR. NNs lie in-between, with larger NNs giving higher EvR.

Figure 3.5: Simple 2-D classification (Section 3.4) The specific model used in (a) are bolded in (b).

Comparing cue conflict, zero shot, and partial exposure

We consider one model from each class: NN with 1 hidden layer of 2 units (NN:2h1d); linear GLM (GLM:lin); RBF GP with fitted lengthscale (GP:fit). The decision boundaries learned by these models are shown in Fig. 3.5a. $\mathbf{z}_{\text{dist}}, \mathbf{z}_{\text{disc}}$ are equivalent by design, and permit no feature-level bias, so cue conflict is exactly at chance. This lets us focus on validating our novel protocol for measuring EvR without confounds. We generalize to cases with feature-level bias in later sections. The GLM, sparse and therefore rule-based by definition, can only learn a linear boundary. It is therefore unaffected by the distractor dimension, showing no difference in extrapolation behavior between zero shot and partial exposure (zero EvR). On the other hand, the GP is exemplar-based by definition and displays a high EvR. The NN shows an intermediate EvR, more rule-based than the purely-exemplar-based GP but not entirely rule-based like the GLM.

The influence of model properties on EvR

We first examine EvR in our control model classes (GLMs and GPs) to validate that it tracks rule- vs exemplar-based extrapolation, followed by analyses of various NNs.

Regularized GLMs: EvR reduces with rule propensity. A key property of rule propensity is sparsity in feature space. A linear GLM (GLM:lin) is sparse by definition, we examine a GLM on an expanded feature set so we can manipulate this sparsity. The additional feature $\Phi \propto \mathbf{z}_{\text{dist}} * \mathbf{z}_{\text{disc}}$ is the product of the observed features and normalizing by α . We

compute EvR for this GLM with different regularizers (regularization weight 1.0), shown in Fig. 3.5b.

GLM with no regularization (GLM: Φ) displays a significant EvR. L2 regularization reduces it but L1 (which directly induces feature sparsity*) brings it to zero (or perfectly rule-based). This demonstrates that a low EvR tracks rule propensity via feature-level sparsity.

Lengthscales in GPs: EvR increases with exemplar propensity. A sufficient condition for exemplar propensity is the locality of decision boundaries. We can directly manipulate this in a GPs with its lengthscale. We evaluate EvR in GPs with RBF kernels of different lengthscales in Fig. 3.5b. We find that the EvR is lowest with high lengthscales and grows as the lengthscale reduces, demonstrating that a high EvR tracks exemplar propensity via locality of decision boundaries.

NNs: The necessary but insufficient role of expressivity. The results from GLMs and GPs indicate that some ways to reduce expressivity (L1 regularization in GLMs and high lengthscale in RBF GPs) encourage rule propensity over exemplar propensity (thereby a lower EvR). We manipulate the most common variable in NN expressivity—its size.

We increase the width of an NN with fixed depth of 1 (Fig. 3.5b) and find that the EvR increases. A deep NN with the same number of units, however, exhibits comparable EvR to a wide network. Deeper networks with the same number of units are more expressive than wide ones (Raghu et al. 2017), indicating that excess expressivity, while necessary, is not the sole driver of EvR.

EvR is distinct from sensitivity to spurious correlation

A crucial difference between the zero shot and the partial exposure conditions is that the partial exposure condition creates a (spurious) correlation $\rho = 0.58$ between \mathbf{z}_{dist} and \mathbf{z}_{disc} . Is sensitivity to this spurious correlation (ρ) the sole driver of the difference in performances between the partial exposure and zero shot conditions, *i.e.*, of the EvR? We show that this is not the case; the EvR is measuring something distinct. As described in Section 3.3, there are multiple data-settings with the same ρ . We consider training conditions specified by other π_0, π_1 that have the same ρ as the partial exposure condition (dots along the solid contour in Fig. 3.4). We find that performance on the extrapolation quadrant after training on these new data distributions is much higher (and closer to zero shot performance) than when trained on the partial exposure condition—even though ρ is exactly the same. This indicates that performance on the partial exposure condition (normalized by zero shot performance to give the EvR) is uniquely indicative of something different from sensitivity to spurious correlations—it measures the inductive bias toward exemplar-vs-rule based extrapolation.

We can reduce ρ in different ways by increasing π_1 or by reducing π_0 . We find that these are not equivalent and result in different extrapolation behaviors (*e.g.*, increasing π_1

*Weight sparsity from L1-regularizer is equivalent to feature-sparsity only in special cases, including GLM.

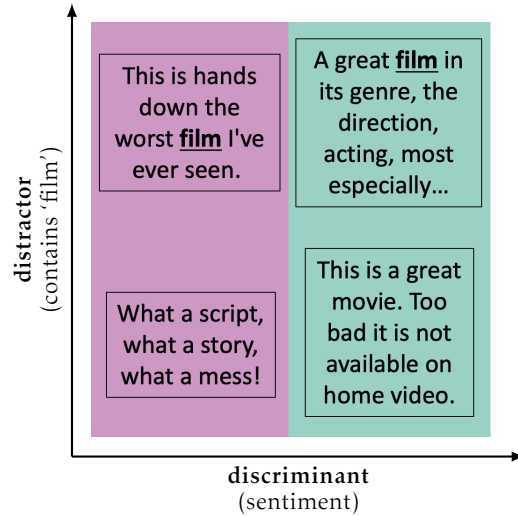


Figure 3.6: Example stimuli from the IMDb dataset.

gives more rule-based generalization than reducing π_0 ; see results for the 2-D classification setting in Appendix B.2 and for the vision domain in Fig. 3.7c). This has implications for data manipulation methods (*e.g.*, subsampling or augmentation) that manipulate this ρ to control extrapolation. This further supports that spurious correlation alone cannot explain extrapolation behavior, highlighting the importance of FLB and EvR that measure behavior under different feature *combinations* in training.

Conclusions. EvR tracks exemplar- and rule-based extrapolation, as validated on interpretable models such as GLMs and GPs. In particular, EvR decreases with reductions in expressivity mediated by regularization and lengthscales, and, in NNs, also decreases with (some kinds of) expressivity. Finally, sensitivity to spurious correlation cannot explain the EvR.

3.5 IMDb text classification

In this section, we demonstrate our protocol on a standard text classification task: sentiment analysis on the Internet Movie Database Movie Reviews (IMDb) dataset (Maas et al. 2011).

Selecting features. The sentiment label (“positive” or “negative”) is the discriminant \mathbf{z}_{disc} . We manufacture an orthogonal distractor \mathbf{z}_{dist} as the presence or absence of a word that occurs in roughly 50% of the sentences in the dataset and does not occur more frequently for either positive or negative reviews. Some examples are “film” and “you”: we use the word “film” (see Fig. 3.6).

Models. We train a single-layer long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) model of 20 hidden units on each condition and test on the held-out quadrant. We exclude models that do not reach 80% validation accuracy.

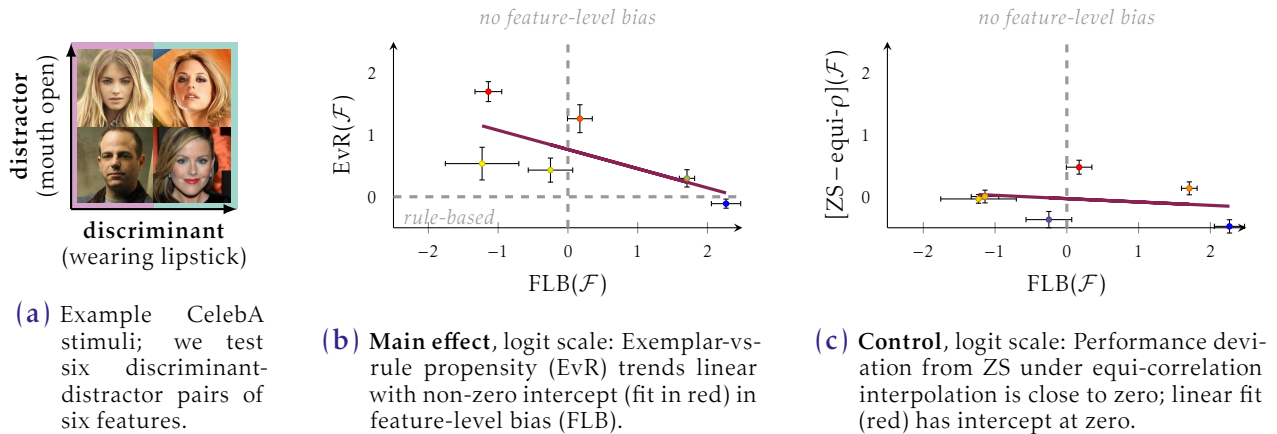


Figure 3.7: CelebA results. Stimuli and results on various feature pairings from the CelebA domain (Section 3.6). Error bars represent 95% confidence intervals across ResNets of various sizes. See figure sub-captions and main text for details.

Feature-level bias. The distractor \mathbf{z}_{dist} is easier to learn than the discriminant \mathbf{z}_{disc} , as reflected in the cue conflict condition (19.7%, $\text{FLB} = -0.3$).

Exemplar bias. We see good performance in zero shot (84%): Despite never having seen the word “film,” the system can generalize to reviews containing it. The performance in partial exposure drops significantly (30.1%) giving a large EvR ($\text{EvR} = 0.54$), indicating exemplar-based reasoning. As such, the exemplar-based tendency to utilize an additional unnecessary feature (*e.g.*, the presence of the word “film”) hurts performance on the extrapolation quadrant. Performance in PE is higher than in CC, indicating that the system can learn to use the discriminant (*i.e.*, it is not purely relying on FLB).

3.6 CelebA image classification

We now test our protocol on a standard classification task on a large-scale image dataset, CelebFaces Attributes (CelebA) (Liu, Luo, et al. 2015). Each image in this dataset is labeled with 40 binary attributes, each of which can be assigned discriminant or distractor. We examine FLB and EvR for standard models across different feature pairs, and discuss the practical implications of our findings.

Selecting features. We select feature pairs that split the data roughly evenly and thus maximizing the number of training datapoints in each quadrant. We carry out our analyses across a range of feature pairs; an example is depicted in Fig. 3.7a, and further details are in the Appendix.

Models. We train residual neural network (ResNet) (He et al. 2016) models of various depths ($\{10, 18, 34\}$) and widths ($\{2, 4, 8, 16, 32, 64\}$) on 6 different choices for feature pairs, with standard hyperparameters (see Appendix B.2 for the complete feature set). We limit our analyses to networks that achieve at least 75% validation accuracy (on held-out

samples from its own training distribution) to ensure that, despite differences in data variability across training conditions, all models learn a meaningful decision boundary.

Feature-level bias. There is a wide range of FLB across feature pairs; e.g., “male” is easier to learn than “high cheekbones” giving high FLB, and “mouth open” and “wearing lipstick” are equally difficult and give FLB of close to 0. FLB values for each feature pair were consistent across ResNet widths and depths.

Exemplar-rule bias. We observe good ZS performance: the models can generalize to new feature values outside the training support. We see a wide range of EvR across feature pairs, Fig. 3.7b. Across all feature pairs, the EvR is non-negative: generalization in the PE condition is always worse (or not significantly better) than in the ZS condition. Further, we see a linear correlation between EvR and FLB in logit space across feature pairs. EvR therefore depends on how easy or hard the features are to learn. The key, however, is that this regression of the EvR onto FLB has a positive intercept: there is a positive EvR even for feature pairs with no FLB. That is, we see lower performance in PE compared to ZS (a nonzero EvR, exemplar propensity) even when FLB is controlled for.

We find no differences in EvR across ResNet widths and depths: Fig. 3.7b plots EvR and FLB averaged over ResNet sizes.[†] One explanation is that the features in CelebA are complex; to learn these, we need reasonably high model expressivity, and differences in parameter count do not further modulate EvR. This is consistent with findings in Section 3.4 where expressivity is necessary but not sufficient for increases in EvR: we see a jump in EvR going from NN:2h1d to NN:16h1d, but no further change going to the even more expressive NN:4h4d.

Controlling spurious correlation. We replicate the findings in Section 3.4: the EvR cannot be explained by sensitivity to spurious correlation ρ . This is demonstrated in Fig. 3.7c, where we substitute performance in the PE condition with performance in a different data condition ($\pi_0 = 0.825, \pi_1 = 0.25$) with the same $\rho = 0.58$ as in the PE condition. We find none of the effects discussed above, indicating that the PE condition is measuring something unique—exemplar-vs-rule propensity—which is not accounted for by sensitivity to spurious correlation. Further, EvR does not increase with model expressivity, unlike sensitivity to spurious correlation (Sagawa, Raghunathan, et al. 2020).

Practical implications of the EvR. The nonzero EvR (*i.e.*, exemplar bias) reveals that models are better at extrapolating zero-shot to a new feature value than when they have partial exposure to that feature value *even though the additional data need not change the learned decision boundary*. In particular, the training examples added in PE can be classified with the decision function from ZS without incurring additional training loss. A rule-based system recognizes this and bases its generalization on the minimal features that support the category boundary. However, an exemplar-based model changes its decision boundary in response to this additional data.

PE-approximating data distributions ($\pi_0 \approx 0.5, \pi_1 \approx 0.0$) occur naturally. For example, as Sagawa, Raghunathan, et al. (2020) observe, “blond” “male”s are under-represented

[†]We report width-and-depth-specific results in Appendix B.2.

in CelebA. Consistent with the rest of our results, we find better classification for the extrapolation quadrant (blond males) if we discard data from an adjacent quadrant (blond non-males, or non-blond males) simulating the zero-shot condition, as opposed to the PE condition if such data is included: ResNet10, width 2, gives $ZS = 75.12 \pm 3.09\%$; $PE = 60.22 \pm 7.27\%$ for $\mathbf{z}_{\text{disc}} = \text{“male”}$ (discard blond non-males to get ZS) and $ZS = 68.16 \pm 3.34\%$; $PE = 49.78 \pm 3.76\%$ for $\mathbf{z}_{\text{disc}} = \text{“blond”}$ (discard non-blond males to get ZS).

These results demonstrate the practical impact of understanding the exemplar-vs-rule bias in a model: an exemplar biased model (like the ResNet here) generalizes poorly in combinatorial settings, and can be made to generalize better by discarding an entire quadrant of data. Previous sub-sampling approaches (Sagawa, Raghunathan, et al. 2020; Haixiang et al. 2017) do not manipulate feature combinations and only manipulate spurious correlations. The aforementioned analyses (Fig. 3.4) and results (Fig. 3.7c) demonstrate that this underspecifies extrapolation behavior.

3.7 Related work and future directions

Model design for systematic generalization. Rule-based generalization permits systematic extrapolation in combinatorial domains. This systematicity has been found lacking in neural networks (Lake and Baroni 2018b; Barrett et al. 2018), leading to renewed interest in hybrid symbolic–connectionist methods (*e.g.*, Garnelo and Shanahan 2019). However, works proposing new methods usually do not examine how feature co-occurrences modulate the systematicity of extrapolation. Using our protocol to examine exemplar- vs. rule-based generalization in these models is a promising future direction.

Learning causal features. Rule-based generalization, is equivalent to learning causal features under the assumption that the causal model is the simplest model that explains the data. Recent work has investigated data settings that separate causal features from spurious ones (*e.g.*, Arjovsky et al. 2019). We showed that a model with exemplar propensity makes more rule-based extrapolations for certain training feature combinations (*i.e.*, zero shot vs. partial exposure). Investigating how feature coverage impacts causal generalization is a fruitful future direction.

Similarity-based generalization and kernels. We use similarity-based kernels (*e.g.*, radial basis function (RBF)) to exemplify exemplar-based extrapolation. Recent work has interpreted neural networks as kernel regression (Jacot et al. 2018). Using a kernel framing to formalize the causes of exemplar bias is an exciting future direction.

Data augmentation. The EvR measure allows us to demonstrate that increased data variation in the form of feature coverage worsens systematic generalization. The negative effect of data variation on generalization has been documented for adversarial augmentations (Raghunathan et al. 2020). We show that this can persist even when augmentation is not adversarial, rendering it generally relevant for the design of data augmentations.

3.8 Conclusions

Taking inspiration from—and going beyond—psychological studies, we design a behavioral protocol to distinguish the effects of two inductive biases (feature-level bias and exemplar bias) that is easily applicable to any classification domain. This follows in a promising line of recent work that analyses and interprets deep learning systems based on their external behavior (Ritter, Barrett, et al. 2017; Dasgupta et al. 2020). It complements other approaches that follow in the neuroscience tradition of analyzing internal representations (Zeiler and Fergus 2014; Karpathy et al. 2015) or make approximations of these internal workings to support theoretical results (Jacot et al. 2018; Allen-Zhu et al. 2019). The behavioral approach has the advantage that it makes no assumptions about the model, allowing comparisons across systems that differ in design.

Both rule- and exemplar-based extrapolation are valuable depending on domain, underscoring the importance of diagnosing feature-level bias and exemplar bias. Moreover, studying this trade-off allows us to demonstrate an important phenomenon: We find that more feature coverage (as in partial exposure compared to zero shot) hurts generalization for exemplar-based models. This has implications for methods that manipulate data distributions to improve performance (*e.g.*, data subsampling (Haixiang et al. 2017), data augmentation (Perez and Wang 2017), and contrastive learning (Chen et al. 2020)). Since an exemplar-based model tends to acquire spurious associations, our measures have the potential to be useful as diagnostics in application settings where the goal is to control model behavior on non-representative factors (*e.g.*, Mitchell et al. 2019).

A limitation of the present work is that we do not provide a conclusive answer as to what properties of a model family influence both feature-level bias and exemplar bias. A broader study of these factors and theoretical work formalizing this effect are exciting avenues for future work.

Part II

Hierarchical modeling

Chapter 4

Recasting meta-learning as hierarchical Bayes

The work described in this chapter is published as Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths (2018). “Recasting gradient-based meta-learning as hierarchical Bayes”. In: *Proceedings of the International Conference on Learning Representations*.

4.1 Introduction

A remarkable aspect of human intelligence is the ability to quickly solve a novel problem and to be able to do so even in the face of limited experience in a novel domain. Such fast adaptation is made possible by leveraging prior learning experience in order to improve the efficiency of later learning. This capacity for *meta-learning* also has the potential to enable an artificially intelligent agent to learn more efficiently in situations with little available data or limited computational resources (Schmidhuber 1987; Bengio, Bengio, and Cloutier 1991; Naik and Mammone 1992).

In machine learning, meta-learning is formulated as the extraction of domain-general information that can act as an inductive bias to improve learning efficiency in novel tasks (Caruana 1998; Thrun and Pratt 1998). This inductive bias has been implemented in various ways: as learned hyperparameters in a hierarchical Bayesian model that regularize task-specific parameters (Heskes 1998), as a learned metric space in which to group neighbors (Bottou and Vapnik 1992), as a trained recurrent neural network that allows encoding and retrieval of episodic information (Santoro et al. 2016), or as an optimization algorithm with learned parameters (Schmidhuber 1987; Bengio, Bengio, Cloutier, and Gecsei 1992).

The model-agnostic meta-learning (MAML) algorithm of Finn, Abbeel, et al. (2017) is an instance of a learned optimization procedure that directly optimizes the standard gradient descent rule. The algorithm estimates an initial parameter set to be shared among the task-specific models; the intuition is that gradient descent from the learned initialization provides a favorable inductive bias for fast adaptation. However, this inductive bias has been evaluated only empirically in prior work (Finn, Abbeel, et al. 2017).

In this work, we present a novel derivation of and a novel extension to MAML, illustrating that this algorithm can be understood as inference for the parameters of a prior distribution in a hierarchical Bayesian model. The learned prior allows for quick adaptation to unseen tasks on the basis of an implicit predictive density over task-specific parameters. The reinterpretation as hierarchical Bayes gives a principled statistical motivation for MAML as a meta-learning algorithm, and sheds light on the reasons for its favorable performance even among methods with significantly more parameters. More importantly, by casting gradient-based meta-learning within a Bayesian framework, we are able to improve MAML by taking insights from Bayesian posterior estimation as novel augmentations to the gradient-based meta-learning procedure. We experimentally demonstrate that this enables better performance on a few-shot learning benchmark.

4.2 Meta-learning formulation

The goal of a meta-learner is to extract task-general knowledge through the experience of solving a number of related tasks. By using this learned prior knowledge, the learner has

the potential to quickly adapt to novel tasks even in the face of limited data or limited computation time.

Formally, we consider a dataset \mathcal{D} that defines a distribution over a family of tasks \mathcal{T} . These tasks share some common structure such that learning to solve a single task has the potential to aid in solving another. Each task \mathcal{T} defines a distribution over data points \mathbf{x} , which we assume in this work to consist of inputs and either regression targets or classification labels \mathbf{y} in a supervised learning problem (although this assumption can be relaxed to include reinforcement learning problems; *e.g.*, see Finn, Abbeel, et al. 2017). The objective of the meta-learner is to be able to minimize a task-specific performance metric associated with any given unseen task from the dataset given even only a small amount of data from the task; *i.e.*, to be capable of fast adaptation to a novel task.

In the following subsections, we discuss two ways of formulating a solution to the meta-learning problem: gradient-based hyperparameter optimization and probabilistic inference in a hierarchical Bayesian model. These approaches were developed orthogonally, but, in Section 4.3, we draw a novel connection between the two.

Meta-learning as gradient-based hyperparameter optimization

A parametric meta-learner aims to find some shared parameters θ that make it easier to find the right task-specific parameters ϕ when faced with a novel task. A variety of meta-learners that employ gradient methods for task-specific fast adaptation have been proposed (Andrychowicz et al. 2016; Li and Malik 2017a; Li and Malik 2017b; Wichrowska et al. 2017). MAML (Finn, Abbeel, et al. 2017) is distinct in that it provides a gradient-based meta-learning procedure that employs a single additional parameter (the meta-learning rate) and operates on the same parameter space for both meta-learning and fast adaptation. These are necessary features for the equivalence we show in Section 4.3.

To address the meta-learning problem, MAML estimates the parameters θ of a set of models so that when one or a few batch gradient descent steps are taken from the initialization at θ given a small sample of task data $\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_N} \sim p_{\mathcal{T}_j}(\mathbf{x})$ each model has good generalization performance on another sample $\mathbf{x}_{j_{N+1}}, \dots, \mathbf{x}_{j_{N+M}} \sim p_{\mathcal{T}_j}(\mathbf{x})$ from the same task. The MAML objective in a maximum likelihood setting is

$$\mathcal{L}(\theta) = \frac{1}{J} \sum_j \left[\frac{1}{M} \sum_m -\log p(\mathbf{x}_{j_{N+m}} | \underbrace{\theta - \alpha \nabla_{\theta} \frac{1}{N} \sum_n -\log p(\mathbf{x}_{j_n} | \theta)}_{\phi_j}) \right] \quad (4.1)$$

where we use ϕ_j to denote the updated parameters after taking a single batch gradient descent step from the initialization at θ with step size α on the negative log-likelihood associated with the task \mathcal{T}_j . Note that since ϕ_j is an iterate of a gradient descent procedure that starts from θ , each ϕ_j is of the same dimensionality as θ . We refer to the inner gradient descent procedure that computes ϕ_j as *fast adaptation*. The computational graph of MAML is given in Figure 4.1 (left).

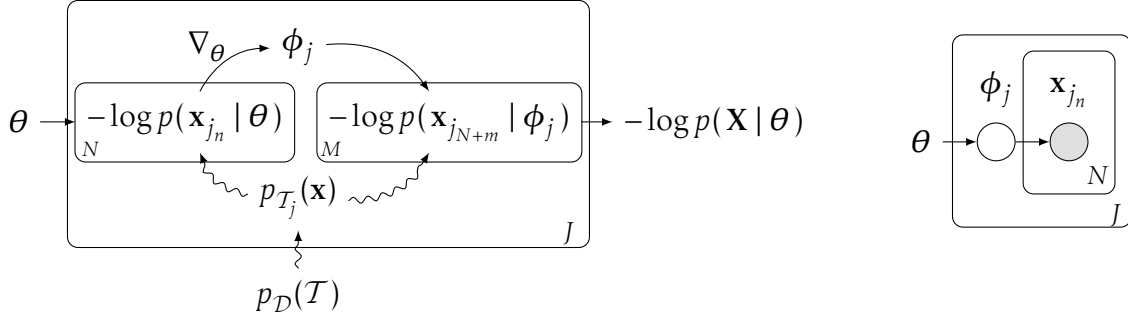


Figure 4.1: (Left) The computational graph of the MAML algorithm covered in Section 4.2. Straight arrows denote deterministic computations and crooked arrows denote sampling operations. (Right) The probabilistic graphical model for which MAML provides a parameter estimation procedure as described in Section 4.3. In each figure, plates denote repeated computations (left) or factorization (right) across independent and identically distributed samples.

Meta-learning as hierarchical Bayesian inference

An alternative way to formulate meta-learning is as a problem of probabilistic inference in the hierarchical model depicted in Figure 4.1 (right). In particular, in the case of meta-learning, each task-specific parameter ϕ_j is distinct from but should influence the estimation of the parameters $\{\phi_{j'} \mid j' \neq j\}$ from other tasks. We can capture this intuition by introducing a meta-level parameter θ on which each task-specific parameter is statistically dependent. With this formulation, the mutual dependence of the task-specific parameters ϕ_j is realized only through their individual dependence on the meta-level parameters θ . As such, estimating θ provides a way to constrain the estimation of each of the ϕ_j .

Given some data in a multi-task setting, we may estimate θ by integrating out the task-specific parameters to form the marginal likelihood of the data. Formally, grouping all of the data from each of the tasks as \mathbf{X} and again denoting by $\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_N}$ a sample from task \mathcal{T}_j , the marginal likelihood of the observed data is given by

$$p(\mathbf{X} \mid \theta) = \prod_j \left(\int p(\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_N} \mid \phi_j) p(\phi_j \mid \theta) d\phi_j \right). \quad (4.2)$$

Maximizing (4.2) as a function of θ gives a point estimate for θ , an instance of a method known as empirical Bayes (Bernardo and Smith 2006; Gelman et al. 2014) due to its use of the data to estimate the parameters of the prior distribution.

Hierarchical Bayesian models have a long history of use in both transfer learning and domain adaptation (e.g., Lawrence and Platt 2004; Yu et al. 2005; Gao et al. 2008; Daumé III 2009; Wan et al. 2012). However, the formulation of meta-learning as hierarchical Bayes does not automatically provide an inference procedure, and furthermore, there is no guarantee that inference is tractable for expressive models with many parameters such as deep neural networks.

Algorithm MAML-HB(\mathcal{D})

```

Initialize  $\theta$  randomly
while not converged do
  Draw  $J$  samples  $\mathcal{T}_1, \dots, \mathcal{T}_J \sim p_{\mathcal{D}}(\mathcal{T})$ 
  Estimate  $\mathbb{E}_{\mathbf{x} \sim p_{\mathcal{T}_1}(\mathbf{x})}[-\log p(\mathbf{x} | \theta)], \dots, \mathbb{E}_{\mathbf{x} \sim p_{\mathcal{T}_J}(\mathbf{x})}[-\log p(\mathbf{x} | \theta)]$  using ML-...
  Update  $\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_j \mathbb{E}_{\mathbf{x} \sim p_{\mathcal{T}_j}(\mathbf{x})}[-\log p(\mathbf{x} | \theta)]$ 
end

```

Algorithm 4.2: Model-agnostic meta-learning as hierarchical Bayesian inference. The choices of the subroutine ML-... that we consider are defined in Subroutine 4.3 and Subroutine 4.4.

4.3 Linking gradient-based meta-learning & hierarchical Bayes

In this section, we connect the two independent approaches of Section 4.2 and Section 4.2 by showing that MAML can be understood as empirical Bayes in a hierarchical probabilistic model. Furthermore, we build on this understanding by showing that a choice of update rule for the task-specific parameters ϕ_j (*i.e.*, a choice of inner-loop optimizer) corresponds to a choice of prior over task-specific parameters, $p(\phi_j | \theta)$.

Model-agnostic meta-learning as empirical Bayes

In general, when performing empirical Bayes, the marginalization over task-specific parameters ϕ_j in (4.2) is not tractable to compute exactly. To avoid this issue, we can consider an approximation that makes use of a point estimate $\hat{\phi}_j$ instead of performing the integration over ϕ in (4.2). Using $\hat{\phi}_j$ as an estimator for each ϕ_j , we may write the negative logarithm of the marginal likelihood as

$$-\log p(\mathbf{X} | \theta) \approx \sum_j \left[-\log p(\mathbf{x}_{j_{N+1}}, \dots, \mathbf{x}_{j_{N+M}} | \hat{\phi}_j) \right]. \quad (4.3)$$

Setting $\hat{\phi}_j = \theta + \alpha \nabla_{\theta} \log p(\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_N} | \theta)$ for each j in (4.3) recovers the unscaled form of the one-step MAML objective in (4.1). This tells us that the MAML objective is equivalent to a maximization with respect to the meta-level parameters θ of the marginal likelihood $p(\mathbf{X} | \theta)$, where a point estimate for each task-specific parameter ϕ_j is computed via one or a few steps of gradient descent. By taking only a few steps from the initialization at θ , the point estimate $\hat{\phi}_j$ trades off minimizing the fast adaptation objective $-\log p(\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_N} | \theta)$ with staying close in value to the parameter initialization θ .

We can formalize this trade-off by considering the linear regression case. Recall that the *maximum a posteriori* (MAP) estimate of ϕ_j corresponds to the global mode of the posterior $p(\phi_j | \mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_N}, \theta) \propto p(\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_N} | \phi_j) p(\phi_j | \theta)$. In the case of a linear model, early stopping of an iterative gradient descent procedure to estimate ϕ_j is exactly

Subroutine ML-POINT(θ, \mathcal{T})
 Draw N samples $\mathbf{x}_1, \dots, \mathbf{x}_N \sim p_{\mathcal{T}}(\mathbf{x})$
 Initialize $\phi \leftarrow \theta$
for k in $1, \dots, K$ **do**
 | Update $\phi \leftarrow \phi + \alpha \nabla_{\phi} \log p(\mathbf{x}_1, \dots, \mathbf{x}_N | \phi)$
end
 Draw M samples $\mathbf{x}_{N+1}, \dots, \mathbf{x}_{N+M} \sim p_{\mathcal{T}}(\mathbf{x})$
return $-\log p(\mathbf{x}_{N+1}, \dots, \mathbf{x}_{N+M} | \phi)$

Subroutine 4.3: Subroutine for computing a point estimate $\hat{\phi}$ using truncated gradient descent to approximate the marginal negative log likelihood (NLL).

equivalent to MAP estimation of ϕ_j under the assumption of a prior that depends on the number of descent steps as well as the direction in which each step is taken. In particular, write the input examples as \mathbf{X} and the vector of regression targets as \mathbf{y} , omit the task index from ϕ , and consider the gradient descent update

$$\begin{aligned} \phi^{(k)} &= \phi^{(k-1)} - \alpha \nabla_{\phi} \left[\|\mathbf{y} - \mathbf{X}\phi\|_2^2 \right]_{\phi=\phi^{(k-1)}} \\ &= \phi^{(k-1)} - \alpha \mathbf{X}^T (\mathbf{X}\phi^{(k-1)} - \mathbf{y}) \end{aligned} \quad (4.4)$$

for iteration index k and learning rate $\alpha \in \mathbb{R}^+$. Santos (1996) shows that, starting from $\phi_{(0)} = \theta$, $\phi_{(k)}$ in (4.4) solves the regularized linear least squares problem

$$\min \left(\|\mathbf{y} - \mathbf{X}\phi\|_2^2 + \|\theta - \phi\|_{\mathbf{Q}}^2 \right) \quad (4.5)$$

with \mathbf{Q} -norm defined by $\|\mathbf{z}\|_{\mathbf{Q}} = \mathbf{z}^T \mathbf{Q}^{-1} \mathbf{z}$ for a symmetric positive definite matrix \mathbf{Q} that depends on the step size α and iteration index k as well as on the covariance structure of \mathbf{X} . We describe the exact form of the dependence in Section 4.3. The minimization in (4.5) can be expressed as a posterior maximization problem given a conditional Gaussian likelihood over \mathbf{y} and a Gaussian prior over ϕ . The posterior takes the form

$$p(\phi | \mathbf{X}, \mathbf{y}, \theta) \propto \mathcal{N}(\mathbf{y}; \mathbf{X}\phi, \mathbb{I}) \mathcal{N}(\phi; \theta, \mathbf{Q}) . \quad (4.6)$$

Since $\phi_{(k)}$ in (4.4) maximizes (4.6), we may conclude that k iterations of gradient descent in a linear regression model with squared error exactly computes the MAP estimate of ϕ , given a Gaussian-noised observation model and a Gaussian prior over ϕ with parameters $\mu_0 = \theta$ and $\Sigma_0 = \mathbf{Q}$. Therefore, in the case of linear regression with squared error, MAML is exactly empirical Bayes using the MAP estimate as the point estimate of ϕ .

In the nonlinear case, MAML is again equivalent to an empirical Bayes procedure to maximize the marginal likelihood that uses a point estimate for ϕ computed by one or a few steps of gradient descent. However, this point estimate is not necessarily the global

mode of a posterior. We can instead understand the point estimate given by truncated gradient descent as the value of the mode of an implicit posterior over ϕ resulting from an empirical loss interpreted as a negative log-likelihood, and regularization penalties and the early stopping procedure jointly acting as priors Sjöberg and Ljung (for similar interpretations, see 1995), Bishop (1995), and Duvenaud, Maclaurin, et al. (2016).

The exact equivalence between early stopping and a Gaussian prior on the weights in the linear case, as well as the implicit regularization to the parameter initialization the nonlinear case, tells us that every iterate of truncated gradient descent is a mode of an implicit posterior. In particular, we are not required to take the gradient descent procedure of fast adaptation that computes $\hat{\phi}$ to convergence in order to establish a connection between MAML and hierarchical Bayes. MAML can therefore be understood to approximate an expectation of the marginal negative log likelihood (NLL) for each task \mathcal{T}_j as

$$\mathbb{E}_{\mathbf{x} \sim p_{\mathcal{T}_j}(\mathbf{x})}[-\log p(\mathbf{x} | \theta)] \approx \frac{1}{M} \sum_m -\log p(\mathbf{x}_{j_{N+m}} | \hat{\phi}_j)$$

using the point estimate $\hat{\phi}_j = \theta + \alpha \nabla_{\theta} \log p(\mathbf{x}_{j_n} | \theta)$ for single-step fast adaptation.

The algorithm for MAML as probabilistic inference is given in Algorithm 4.2; Subroutine 4.3 computes each marginal NLL using the point estimate of $\hat{\phi}$ as just described. Formulating MAML in this way, as probabilistic inference in a hierarchical Bayesian model, motivates the interpretation in Section 4.3 of using various meta-optimization algorithms to induce a prior over task-specific parameters.

The prior over task-specific parameters

From Section 4.3, we may conclude that early stopping during fast adaptation is equivalent to a specific choice of a prior over task-specific parameters, $p(\phi_j | \theta)$. We can better understand the role of early stopping in defining the task-specific parameter prior in the case of a quadratic objective. Omit the task index from ϕ and \mathbf{x} , and consider a second-order approximation of the fast adaptation objective $\ell(\phi) = -\log p(\mathbf{x}_1, \dots, \mathbf{x}_N | \phi)$ about a minimum ϕ^* :

$$\ell(\phi) \approx \tilde{\ell}(\phi) := \frac{1}{2} \|\phi - \phi^*\|_{\mathbf{H}^{-1}}^2 + \ell(\phi^*) \quad (4.7)$$

where the Hessian $\mathbf{H} = \nabla_{\phi}^2 \ell(\phi^*)$ is assumed to be positive definite so that $\tilde{\ell}$ is bounded below. Furthermore, consider using a curvature matrix \mathcal{B} to precondition the gradient in gradient descent, giving the update

$$\phi_{(k)} = \phi_{(k-1)} - \mathcal{B} \nabla_{\phi} \tilde{\ell}(\phi_{(k-1)}) . \quad (4.8)$$

If \mathcal{B} is diagonal, we can identify (4.8) as a Newton method with a diagonal approximation to the inverse Hessian; using the inverse Hessian evaluated at the point $\phi_{(k-1)}$ recovers Newton’s method itself. On the other hand, meta-learning the matrix \mathcal{B} matrix via gradient descent provides a method to incorporate task-general information into the covariance of the fast adaptation prior, $p(\phi | \theta)$. For instance, the meta-learned matrix \mathcal{B} may encode correlations between parameters that dictates how such parameters are updated relative to each other.

Formally, taking k steps of gradient descent from $\phi_{(0)} = \theta$ using the update rule in (4.8) gives a $\phi_{(k)}$ that solves

$$\min \left(\|\phi - \phi^*\|_{\mathbf{H}^{-1}}^2 + \|\phi_{(0)} - \phi\|_{\mathbf{Q}}^2 \right) . \quad (4.9)$$

The minimization in (4.9) corresponds to taking a Gaussian prior $p(\phi | \theta)$ with mean θ and covariance \mathbf{Q} for $\mathbf{Q} = \mathbf{O}\mathbf{\Lambda}^{-1}((\mathbb{I} - \mathbf{B}\mathbf{\Lambda})^{-k} - \mathbb{I})\mathbf{O}^T$ (Santos 1996) where \mathbf{B} is a diagonal matrix that results from a simultaneous diagonalization of \mathbf{H} and \mathcal{B} as $\mathbf{O}^T\mathbf{H}\mathbf{O} = \text{diag}(\lambda_1, \dots, \lambda_n) = \mathbf{\Lambda}$ and $\mathbf{O}^T\mathcal{B}^{-1}\mathbf{O} = \text{diag}(b_1, \dots, b_n) = \mathbf{B}$ with $b_i, \lambda_i \geq 0$ for $i = 1, \dots, n$ Golub and Van Loan (Theorem 8.7.1 in 1983). If the true objective is indeed quadratic, then, assuming the data is centered, \mathbf{H} is the unscaled covariance matrix of features, $\mathbf{X}^T\mathbf{X}$.

4.4 Improving model-agnostic meta-learning

Identifying MAML as a method for probabilistic inference in a hierarchical model allows us to develop novel improvements to the algorithm. In Section 4.4, we consider an approach from Bayesian parameter estimation to improve the MAML algorithm, and in Section 4.4, we discuss how to make this procedure computationally tractable for high-dimensional models.

Laplace’s method of integration

We have shown that the MAML algorithm is an empirical Bayes procedure that employs a point estimate for the mid-level, task-specific parameters in a hierarchical Bayesian model. However, the use of this point estimate may lead to an inaccurate point approximation of the integral in (4.2) if the posterior over the task-specific parameters, $p(\phi_j | \mathbf{x}_{j_{N+1}}, \dots, \mathbf{x}_{j_{N+M}}, \theta)$, is not sharply peaked at the value of the point estimate. The Laplace approximation (Laplace 1986; MacKay 1992b; MacKay 1992a) is applicable in this case as it replaces a point estimate of an integral with the volume of a Gaussian centered at a mode of the integrand, thereby forming a local quadratic approximation.

We can make use of this approximation to incorporate uncertainty about the task-specific parameters into the MAML algorithm at fast adaptation time. In particular, suppose that each integrand in (4.2) has a mode ϕ_j^* at which it is locally well-approximated by a quadratic function. The Laplace approximation uses a second-order Taylor expansion

Subroutine ML-LAPLACE (θ, \mathcal{T})

```

  Draw  $N$  samples  $\mathbf{x}_1, \dots, \mathbf{x}_N \sim p_{\mathcal{T}}(\mathbf{x})$ 
  Initialize  $\phi \leftarrow \theta$ 
  for  $k$  in  $1, \dots, K$  do
    | Update  $\phi \leftarrow \phi + \alpha \nabla_{\phi} \log p(\mathbf{x}_1, \dots, \mathbf{x}_N | \phi)$ 
  end
  Draw  $M$  samples  $\mathbf{x}_{N+1}, \dots, \mathbf{x}_{N+M} \sim p_{\mathcal{T}}(\mathbf{x})$ 
  Estimate quadratic curvature  $\hat{\mathbf{H}}$ 
  return  $-\log p(\mathbf{x}_{N+1}, \dots, \mathbf{x}_{N+M} | \phi) + \eta \log \det(\hat{\mathbf{H}})$ 

```

Subroutine 4.4: Subroutine for computing a Laplace approximation of the marginal likelihood.

of the negative log posterior in order to approximate each integral in the product in (4.2) as

$$\int p(\mathbf{X}_j | \phi_j) p(\phi_j | \theta) d\phi_j \approx p(\mathbf{X}_j | \phi_j^*) p(\phi_j^* | \theta) \det(\mathbf{H}_j / 2\pi)^{-\frac{1}{2}} \quad (4.10)$$

where \mathbf{H}_j is the Hessian matrix of second derivatives of the negative log posterior.

Classically, the Laplace approximation uses the MAP estimate for ϕ_j^* , although any mode can be used as an expansion site provided the integrand is well enough approximated there by a quadratic. We use the point estimate $\hat{\phi}_j$ uncovered by fast adaptation, in which case the MAML objective in (4.1) becomes an appropriately scaled version of the approximate marginal likelihood

$$-\log p(\mathbf{X} | \theta) \approx \sum_j \left[-\log p(\mathbf{X}_j | \hat{\phi}_j) - \log p(\hat{\phi}_j | \theta) + \frac{1}{2} \log \det(\mathbf{H}_j) \right]. \quad (4.11)$$

The term $\log p(\hat{\phi}_j | \theta)$ results from the implicit regularization imposed by early stopping during fast adaptation, as discussed in Section 4.3. The term $\frac{1}{2} \log \det(\mathbf{H}_j)$, on the other hand, results from the Laplace approximation and can be interpreted as a form of regularization that penalizes model complexity.

Using curvature information

Using (4.11) as a training criterion for a neural network model is difficult due to the required computation of the determinant of the Hessian of the log posterior \mathbf{H}_j , which itself decomposes into a sum of the Hessian of the log likelihood and the Hessian of the log prior as

$$\mathbf{H}_j = \nabla_{\phi_j}^2 \left[-\log p(\mathbf{X}_j | \phi_j) \right] + \nabla_{\phi_j}^2 \left[-\log p(\phi_j | \theta) \right].$$

In our case of early stopping as regularization, the prior over task-specific parameters $p(\phi_j | \theta)$ is implicit and thus no closed form is available for a general model. Although we may use the quadratic approximation derived in Section 4.3 to obtain an approximate Gaussian prior, this prior is not diagonal and does not, to our knowledge, have a convenient factorization. Therefore, in our experiments, we instead use a simple approximation in which the prior is approximated as a diagonal Gaussian with precision τ . We keep τ fixed, although this parameter may be cross-validated for improved performance.

Similarly, the Hessian of the log likelihood is intractable to form exactly for all but the smallest models, and furthermore, is not guaranteed to be positive definite at all points, possibly rendering the Laplace approximation undefined. To combat this, we instead seek a curvature matrix $\hat{\mathbf{H}}$ that approximates the quadratic curvature of a neural network objective function. Since it is well-known that the curvature associated with neural network objective functions is highly non-diagonal (e.g., Martens 2016), a further requirement is that the matrix have off-diagonal terms.

Due to the difficulties listed above, we turn to second order gradient descent methods, which precondition the gradient with an inverse curvature matrix at each iteration of descent. The Fisher information matrix (Fisher 1925) has been extensively used as an approximation of curvature, giving rise to a method known as natural gradient descent (Amari 1998). A neural network with an appropriate choice of loss function is a probabilistic model and therefore defines a Fisher information matrix. Furthermore, the Fisher information matrix can be seen to define a convex quadratic approximation to the objective function of a probabilistic neural model (Pascanu and Bengio 2014; Martens 2014). Importantly for our use case, the Fisher information matrix is positive definite by definition as well as non-diagonal.

However, the Fisher information matrix is still expensive to work with. Martens and Grosse (2015) developed Kronecker-factored approximate curvature (K-FAC), a scheme for approximating the curvature of the objective function of a neural network with a block-diagonal approximation to the Fisher information matrix. Each block corresponds to a unique layer in the network, and each block is further approximated as a Kronecker product (see Van Loan 2000) of two much smaller matrices by assuming that the second-order statistics of the input activation and the back-propagated derivatives within a layer are independent. These two approximations ensure that the inverse of the Fisher information matrix can be computed efficiently for the natural gradient.

For the Laplace approximation, we are interested in the determinant of a curvature matrix instead of its inverse. However, we may also make use of the approximations to the Fisher information matrix from K-FAC as well as properties of the Kronecker product. In particular, we use the fact that the determinant of a Kronecker product is the product of the exponentiated determinants of each of the factors, and that the determinant of a block diagonal matrix is the product of the determinants of the blocks (Van Loan 2000). The determinants for each factor can be computed as efficiently as the inverses required by K-FAC, in $\mathcal{O}(d^3)$ time for a d -dimensional Kronecker factor.

We make use of the Laplace approximation and K-FAC to replace Subroutine 4.3,

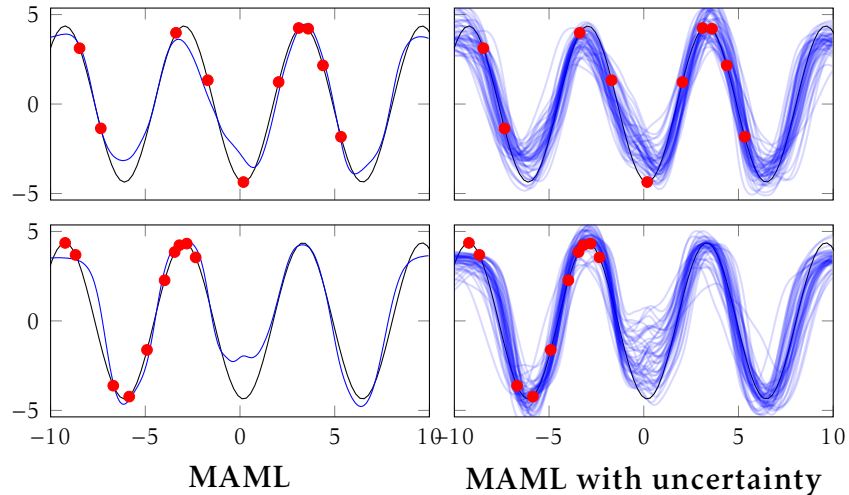


Figure 4.5: Our method is able to meta-learn a model that can quickly adapt to sinusoids with varying phases and amplitudes, and the interpretation of the method as hierarchical Bayes makes it practical to directly sample models from the posterior. In this figure, we illustrate various samples from the posterior of a model that is meta-trained on different sinusoids, when presented with a few datapoints (in red) from a new, previously unseen sinusoid. Note that the random samples from the posterior predictive describe a distribution of functions that are all sinusoidal and that there is increased uncertainty when the datapoints are less informative (*i.e.*, when the datapoints are sampled only from the lower part of the range input, shown in the bottom-right example).

which computes the task-specific marginal NLLs using a point estimate for $\hat{\phi}$. We call this method the Lightweight Laplace Approximation for Meta-Adaptation (LLAMA), and give a replacement subroutine in Subroutine 4.4.

4.5 Experimental evaluation

The goal of our experiments is to evaluate if we can use our probabilistic interpretation of MAML to generate samples from the distribution over adapted parameters, and furthermore, if our method can be applied to large-scale meta-learning problems such as *miniImageNet*.

Warmup: Toy nonlinear model

The connection between MAML and hierarchical Bayes suggests that we should expect MAML to behave like an algorithm that learns the mean of a Gaussian prior on model parameters, and uses the mean of this prior as an initialization during fast adaptation. Using the Laplace approximation to the integration over task-specific parameters as in (4.10) assumes a task-specific parameter posterior with mean at the adapted parameters $\hat{\phi}$ and covariance equal to the inverse Hessian of the log posterior evaluated at the adapted

Model	5-way acc. (%)	
	1-shot	
Fine-tuning*	28.86	± 0.54
Nearest Neighbor*	41.08	± 0.70
Matching Networks FCE (Vinyals et al. 2016)*	43.56	± 0.84
Meta-Learner LSTM (Ravi and Larochelle 2017)*	43.44	± 0.77
SNAIL (Mishra et al. 2018)**	45.1	± —
Prototypical Networks (Snell et al. 2017)**	46.61	± 0.78
mAP-DLM (Triantafillou et al. 2017)	49.82	± 0.78
MAML (Finn, Abbeel, et al. 2017)	48.70	± 1.84
LLAMA (Ours)	49.40	± 1.83

Table 4.1: One-shot classification performance on the *miniImageNet* test set, with comparison methods ordered by one-shot performance. All results are averaged over 600 test episodes, and we report 95% confidence intervals.

parameter value. Instead of simply using this density in the Laplace approximation as an additional regularization term as in (4.11), we may sample parameters ϕ_j from this density and use each set of sampled parameters to form a set of predictions for a given task.

To illustrate this relationship between MAML and hierarchical Bayes, we present a meta-dataset of sinusoid tasks in which each task involves regressing to the output of a sinusoid wave in Figure 4.5. Variation between tasks is obtained by sampling the amplitude uniformly from $[0.1, 5.0]$ and the phase from $[0, \pi]$. During training and for each task, 10 input datapoints are sampled uniformly from $[-10.0, 10.0]$ and the loss is the mean squared error between the prediction and the true value.

We observe in Figure 4.5 that our method allows us to directly sample models from the task-specific parameter distribution after being presented with 10 datapoints from a new, previously unseen sinusoid curve. In particular, the column on the right of Figure 4.5 demonstrates that the sampled models display an appropriate level of uncertainty when the datapoints are ambiguous (as in the bottom right).

Large-scale experiment: *miniImageNet*

We evaluate LLAMA on the *miniImageNet* Ravi and Larochelle (2017) 1-shot, 5-way classification task, a standard benchmark in few-shot classification. *miniImageNet* comprises 64 training classes, 12 validation classes, and 24 test classes. Following the setup of Vinyals et al. (2016), we structure the N -shot, J -way classification task as follows: The model observes N instances of J unseen classes, and is evaluated on its ability to classify M new instances within the J classes.

*Results reported by Ravi and Larochelle (2017). **We report test accuracy for a comparable architecture.

***We report test accuracy for models matching train and test “shot” and “way”.

We use a neural network architecture standard to few-shot classification Vinyals et al. (e.g., 2016) and Ravi and Larochelle (2017), consisting of 4 layers with 3×3 convolutions and 64 filters, followed by batch normalization (BN) (Ioffe and Szegedy 2015), a ReLU nonlinearity, and 2×2 max-pooling. For the scaling variable β and centering variable γ of BN (see Ioffe and Szegedy 2015), we ignore the fast adaptation update as well as the Fisher factors for K-FAC. We use Adam (Kingma and Ba 2014) as the meta-optimizer, and standard batch gradient descent with a fixed learning rate to update the model during fast adaptation. LLAMA requires the prior precision term τ as well as an additional parameter $\eta \in \mathbb{R}^+$ that weights the regularization term $\log \det \hat{\mathbf{H}}$ contributed by the Laplace approximation. We fix $\tau = 0.001$ and selected $\eta = 10^{-6}$ via cross-validation; all other parameters are set to the values reported in Finn, Abbeel, et al. (2017).

We find that LLAMA is practical enough to be applied to this larger-scale problem. In particular, our TensorFlow implementation of LLAMA trains for 60,000 iterations on one TITAN Xp GPU in 9 hours, compared to 5 hours to train MAML. As shown in Table 4.1, LLAMA achieves comparable performance to the state-of-the-art meta-learning method by Triantafillou et al. (2017). While the gap between MAML and LLAMA is small, the improvement from the Laplace approximation suggests that a more accurate approximation to the marginalization over task-specific parameters will lead to further improvements.

4.6 Related work

Marginal likelihood (ML) and few-shot learning have a long history in hierarchical Bayesian modeling (e.g., Tenenbaum 1999; Fei-Fei et al. 2003; Lawrence and Platt 2004; Yu et al. 2005; Gao et al. 2008; Daumé III 2009; Wan et al. 2012). A related subfield is that of transfer learning, which has used hierarchical Bayes extensively (e.g., Raina et al. 2006). A variety of inference methods have been used in Bayesian models, including exact inference (Lake, Salakhutdinov, Gross, et al. 2011), sampling methods (Salakhutdinov et al. 2012), and variational methods (Edwards and Storkey 2017). While some prior works on hierarchical Bayesian models have proposed to handle basic image recognition tasks, the complexity of these tasks does not yet approach the kinds of complex image recognition problems that can be solved by discriminatively trained deep networks, such as the *mini*ImageNet experiment in our evaluation (Mansinghka et al. 2013).

Recently, the Omniglot benchmark Lake, Ullman, et al. (2018) has rekindled interest in the problem of learning from few examples. Modern methods accomplish few-shot learning either through the design of network architectures that ingest the few-shot training samples directly (e.g., Koch 2015; Vinyals et al. 2016; Snell et al. 2017; Hariharan and Girshick 2017; Triantafillou et al. 2017), or formulating the problem as one of *learning to learn*, or *meta-learning* (e.g., Schmidhuber 1987; Bengio, Bengio, and Cloutier 1991; Schmidhuber 1992; Bengio, Bengio, Cloutier, and Gecsei 1992). A variety of inference methods have been used in Bayesian models, including exact inference (Lake, Salakhutdi-

nov, Gross, et al. 2011), sampling methods (Salakhutdinov et al. 2013), and variational methods (Edwards and Storkey 2017).

Our work bridges the gap between gradient-based meta-learning methods and hierarchical Bayesian modeling. Our contribution is not to formulate the meta-learning problem as a hierarchical Bayesian model, but instead to formulate a gradient-based meta-learner as hierarchical Bayesian inference, thus providing a way to efficiently perform posterior inference in a model-agnostic manner.

4.7 Conclusion

We have shown that model-agnostic meta-learning (MAML) estimates the parameters of a prior in a hierarchical Bayesian model. By casting gradient-based meta-learning within a Bayesian framework, our analysis opens the door to novel improvements inspired by probabilistic machinery.

As a step in this direction, we propose an extension to MAML that employs a Laplace approximation to the posterior distribution over task-specific parameters. This technique provides a more accurate estimate of the integral that, in the original MAML algorithm, is approximated via a point estimate. We show how to estimate the quantity required by the Laplace approximation using Kronecker-factored approximate curvature (K-FAC), a method recently proposed to approximate the quadratic curvature of a neural network objective for the purpose of a second-order gradient descent technique.

Our contribution illuminates the road to exploring further connections between gradient-based meta-learning methods and hierarchical Bayesian modeling. For instance, in this work we assume that the predictive distribution over new data-points is narrow and well-approximated by a point estimate. We may instead employ methods that make use of the variance of the distribution over task-specific parameters in order to model the predictive density over examples from a novel task.

Furthermore, it is known that the Laplace approximation is inaccurate in cases where the integral is highly skewed, or is not unimodal and thus is not amenable to approximation by a single Gaussian mode. This could be solved by using a finite mixture of Gaussians, which can approximate many density functions arbitrarily well (Sorenson and Alspach 1971; Alspach and Sorenson 1972). The exploration of additional improvements such as this is an exciting line of future work.

Chapter 5

Concept learning from few positive examples

The work described in this chapter is published as Erin Grant, Joshua C Peterson, and Thomas L Griffiths (2019). "Learning deep taxonomic priors for concept learning from few positive examples". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*.

5.1 Introduction

One of the hallmarks of human intelligence is the ability to rapidly learn new concepts given only limited information (Lake, Ullman, et al. 2018). This task is difficult because we are often presented with only a handful of (positive) examples of a new concept, and no examples outside of the concept (negative examples). Quine (1960) was the first to recognize that this poses a seemingly crippling problem for induction: hearing only the word “gavagai” as a rabbit passes by, we have no way of knowing with certainty whether the new word applies to all animals, all rabbits, one pet rabbit, potential food, or any other of a nearly infinite number of likewise compatible hypotheses.

Nevertheless, humans appear to possess prior knowledge, whether learned, innate, or both, that makes for effective generalizations even under such conditions. In some situations, these constraints are simple and easy to model (Tenenbaum 1999; Tenenbaum and Griffiths 2001; Kemp et al. 2007). However, in general, modeling the rich prior knowledge that humans bring to bear on problems in complex domains such as natural images is difficult and reliant on explicit domain knowledge (Xu and Tenenbaum 2007; Jia et al. 2013). A recent line of follow-up work has made strides by using deep neural networks as a proxy for psychological representations (Campero et al. 2017; Peterson, Soulos, et al. 2018). Although these representations are largely perceptual, they are nevertheless an improvement over hand-specified features given that they are less prone to experimenter bias and have been shown to explain some aspects of human visual representations (Peterson, Abbott, et al. 2018). However, unlike most cognitive models of concept learning and unlike humans, these networks are trained on millions of both positive and negative examples of mutually exclusive categories. Moreover, they fail to capture the taxonomic biases that humans bring to bear in concept learning (Peterson, Abbott, et al. 2018).

Challenged by the cognitive science community (Lake, Salakhutdinov, and Tenenbaum 2015), machine learning researchers have developed a number of their own improvements to deep learning algorithms to tackle the problem of learning from few examples (*e.g.*, Vinyals et al. 2016; Ravi and Larochelle 2017). These approaches constitute impressive new candidate accounts of human concept learning from naturalistic stimuli, but differ from human learning scenarios in that they (1) rely on negative evidence to infer the extent of a novel concept, and (2) ignore the overlapping and hierarchical structure of real-world concepts that humans use to inform their generalization judgments (Rosch et al. 1976; Xu and Tenenbaum 2007).

In the following paper, we aim to address many of the shortcomings of previous work by demonstrating how a deep meta-learning algorithm combined with a novel stimulus sampling procedure can provide an end-to-end framework for modeling human concept learning, for the first time with no hand-specified prior knowledge or negative examples of a novel concept. We introduce a new, taxonomically structured dataset of concepts compiled by sampling from both internal nodes and leaf nodes within the ImageNet hierarchy (Deng et al. 2009). Our method learns concepts at different levels of this

hierarchy, but the hierarchical structure itself is never provided to the model explicitly at any point. To evaluate our model against human behavior, we present a new human benchmark inspired by Rosch’s classic object taxonomies (Rosch et al. 1976). Our model not only mimics human generalization behavior, reproducing classic generalization gradients (Shepard 1987; Xu and Tenenbaum 2007), but also encompasses a general taxonomic prior that allows for human-like generalization even when presented with novel concepts from different image taxonomies (*i.e.*, held-out supercategories).

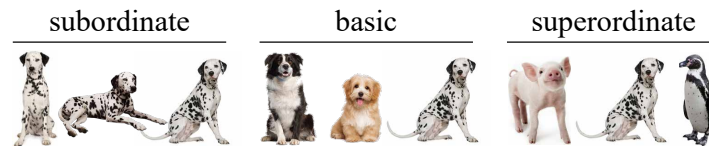
5.2 Background

Computational models of concept learning in cognitive science have historically focused on the problem of density estimation (Ashby and Alfonso-Reese 1995). Under this paradigm, learning about a category C amounts to the estimation of the density $p(x | C)$, where x represents the space of stimuli. This modeling framework assumes that a density can be learned for each of a set of mutually exclusive categories, where positive examples from one category implicitly serve as negative examples for all other categories. However, the conditions under which humans learn concepts are rarely this straightforward.

Learning concepts from few positive examples. More recent work has begun to examine how humans learn concepts in more natural settings where often only a few positive examples of a single concept are provided. Despite this impoverished learning environment, even young children are able to generalize surprisingly well (Carey 1978; Markman 1989). Extending Shepard (1987), Tenenbaum (1999) and Tenenbaum and Griffiths (2001) formalize the concept learning problem as follows: Given n positive examples $\mathbf{x} = \{x_1, \dots, x_n\}$ of a concept C , the learner estimates the probability $p(x^* \in C | \mathbf{x})$ that a new stimulus x^* is also an example of that concept. The challenge the learner faces in making such a generalization is that the extension of C is underspecified (*i.e.*, it could include only the present examples, all possible stimuli, or anything in between). To address this challenge, the authors propose a Bayesian generalization model that averages the predictions made by a number of hypotheses about the extent of C . By making the plausible assumption that learners expect examples to be randomly sampled from concepts, the authors show that smaller hypotheses will be preferred, thus deriving constraints on the expected extent of C .

Armed with this framework, Xu and Tenenbaum (2007) conducted an extensive analysis of human generalization behavior through *word learning* experiments. Participants were given either one or three examples of a new concept such as “dax” and asked to pick out other instances of that concept from a set of test stimuli. The examples of each concept were unique images that could be drawn from either a subordinate-level (*e.g.*, Dalmatian), basic-level (*e.g.*, dog), or superordinate-level (*e.g.*, animal) category, and the test stimuli were sampled from all three levels. An example of this task is shown in Figure 5.1. Replicating Shepard (1987), the authors found that generalization from a single example

Training Conditions - Possible examples of a *dax*



Test Phase - Pick everything that is a *dax*



Figure 5.1: The *word learning* paradigm from Xu and Tenenbaum (2007). In each trial, participants see a few instances exemplifying a novel word such as “*dax*” and are asked to select other instances that fall under the same word from a test array. The training conditions vary by the levels of the underlying image taxonomy from which the instances are drawn, *e.g.*, Dalmatians (subordinate) vs. dogs (basic) vs. animals (superordinate).

of a concept to a test stimulus decreases with psychological similarity. However, their experiments also yielded two new insights into human concept learning:

1. Given multiple examples of a concept, generalization goes only as far at the most specific level that contains those examples. For example, shown three examples from different dog breeds, other dog breeds are included in the concept at test time, but not other animals.
2. There is a bias towards generalizing to test items at the basic level, in particular when only a single subordinate example is shown. For example, given a single example of a Dalmatian, participants predictably generalize the concept to other Dalmatians, but also generalize to other breeds.

The only modification to the Bayesian concept learning model required to capture these data was a structured, taxonomic prior computed from human similarity judgments over the set of objects used in the experiments. While this work constitutes one of the first successful attempts to explain concept learning in realistic contexts, it arguably leaves much of the structured, taxonomic representation assumed and raises questions about how this knowledge is acquired.

The role of prior knowledge. Given the aforementioned dependence on highly structured priors in explaining people’s robust generalization behavior, subsequent work has focused on incorporating this information into the modeling of human concept learning.

Jia et al. (2013) provided an automated framework for modeling human generalization behavior by leveraging perceptual stimulus features provided by a computer vision algorithm along with information contained in the WordNet taxonomy (Fellbaum 1998), but gave no account for how this information is learned by humans. Kemp et al. (2007) provided the first account of how such knowledge could be acquired: The authors start with an unstructured representation and apply a structured hierarchical Bayesian model that learns taxonomic abstractions from data. Despite its elegance, the method does not immediately scale to high-dimensional stimuli such as the images used in Jia et al. (2013).

Deep neural networks (LeCun, Bengio, et al. 2015) have served as both candidate models of object perception and rich image representations that can be used for cognitive modeling. However, these model do not capture even coarse taxonomic information out-of-the-box (Peterson, Abbott, et al. 2018). Despite this, Peterson and Griffiths (2017) found that the sampling assumptions of Bayesian concept learning could be verified in human generalization judgments when modeling stimuli using deep feature representations. Campero et al. (2017) deployed a hierarchical model similar to Kemp et al. (2007) over a deep feature space and found both good one-shot learning performance as well as the ability to recover some stimulus clusters representative of human categorization judgments. Noting that most deep networks are trained using subordinate-level labels, (Peterson, Soulos, et al. 2018) trained a deep neural network with coarser, basic-level labels to more closely mimic the supervision children receive. A relatively simple generalization model over the resulting representation reproduced both the basic-level bias and the gradient of generalization from Xu and Tenenbaum (2007).

Few-shot learning in machine learning. The problem facing cognitive models of concept learning is closely related to *one-* or *few-shot* classification in machine learning, in which the aim is to learn to discriminate between classes given only a few labeled examples from each class (Fei-Fei et al. 2003; Vinyals et al. 2016). A powerful solution to few-shot learning is *meta-learning*, where learning episodes—themselves consisting of training and testing intervals—are used to train a model to adapt quickly to solve a new task given only a small amount of labeled data for the task (Schmidhuber 1987; Bengio, Bengio, and Cloutier 1991; Schmidhuber 1992; Bengio, Bengio, Cloutier, and Gecsei 1992). The learning episodes are leveraged in the form of a data-driven prior that is combined with a small amount of test-time evidence (*i.e.*, a few “shots” of data and their corresponding labels from a novel task) in order to make a test-time inference.

5.3 Modeling approach

We propose to bridge cognitive science and machine learning by formulating concept learning as a few-shot learning problem. As we will see, the meta-learning problem formulation allows a machine learning model to estimate a decision boundary from only positive samples of a class, similarly to how people learn concepts from only a few positive

Superordinate	Basic	Subordinates	
Musical Instrument	Guitar	Acoustic guitar	Electric guitar
	Piano	Grand piano	Upright piano
	Drum	Tambourine	Bass drum
Fruit	Apple	Delicious apple	Mackintosh apple
	Currant	Black currant	Red currant
	Grapes	Concord grapes	Thompson seedless grapes
Tool	Hammer	Ball-peen hammer	Carpenter's hammer
	Saw	Hack saw	Cross-cutting saw
	Screwdriver	Phillips screwdriver	Flat tip screwdriver
Clothing	Trousers	Jeans	Sweat pants
	Socks	Athletic socks	Knee-high socks
	Shirt	Dress shirt	Polo shirt
Furniture	Table	Kitchen table	Dining-room table
	Lamp	Floor lamp	Table lamp
	Chair	Armchair	Straight chair
Vehicle	Car	Sports car	Sedan car
	Airplane	Airliner plane	Fighter jet plane
	Truck	Pickup truck	Trailer truck
Fish	Snapper	Grey snapper	Red snapper
	Trout	Rainbow trout	Lake trout
	Salmon	Atlantic salmon	Chinook salmon
Bird	Owl	Barn owl	Great grey owl
	Eagle	Bald eagle	Golden eagle
	Sparrow	Song sparrow	Field sparrow

Table 5.1: The eight taxonomies of Rosch et al. (1976) used to define ImageNet concepts that our images are sampled from.

examples. Moreover, the use of a meta-learning algorithm provides a principled way to present entirely novel concepts at test time as held-out test *tasks*. As such, we can investigate the taxonomic priors encoded in a neural network embedding function, as compared to prior work that examines the representations of images from categories observed during training time (Peterson, Soulos, et al. 2018).

Concept learning as meta-learning. Meta-learning algorithms aim to learn how to learn by extracting task-general knowledge through the experience of solving a number of specific tasks (Thrun and Pratt 1998; Hochreiter, Younger, et al. 2001). In the case of concept learning, the j th task corresponds to learning a decision boundary for the j th concept using only positive examples, and meta-learning corresponds to learning how to estimate decision boundaries for arbitrary unseen concepts. We can thus formalize the concept learning problem as the task of predicting a target label y (which indicates

whether or not the input belongs to a given category) from an input observation x (i.e., an image). Note that this formulation differs from the standard discriminative classification problem, where the task corresponds to a K -way discriminative classification task in which each of the K class labels are mutually exclusive.

Formally, let $\mathcal{T}_j = (\mathbf{X}_j^{\text{trn}}, \mathbf{Y}_j^{\text{trn}}, \mathbf{X}_j^{\text{val}}, \mathbf{Y}_j^{\text{val}})$ denote a task drawn from a given task distribution $p(\mathcal{T})$, where $\mathbf{X}_j^{\text{trn}}$ and $\mathbf{Y}_j^{\text{trn}}$ are a small collection of training inputs and labels, disjoint from validation samples $\mathbf{X}_j^{\text{val}}$ and $\mathbf{Y}_j^{\text{val}}$ but belonging to the same task \mathcal{T}_j . A meta-learning algorithm (e.g., Vinyals et al. 2016; Ravi and Larochelle 2017) aims to estimate parameters θ that can be adapted to solve an unseen task $\mathcal{T}_j \sim p(\mathcal{T})$, using only the training samples $(\mathbf{X}_j^{\text{trn}}, \mathbf{Y}_j^{\text{trn}})$, to ensure the updated model achieves good performance on the validation samples $(\mathbf{X}_j^{\text{val}}, \mathbf{Y}_j^{\text{val}})$ according to some loss function \mathcal{L} .

In this work, we use the *model-agnostic meta-learning* (MAML; Finn, Abbeel, et al. (2017)) algorithm, which formulates meta-learning as estimating the parameters θ of a model so that when one or a few gradient descent steps are taken from the initialization at θ on the training data $(\mathbf{X}_j^{\text{trn}}, \mathbf{Y}_j^{\text{trn}})$, the updated model has good generalization performance on that task’s validation set, $(\mathbf{X}_j^{\text{val}}, \mathbf{Y}_j^{\text{val}})$. At test time, a new task from the test set is presented to the model for few-shot adaptation, i.e., gradient descent with $(\mathbf{X}_j^{\text{trn}}, \mathbf{Y}_j^{\text{trn}})$, and computation of test-time performance metrics, e.g., accuracy on $(\mathbf{X}_j^{\text{val}}, \mathbf{Y}_j^{\text{val}})$. The training examples in the inner gradient computation are strictly positive examples (i.e., $\mathbf{Y}_j^{\text{trn}} = 1$) of a particular concept j , whereas validation examples in the outer gradient computation include both positives and negatives (i.e., $\mathbf{Y}_j^{\text{val}} \in \{0, 1\}$); thus, at test time, the meta-learning algorithm is able to estimate a decision boundary for a novel concept from only positive examples of that concept.

5.4 Behavioral experiment

In order to compare our method directly to human behavior, we conducted a large human generalization experiment using the same naturalistic stimuli we will use to evaluate our method. We assess generalization behavior using a concept learning experiment following previous work on Bayesian concept and word learning (Xu and Tenenbaum 2007; Abbott et al. 2012; Jia et al. 2013).

Stimuli. We mapped a subset of the graph structure embedded in the ImageNet dataset used for the ImageNet Large Scale Visual Recognition Competition (ILSVRC; Russakovsky et al. (2015)) to the classic taxonomy used by cognitive scientists and developed by Rosch et al. (1976). ILSVRC is a commonly used object-classification dataset that contains more than 1 million images distributed across 1000 categories. Instead of using the leaf classes as categories, we create concepts by picking a node in the ImageNet hierarchy and sampling images from leaves dominated by the given node. Note that, in this case, concepts are not necessarily mutually exclusive in the sense that a single image may belong to one or more classes (e.g., a Dalmatian may be labeled as both a *dog* and an *animal*). If the exact

subordinate node from Rosch et al. (1976) was not available in ImageNet, we found a close semantic match via the WordNet (Fellbaum 1998) taxonomy. We provide the full taxonomy for this dataset in Table 5.1.

Task. In each of 8 trials, participants observed 5 images of a single concept, sampled from one of the three levels of taxonomic abstraction. For instance, in a subordinate training condition, the examples could be all Dalmatians; in a basic-level training condition, all dogs; in a superordinate training condition, all animals. To test generalization behavior, participants were then given a test array of 24 images and were asked to pick which images also belonged to the learned concept. The test array comprised 2 subordinate matches (e.g., other Dalmatians), 2 basic-level matches (e.g., other breeds of dog), 4 superordinate matches (e.g., other animals), and 16 out-of-domain items (e.g., inanimate objects), following Xu and Tenenbaum (2007). See Figure 5.2 for an example set of training and test stimuli. In total, we collected data for 180 unique trials and 1 180 unique images.

Participants. We recruited 900 unique participants from Amazon Mechanical Turk to each complete 8 trials as described above, one randomly sampled for each of the superordinate categories. The test sets were fixed within a superordinate category. Participants were paid \$0.40 each.

Results. Figure 5.3 (a) presents the results of the behavioral experiment for each of the three taxonomic levels. As expected on the basis of previous work, there is an exponentially decreasing generalization gradient as the level of taxonomic abstraction of the test matches (bar color) increases. However, this effect diminishes as the intra-class variation of the few-shot examples (x -axis) increases: Moving from the *subordinate* condition to the *basic*-level condition, we find an increase in the number of basic-level matches selected from the test set. The condition in which there is greatest intra-class variation—the superordinate condition—exhibits only a small generalization gradient.

5.5 Meta-learning simulations

Our modeling goal is to investigate whether we can use meta-learning to learn new concepts from only few positive examples, even though these concepts are potentially overlapping and therefore not mutually exclusive. Furthermore, we aim to investigate whether a meta-learning algorithm is able to use information about the underlying concept taxonomy that generates observations of the extension of a concept in order to generalize to novel concepts in a human-like manner.

Meta-learning formalism. Our model observes K positive examples $\mathbf{x} = \{x_1, \dots, x_K\}$ of a concept \mathcal{C} , and must learn the generalization function $p(x^* \in \mathcal{C})$ to correctly identify whether a novel example x^* is also a member of the concept. Training proceeds as follows:

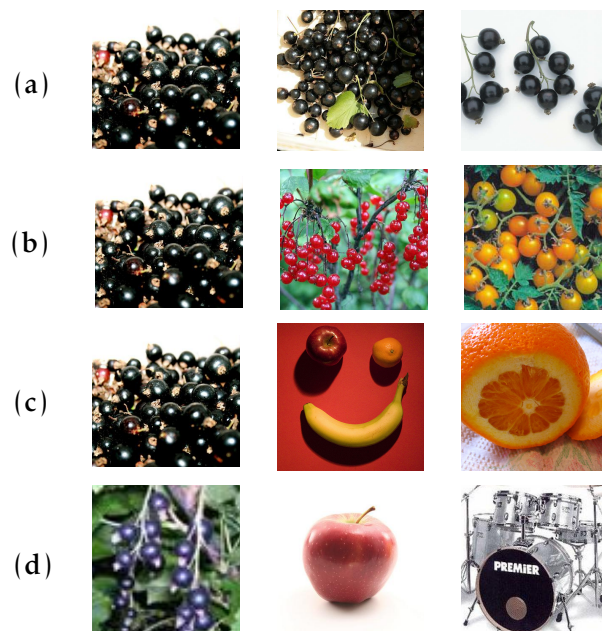


Figure 5.2: Examples of training stimuli for the (a) subordinate, (b) basic-level, and (c) superordinate level training conditions, as well as (d) a subset of the stimuli from the test array for a specific concept learning task (here, learning the concept *black currant* (a), *currant* (b) or *fruit* (c)). The test array (d) displays, from left to right, a subordinate match, a basic-level match and a superordinate match.

A concept index j is sampled from the meta-training set. Then, for K -shot learning, $2K$ positive examples of the concept and K negatives are sampled. The parameters θ are adapted using K of the positives, and then the model is optimized with a loss computed using the remaining positive and negative examples of the concept. At test time, the model with trained parameters θ is presented with K positive examples from a new concept in the test set; the model adapts θ and is evaluated on its ability to distinguish new positive examples of that concept from negatives.

Taxonomic dataset construction. For training and validation, we created a large-scale taxonomy of classes by using the graph structure embedded in the subset of the ImageNet dataset used for the ImageNet Large Scale Visual Recognition Competition (ILSVRC; Russakovsky et al. (2015)), similar to the behavioral experiment described earlier, but using the entirety of the ImageNet hierarchy. We then created few-shot concept learning tasks for training by sampling positive and negative examples for each concept, where negative examples of a concept are generated by sampling from the complement set of leaf nodes. Superordinate-level nodes are not shared between training, validation, and test to ensure that test-time generalization is measured on novel concepts. We use 494, 193, and 223 leaf nodes in the training, validation, and test sets, respectively (*c.f.*, 80, 20, and 20 in the few-shot classification dataset *miniImageNet* (Vinyals et al. 2016)). The training,

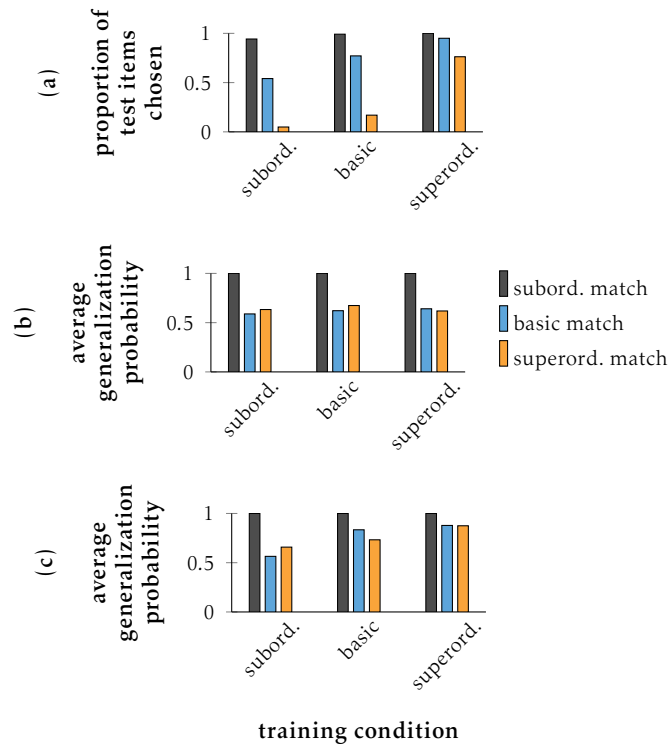


Figure 5.3: Human behavioral data (a), flat (b), and hier (c) modeling results on the concept generalization task. The horizontal axis identifies the training condition (*i.e.*, the level of taxonomic abstraction from which the few-shot examples are drawn). The vertical axis identifies, for each type of match in {subordinate, basic-level, superordinate}, the proportion of selections from the test array (a), or the average probability of generalization (b, c).

validation, and test node sets do not comprise all of the nodes in the ImageNet hierarchy, as some nodes are redundant (*i.e.*, have a single parent) or are too abstract to appropriately define a visual concept (*e.g.*, *physical entity*, *substance*, *equipment*). We make use of the training and validation dataset for training and hyperparameter selection, respectively; the test set is not used in this work but reserved for future works that may wish to perform large-scale evaluation of concept learning. Instead, the evaluations reported in this work are performed on the Rosch-inspired human benchmark described above. We also wish to emphasize that while we make use of the ImageNet hierarchy, we do so only to generate a natural distribution of concepts to learn from, and never present the explicit hierarchical relations to the model at any time.

We consider two dataset conditions in our simulations: In the `hier` dataset condition, the meta-learning algorithm observes concepts sampled from the internal and leaf nodes of the ImageNet hierarchy, and thus can learn a taxonomic prior; in the `flat` dataset condition, the algorithm observes only leaf-node concepts, and thus has no access to such information.

Hyperparameters. The base model that is optimized by MAML is a binary classifier consisting of a convolutional neural network with a sigmoid output.* In our experiments, we downsample the images to each have a width and height of 84 pixels, as is common in the use of *miniImageNet* Vinyals et al. (2016) as a few-shot learning dataset. We select hyperparameters on the same hierarchically structured validation set for both the `hier` and `flat` dataset conditions and evaluate algorithms after a fixed number of training iterations (40K with a batch size of 4). We take the value of the scalar output of the network evaluated on a test example as the *generalization probability* and average this quantity across all test examples from a specific level of taxonomic match to produce the *average generalization probability*. When reporting the average generalization probability metric, we standardize each set of probabilities for each training condition by treating the distractor (out-of-domain) generalization probability as a baseline of zero and further dividing by the largest probability in the set. In line with prior work (Peterson, Abbott, et al. 2018), this highlights the quantity of interest: the relative differences in average generalization probabilities across the subordinate, basic-level, and superordinate levels of the taxonomy.

Results. The generalization gradient observed in humans is also exhibited by the `hier` dataset condition in Figure 5.3 (c): When the few-shot examples are taken from a basic-level category (the *basic* condition; e.g., different breeds of dog) as opposed to a subordinate category (the *subord.* condition; e.g., Dalmatians), the model generalizes to more basic-level matches (e.g., different dog breeds) from the test array. In the plot, this can be seen by comparing the ratio of subordinate generalization (black column) to basic-level generalization (blue column) within each training condition (i.e., the gap between the black and blue bars is diminished in the *basic* condition vs. the *subord.* condition). Furthermore, when the few-shot examples are taken from a superordinate category (*superord.* condition), both the model in the `hier` dataset condition and humans are equally likely to pick subordinate, basic-level, or superordinate matches from the test array. In Figure 5.3 (a, c), this can be seen as the generalization to all levels of the taxonomy (black, blue, and yellow bars) being close to equal.

One notable departure of Figure 5.3 (c), from the human generalization behavior in Figure 5.3 (a), is overgeneralization to the superordinate category in the subordinate training condition, and to a lesser extent, in the basic-level training condition, suggesting that it is difficult for the algorithm to discriminate between basic-level and superordinate matches given only subordinate examples of a concept. Nevertheless, in comparison to the `flat` dataset condition in Figure 5.3 (b), which does not change generalization behavior on the basis of the training condition, the behavior of the algorithm exposed to

*The architecture of the model is similar to prior work in meta-learning (e.g., Ravi and Larochelle 2017) with 4 convolutional layers each with $32\ 3 \times 3$ filters, leaky ReLU activation functions with a slope of 0.2, and 2×2 max-pooling, all followed by a linear layer with sigmoid activation. We do not employ batch normalization because of strong batch interdependence, as all of the training examples for a concept are of the same (positive) class.

the hierarchically structured `hier` dataset suggests a learned sensitivity to the underlying taxonomic organization of new concepts.

5.6 Discussion

When humans are presented with an example from a new concept, they can quickly infer which other instances belong to that same concept even without the strong constraints provided by negative examples. In order to achieve this feat, humans bring to bear information about the taxonomic structure of natural categories. Targeting the robustness of human generalization even in highly novel domains (Schmidt 2009), we investigated the extent to which taxonomically structured biases for complex, naturalistic stimuli taken from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) could be acquired and leveraged to learn the extent of novel concepts from only a few positive examples. In contrast to previous work (Peterson, Abbott, et al. 2018), we validate the generalization behavior of our model using *unseen* supercategories drawn from the superordinate levels of Rosch’s classic taxonomy (Rosch et al. 1976).

While our method is successful in both learning a general taxonomic prior and exhibiting human-like generalization behavior, there is room for improvement as the quantitative gradients are not a perfect match to humans. However, it should be noted that our model faces the atypically challenging task of both learning a highly structured representation for complex stimuli and making use of it to generalize to entirely novel concepts. As such, this framework draws on many of the strengths of both cognitive models and deep neural networks in machine learning, and constitutes the most comprehensive account of human visual concept learning to date. Lastly, we note that we do not build in any explicit preference for simple concepts or attention to the number of examples (Tenenbaum 1999; Peterson, Soulos, et al. 2018), although this may be an interesting avenue for improvement in future work.

Chapter 6

Nonparametric priors for non-stationarity

The work described in this chapter is published as Ghassen Jerfel*, Erin Grant*, Thomas L Griffiths, and Katherine Heller (2019). “Reconciling meta-learning and continual learning with online mixtures of tasks”. In: *Advances in Neural Information Processing Systems*.

6.1 Introduction

ML algorithms aim to increase the efficiency of learning by treating task-specific learning episodes as examples from which to generalize (Schmidhuber 1987). The central assumption of a meta-learning algorithm is that some tasks are inherently related and so inductive transfer can improve sample efficiency and generalization (Caruana 1993; Caruana 1998; Baxter 2000). In learning a single set of domain-general hyperparameters that parameterize a metric space (Vinyals et al. 2016) or an optimizer (Ravi and Larochelle 2017; Finn, Abbeel, et al. 2017), recent meta-learning algorithms make the assumption that tasks are equally related, and therefore non-adaptive, mutual transfer is appropriate. This assumption has been cemented in recent few-shot learning benchmarks, which comprise a set of tasks generated in a uniform manner (*e.g.*, Vinyals et al. 2016; Finn, Abbeel, et al. 2017).

However, the real world often presents scenarios in which an agent must decide what degree of transfer is appropriate. In some cases, a subset of tasks are more strongly related to each other, and so non-uniform transfer provides a strategic advantage. On the other hand, transfer in the presence of dissimilar or outlier tasks worsens generalization performance (Rosenstein et al. 2005; Deleu and Bengio 2018). Moreover, when the underlying task distribution is non-stationary, inductive transfer to previously observed tasks should exhibit graceful degradation to address the catastrophic forgetting problem (Kirkpatrick et al. 2017). In these settings, the consolidation of all inductive biases into a single set of hyperparameters is not well-posed to deal with changing or diverse tasks. In contrast, in order to account for this degree of task heterogeneity, humans detect and adapt to novel contexts by attending to relationships between tasks (Collins and Frank 2013).

In this chapter, we learn a mixture of hierarchical models that allows a meta-learner to adaptively select over a set of learned parameter initializations for gradient-based adaptation to a new task. The method is equivalent to clustering task-specific parameters in the hierarchical model induced by recasting gradient-based ML as hierarchical Bayes (Grant et al. 2018) and generalizes the model-agnostic meta-learning (MAML) algorithm introduced in Finn, Abbeel, et al. (2017). By treating the assignment of task-specific parameters to clusters as latent variables, we can directly detect similarities between tasks on the basis of the task-specific likelihood, which may be parameterized by an expressive model such as a neural network. Our approach, therefore, alleviates the need for explicit geometric or probabilistic modeling assumptions about the weights of a complex parametric model and provides a scalable method to regulate information transfer between episodes.

We additionally consider the setting of a non-stationary or evolving task distribution, which necessitates a meta-learning method that possesses adaptive complexity. We translate stochastic point estimation in an infinite mixture (Rasmussen 2000) over model parameters into a gradient-based meta-learning algorithm that is compatible with any differentiable likelihood model and requires no distributional assumptions. We demonstrate the unexplored ability of nonparametric priors over neural network parameters to automatically detect and adapt to task distribution shift in a naturalistic image dataset;

addressing the non-trivial setting of *task-agnostic* continual learning in which the task change is unobserved (*c.f.*, *task-aware* settings such as Kirkpatrick et al. 2017).

6.2 Gradient-based meta-learning as hierarchical Bayes

Since our approach is grounded in the probabilistic formulation of meta-learning as hierarchical Bayes (Baxter 1997), our approach can be applied to any probabilistic meta-learner. In this chapter, we focus on MAML (Finn, Abbeel, et al. 2017), a gradient-based ML approach that estimates global parameters to be shared among task-specific models as an initialization for a few steps of gradient descent. MAML admits a natural interpretation as parameter estimation in a hierarchical probabilistic model, where the learned initialization acts as data-driven regularization for the estimation of task-specific parameters $\hat{\phi}_j$.

In particular, Grant et al. (2018) cast MAML as posterior inference for task-specific parameters ϕ_j given some samples of task-specific data $\mathbf{x}_{j_{1:N}}$ and a prior over ϕ_j that is induced by the early stopping of an iterative optimization procedure; truncation at K steps of gradient descent on the negative log-likelihood $-\log p(\mathbf{x}_{j_{1:N}} | \phi_j)$ starting from $\phi_{j(0)} = \theta$ can be then understood as mode estimation of the posterior $p(\phi_j | \mathbf{x}_{j_{1:N}}, \theta)$. The mode estimates $\hat{\phi}_j = \phi_{j(0)} + \alpha \sum_{k=1}^K \nabla_{\phi} \log p(\mathbf{x}_{j_{1:N}} | \phi_{j(k-1)})$ are then combined to evaluate the marginal likelihood for each task as

$$p(\mathbf{x}_{j_{N+1:N+M}} | \theta) = \int p(\mathbf{x}_{j_{N+1:N+M}} | \phi_j) p(\phi_j | \theta) d\phi_j \approx p(\mathbf{x}_{j_{N+1:N+M}} | \hat{\phi}_j), \quad (6.1)$$

where $\mathbf{x}_{j_{N+1:N+M}}$ is another set of samples from the j th task. A training dataset can then be summarized in an empirical Bayes point estimate of θ computed by gradient-based optimization of the joint marginal likelihood in Eq. (7.3) in across tasks, so that the likelihood of a datapoint sampled from a new task can be computed using only θ and without storing the task-specific parameters.

6.3 Improving meta-learning by modeling latent task structure

If the task distribution is heterogeneous, assuming a single parameter initialization θ for gradient-based meta-learning is not suitable because it is unlikely that the point estimate computed by a few steps of gradient descent will sufficiently adapt the task-specific parameters ϕ to a diversity of tasks. Moreover, explicitly estimating relatedness between tasks has the potential to aid the efficacy of a meta-learning algorithm by modulating both positive and negative transfer (Thrun and O’Sullivan 1996; Zhang and Schneider 2010; Rothman et al. 2010; Zhang and Yeung 2014), and by identifying outlier tasks that require a more significant degree of adaptation (Xue, Liao, et al. 2007; Gupta et al. 2013). Nonetheless, defining an appropriate notion of task relatedness is a difficult problem in the high-dimensional parameter or activation space of models such as neural networks.

Algorithm 1 Stochastic gradient-based EM for **finite** and **infinite** mixtures(*dataset* \mathcal{D} , *meta-learning rate* β , *adaptation rate* α , *temperature* τ , *initial cluster count* L_0 , *meta-batch size* J , *training batch size* N , *validation batch size* M , *adaptation iteration count* K , *global prior* G_0)

```

Initialize cluster count  $L \leftarrow L_0$  and meta-level parameters  $\theta^{(1)}, \dots, \theta^{(L)} \sim G_0$ 
while not converged do
  Draw tasks  $\mathcal{T}_1, \dots, \mathcal{T}_J \sim p_{\mathcal{D}}(\mathcal{T})$ 
  for  $j$  in  $1, \dots, J$  do
    Draw task-specific datapoints,  $\mathbf{x}_{j_1} \dots \mathbf{x}_{j_{N+M}} \sim p_{\mathcal{T}_j}(\mathbf{x})$ 
    Draw a cluster initialization from the global prior,  $\theta^{(L+1)} \sim G_0$ 
    for  $\ell$  in  $\{1, \dots, L, L+1\}$  do
      Initialize  $\hat{\phi}_j^{(\ell)} \leftarrow \theta^{(\ell)}$ 
      Compute task-specific mode estimate,  $\hat{\phi}_j^{(\ell)} \leftarrow \hat{\phi}_j^{(\ell)} + \alpha \sum_k \nabla_{\phi} \log p(\mathbf{x}_{j_{1:N}} | \hat{\phi}_j^{(\ell)})$ 
    end
    Compute assignment of tasks to clusters,  $\gamma_j \leftarrow \text{E-STEP}(\mathbf{x}_{j_{1:N}}, \hat{\phi}_j^{(1:L)})$ 
  end
  Update each component  $\ell$  in  $1, \dots, L$ ,  $\theta^{(\ell)} \leftarrow \theta^{(\ell)} + \text{M-STEP}(\{\mathbf{x}_{j_{N+1:N+M}}, \hat{\phi}_j^{(\ell)}, \gamma_j\}_{j=1}^J)$ 
  Summarize  $\{\theta_1, \dots\}$  to update global prior  $G_0$ 
end
return  $\{\theta^{(1)}, \dots\}$ 

```

Algorithm 6.1: Stochastic gradient-based expectation maximization (EM) for probabilistic clustering of task-specific parameters in a meta-learning setting.

$\text{E-STEP}(\{\mathbf{x}_{j_i}\}_{i=1}^N, \{\hat{\phi}_j^{(\ell)}\}_{\ell=1}^L)$ return $\tau \text{-softmax}_{\ell}(\sum_i \log p(\mathbf{x}_{j_i} \hat{\phi}_j^{(\ell)}))$	$\text{M-STEP}(\{\mathbf{x}_{j_i}\}_{i=1}^M, \hat{\phi}_j^{(\ell)}, \gamma_j)$ return $\beta \nabla_{\theta}[\sum_{j,i} \gamma_j \log p(\mathbf{x}_{j_i} \hat{\phi}_j^{(\ell)})]$
--	--

Subroutine 6.2: The E-STEP and M-STEP for a finite mixture of hierarchical Bayesian models implemented as gradient-based meta-learners.

Using the probabilistic interpretation of Section 6.2, we deal with the variability in the tasks by assuming that each set of task-specific parameters ϕ_j is drawn from a mixture of base distributions, each of which is parameterized by a hyperparameter $\theta^{(\ell)}$. Accordingly, we capture task relatedness by estimating the likelihood of assigning each task to a mixture component based simply on the task-specific negative log-likelihood after a few steps of gradient-based adaptation. The result is a scalable ML algorithm that jointly learns task-specific cluster assignments and model parameters, and is capable of modulating the transfer of information across tasks by clustering together related task-specific parameter settings.

Formally, let \mathbf{z}_j be the categorical latent variable indicating the cluster assignment of each task-specific parameter ϕ_j . Direct maximization of the mixture model likelihood is a combinatorial optimization problem that can grow intractable. This intractability is equally problematic for the posterior distribution over the cluster assignment variables \mathbf{z}_j and the task-specific parameters ϕ_j , which are both treated as latent variables in the probabilistic formulation of meta-learning. A scalable approximation involves representing the conditional distribution for each latent variable with a MAP estimate. In our meta-learning setting of a mixture of hierarchical Bayes models, this suggests an augmented expectation maximization (EM) procedure (Dempster et al. 1977) alternating between an E-STEP that computes an expectation of the task-to-cluster assignments \mathbf{z}_j , which itself involves the computation of a conditional mode estimate for the task-specific parameters ϕ_j , and an M-STEP that updates the hyperparameters $\theta^{(1:L)}$ (see Subroutine 6.2).

To ensure scalability, we use the minibatch variant of stochastic optimization (Robbins and Monro 1951) in both the E-STEP and the M-STEP; such approaches to EM are motivated by a view of the algorithm as optimizing a single free energy at both the E-STEP and the M-STEP (Neal and Hinton 1998). In particular, for each task j and cluster ℓ , we follow the gradients to minimize the negative log-likelihood on the training data points $\mathbf{x}_{j_{1:N}}$, using the cluster parameters $\theta^{(\ell)}$ as initialization. This allows us to obtain a modal point estimate of the task-specific parameters $\hat{\phi}_j^{(\ell)}$. The E-STEP in Subroutine 6.2 leverages the connection between gradient-based ML and hierarchical Bayes (Grant et al. 2018) and the differentiability of our clustering procedure to employ the task-specific parameters to compute the posterior probability of cluster assignment. Accordingly, based on the likelihood of the same training data points under the model parameterized by $\hat{\phi}_j^{(\ell)}$, we compute the cluster assignment probabilities as

$$\gamma_j^{(\ell)} := p(\mathbf{z}_j = \ell \mid \mathbf{x}_{j_{1:N}}, \theta^{(1:L)}) \propto \int p(\mathbf{x}_{j_{1:N}} \mid \phi_j) p(\phi_j \mid \theta^{(\ell)}) d\phi_j \approx p(\mathbf{x}_{j_{1:N}} \mid \hat{\phi}_j^{(\ell)}). \quad (6.2)$$

The cluster means $\theta^{(\ell)}$ are then updated by gradient descent on the validation loss in the M-STEP in Subroutine 6.2; this M-STEP is analogous to the MAML algorithm in Finn, Abbeel, et al. (2017) with the addition of mixing weights $\gamma_j^{(\ell)}$.

Note that, unlike other recent approaches to probabilistic clustering (e.g., Bauer et al. 2017) we adhere to the episodic meta-learning setup for both training and testing since only the task support set $\mathbf{x}_{j_{1:N}}$ is used to compute both the point estimate $\hat{\phi}_j^{(\ell)}$ and the cluster responsibilities $\gamma_j^{(\ell)}$. See Algorithm 6.1 for the full algorithm, whose high-level structure is shared with the nonparametric variant of our method detailed in Section 6.5.

Model	Num. param.	1-shot (%)	5-shot (%)
matching network (Vinyals et al. 2016) ^a		43.56 ± 0.84	55.31 ± 0.73
meta-learner LSTM (Ravi and Larochelle 2017)		43.44 ± 0.77	60.60 ± 0.71
prototypical networks (Snell et al. 2017) ^b		46.61 ± 0.78	65.77 ± 0.70
MAML (Finn, Abbeel, et al. 2017)		48.70 ± 1.84	63.11 ± 0.92
MT-net (Lee and Choi 2018)	38,907	51.70 ± 1.84	
PLATIPUS (Finn, Xu, et al. 2018)	65,546	50.13 ± 1.86	
VERSA (Gordon et al. 2019) ^c	807,938	48.53 ± 1.84	
Our method: 2 components	65,546	49.60 ± 1.50	64.60 ± 0.92
3 components	98,319	51.20 ± 1.52	65.00 ± 0.96
4 components	131,092	50.49 ± 1.46	64.78 ± 1.43
5 components	163,865	51.46 ± 1.68	

Table 6.1: Meta-test set accuracy on the *miniImageNet* 5-way, 1- and 5-shot classification benchmarks from Vinyals et al. (2016) among methods using a comparable architecture (the 4-layer convolutional network from Vinyals et al. (2016)). For methods on which we report results in later experiments, we additionally report the total number of parameters optimized by the meta-learning algorithm.

6.4 Experiment: *miniImageNet* few-shot classification

Clustering task-specific parameters provides a way for a meta-learner to deal with task heterogeneity as each cluster can be associated with a subset of the tasks that would benefit most from mutual transfer. While we do not expect existing tasks to present a significant degree of heterogeneity given the uniform sampling assumptions behind their design, we nevertheless conduct an experiment to validate that our method gives an improvement on a standard benchmark for few-shot learning.

We apply Algorithm 6.1 with Subroutine 6.2 and $L \in \{2, 3, 4, 5\}$ components to the 1-shot and 5-shot, 5-way, *miniImageNet* few-shot classification benchmarks (Vinyals et al. 2016); Appendix B.1 contains additional experimental details. We demonstrate in Table 6.1 that a mixture of meta-learners improves the performance of gradient-based meta-learning on this task for any number of components. However, the performance of the parametric mixture does not improve monotonically with the number of components L . This leads us to the development of nonparametric clustering for continual meta-learning, where enforcing specialization to subgroups of tasks and increasing model complexity is, in fact, necessary to preserve performance on prior tasks due to significant heterogeneity.

^a Results reported by Ravi and Larochelle (2017). ^b We report test accuracy for models matching train and test “shot” and “way”. ^c We report test accuracy for a comparable base (task-specific network) architecture.

```

E-STEP(  $\mathbf{x}_{j:1:N}, \hat{\phi}_j^{(1:L)}$ , concentration  $\zeta$ , threshold  $\epsilon$ )
| DPMM log-likelihood for all  $\ell$  in  $1, \dots, L$ ,  $\rho_j^{(\ell)} \leftarrow \sum_i \log p(\mathbf{x}_{j_i} | \hat{\phi}_j^{(\ell)}) + \log n^{(\ell)}$ 
| DPMM log-likelihood for new component,  $\rho_j^{(L+1)} \leftarrow \sum_i \log p(\mathbf{x}_{j_i} | \hat{\phi}_j^{(L+1)}) + \log \zeta$ 
| DPMM assignments,  $\gamma_j \leftarrow \tau\text{-softmax}(\rho_j^{(1)}, \dots, \rho_j^{(L+1)})$ 
| if  $\gamma_j^{(L+1)} > \epsilon$  then
| | Expand the model by incrementing  $L \leftarrow L + 1$ 
| else
| | Renormalize  $\gamma_j \leftarrow \tau\text{-softmax}(\rho_j^{(1)}, \dots, \rho_j^{(L)})$ 
| return  $\gamma_j$ 

```

```

M-STEP(  $\{\mathbf{x}_{j_i}\}_{i=1}^M, \hat{\phi}_j^{(\ell)}, \gamma_j$ , concentration  $\zeta$ )
| return  $\beta \nabla_{\theta} [\sum_{j,i} \gamma_j \log p(\mathbf{x}_{j_i} | \hat{\phi}_j^{(\ell)}) + \log n^{(\ell)}]$ 

```

Subroutine 6.3: The E-STEP and M-STEP for an infinite mixture of hierarchical Bayesian models.

6.5 Scalable online mixtures for task-agnostic continual learning

The mixture of meta-learners developed in Section 6.3 addresses a drawback of meta-learning approaches such as MAML that consolidate task-general information into a single set of hyperparameters. However, the method adds another dimension to model selection in the form of identifying the correct number of mixture components. While this may be resolved by cross-validation if the dataset is static and therefore the number of components can remain fixed, adhering to a fixed number of components throughout training is not appropriate in the non-stationary regime, where the underlying task distribution changes as different types of tasks are presented sequentially in a continual learning setting. In this regime, it is important to incrementally introduce more components that can each specialize to the distribution of tasks observed at the time of spawning.

To address this, we derive a scalable stochastic estimation procedure to compute the expectation of task-to-cluster assignments (E-STEP) for a growing number of task clusters in a *nonparametric* mixture model (Rasmussen 2000) called the Dirichlet process mixture model (DPMM). The formulation of the Dirichlet process mixture model (DPMM) that is most appropriate for incremental learning is the sequential draws formulation that corresponds to an instantiation of the Chinese restaurant process (CRP) (Rasmussen 2000). A CRP prior over \mathbf{z}_j allows some probability to be assigned to a new mixture component while the task identities are inferred in a sequential manner, and has therefore been key to recent online and stochastic learning of the DPMM (Lin 2013). A draw from a CRP proceeds as follows: For a sequence of tasks, the first task is assigned to the first cluster

and the j th subsequent task is then assigned to the ℓ th cluster with probability

$$p(\mathbf{z}_j = \ell \mid \mathbf{z}_{1:j-1}, \zeta) = \begin{cases} n^{(\ell)}/n + \zeta & \text{for } \ell \leq L \\ \zeta/n + \zeta & \text{for } \ell = L + 1, \end{cases} \quad (6.3)$$

where L is the number of non-empty clusters, $n^{(\ell)}$ is the number of tasks already occupying a cluster ℓ , and ζ is a fixed positive concentration parameter. The prior probability associated with a new mixture component is therefore $p(\mathbf{z}_j = L + 1 \mid \mathbf{z}_{1:j-1}, \zeta)$.

In a similar spirit to Section 6.3, we develop a stochastic EM procedure for the estimation of the latent task-specific parameters $\phi_{1:j}$ and the meta-level parameters $\theta^{(1:L)}$ in the DPMM, which allows the number of observed task clusters to grow in an online manner with the diversity of the task distribution. While computation of the mode estimate of the task-specific parameters ϕ_j is mostly unchanged from the finite variant, the estimation of the cluster assignment variables \mathbf{z} in the E-STEP requires revisiting the Gibbs conditional distributions due to the potential addition of a new cluster at each step. For a DPMM, the conditional distributions for \mathbf{z}_j are

$$p(\mathbf{z}_j = \ell \mid \mathbf{x}_{j_{1:M}}, \mathbf{z}_{1:j-1}) \propto \begin{cases} n^{(\ell)} \int p(\mathbf{x}_{j_{1:M}} \mid \phi_j^{(\ell)}) p(\phi_j^{(\ell)} \mid \theta) d\phi_j dG_\ell(\theta) & \text{for } \ell \leq L \\ \zeta \int p(\mathbf{x}_{j_{1:M}} \mid \phi_j^{(0)}) p(\phi_j^{(0)} \mid \theta) d\phi_j dG_0(\theta) & \text{for } \ell = L + 1 \end{cases} \quad (6.4)$$

with G_0 as the base measure or global prior over the components of the CRP, G_ℓ is the prior over each cluster's parameters, initialized with a draw from a Gaussian centered at G_0 with a fixed variance and updated over time using summary statistics from the set of active components $\{\theta^{(0)}, \dots, \theta^{(L)}\}$.

Taking the logarithm of the posterior over task-to-cluster assignments \mathbf{z}_j in (6.4) and using a mode estimate $\hat{\phi}_j^{(\ell)}$ for task-specific parameters ϕ_j as drawn from the ℓ th cluster gives the E-STEP in Subroutine 6.3. We may also omit the prior term $\log p(\hat{\phi}_j^{(\ell)} \mid \theta^{(\ell)})$ as it arises as an implicit prior resulting from truncated gradient descent, as explained in Section 6.3 of Grant et al. (2018).

6.6 Experiments: *Task-agnostic* continual few-shot regression & classification

By treating the assignment of tasks to clusters as latent variables, the algorithm of Section 6.5 can adapt to a changing distribution of tasks, without any external information to signal distribution shift (*i.e.*, in a *task-agnostic* manner). Here, we present our main experimental results on both a novel synthetic regression benchmark as well as a novel evolving variant of *miniImageNet*, and confirm the algorithm's ability to adapt to distribution shift by spawning a newly specialized cluster.

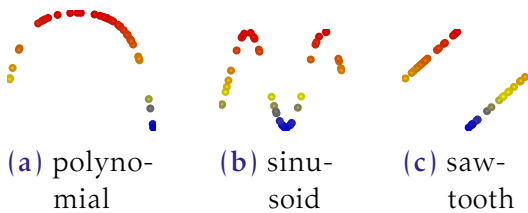


Figure 6.4: The diverse set of periodic functions used for few-shot regression in Section 6.6.



Figure 6.5: Artistic filters (b-d) applied to *miniImageNet* (a) to ensure non-homogeneity of tasks in Section 6.6.

High-capacity baselines. As an ablation, we compare to the **non-uniform** parametric **mixture** proposed in Section 6.3 with the number of components fixed at the total number of task distributions in the dataset (3). We also consider a **uniform** parametric **mixture** in which each component receives equal assignments; this can also be seen as the non-uniform mixture in the infinite temperature (τ) limit. Note that our meta-learner has a lower capacity than these two baselines for most of the training procedure, as it may decide to expand its capacity past one component only when the task distribution changes. Finally, for the large-scale experiment in Section 6.6, we compare with three recent meta-learning algorithms that report improved performance on the standard *miniImageNet* benchmark of Section 6.3, but are not explicitly posed to address the continual learning setting of evolving tasks: **MT-net** (Lee and Choi 2018), **PLATIPUS** (Finn, Xu, et al. 2018), and **VERSA** (Gordon et al. 2019).

Continual few-shot regression

We first consider an explanatory experiment in which three regression tasks are presented sequentially with no overlap. For input x sampled uniformly from $[-5, 5]$, each regression task is generated, in a similar spirit to the sinusoidal regression setup in Finn, Abbeel, et al. (2017), from one of a set of simple but distinct one-dimensional functions (polynomial Fig. 6.4a, sinusoid wave Fig. 6.4b, and sawtooth wave Fig. 6.4c). For the experiment in Fig. 6.6 and Fig. 6.7, we presented the polynomial tasks for 4000 iterations, followed by sinusoid tasks for 3000 iterations, and finally sawtooth tasks. Additional details on the experimental setup can be found in Appendix B.1.

Results: Distribution shift detection. The cluster responsibilities in Fig. 6.7 on the meta-test dataset of tasks, from each of the three regression types in Fig. 6.4, indicate that the nonparametric algorithm recognizes a change in the task distribution and spawns a new cluster at iteration 4000 and promptly after iteration 7000. Each newly created cluster is specialized to the task distribution observed at the time of spawning and remains as such throughout training, since the majority of assignments for each type of regression remains under a given cluster from the time of its introduction.

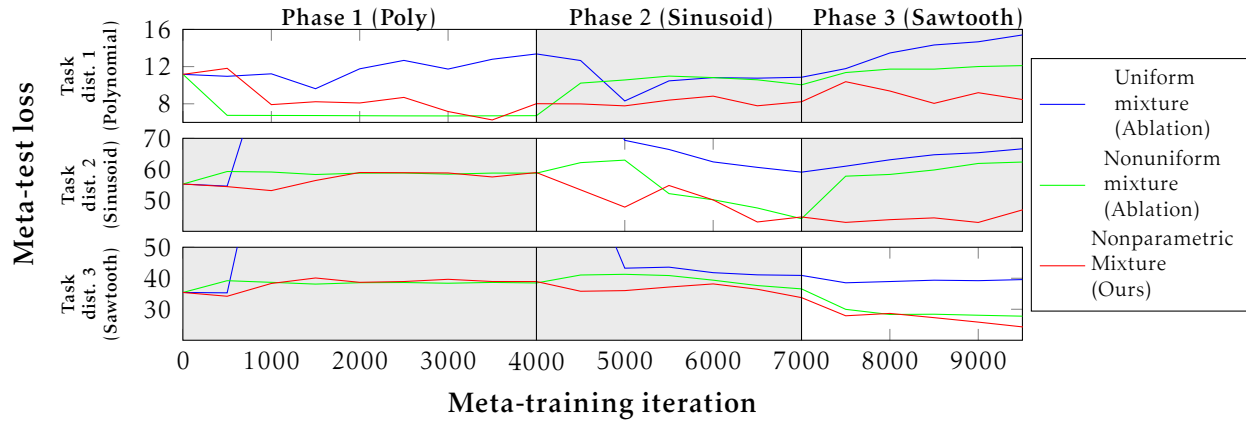


Figure 6.6: Results on the evolving dataset of few-shot regression tasks (lower is better). Each panel (row) presents, for a specific task type (polynomial, sinusoid or sawtooth), the average meta-test set accuracy of each method over cumulative number of few-shot episodes. We additionally report the degree of loss in backward transfer (i.e., catastrophic forgetting) to the tasks in each meta-test set in the legend; all methods but the nonparametric method experience a large degree of catastrophic forgetting during an inactive phase.

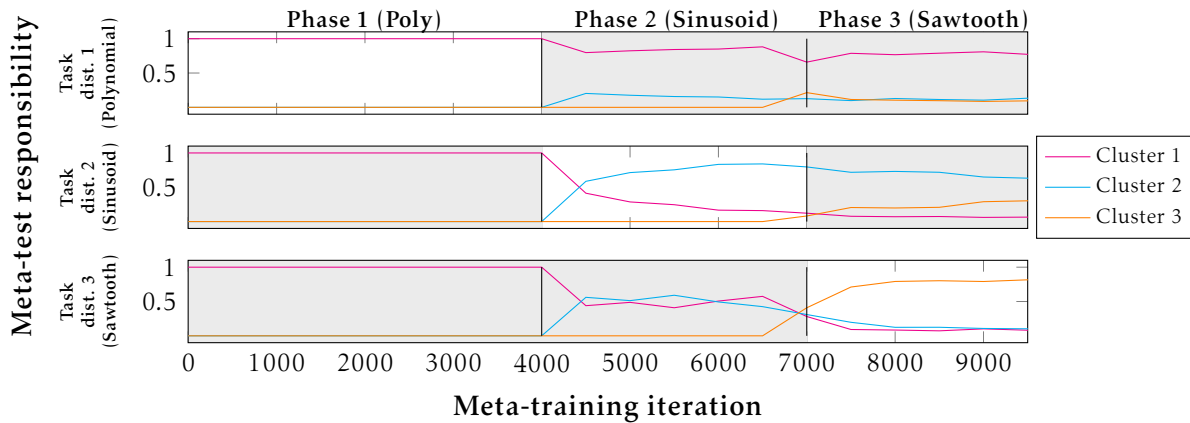


Figure 6.7: Task-specific per-cluster meta-test responsibilities $\gamma^{(\ell)}$ for both active and unspawned clusters. Higher responsibility implies greater specialization of a particular cluster (color) to a particular task distribution (row).

Results: Improved generalization and slower degradation of performance. We investigate the progression of the meta-test mean-squared error (MSE) for the three regression task distributions in Fig. 6.6. We first note the clear advantage of non-uniform assignment both in improved generalization, when testing on the active task distribution, and in slower degradation, when testing on previous distributions. This is due to the ability of these methods to modulate the transfer of information in order to limit negative transfer. In contrast, the uniform method cannot selectively adapt specific clusters to be responsible for any given task, and thus inevitably suffers from catastrophic forgetting.

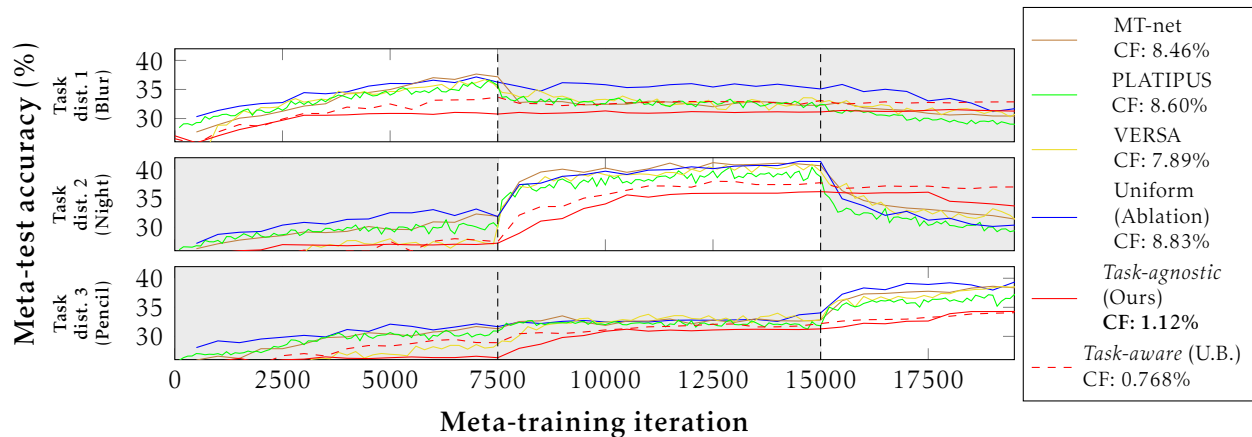


Figure 6.8: Results on the evolving dataset of filtered *miniImageNet* few-shot classification tasks (higher is better). Each panel (row) presents, for a specific task type (filter), the average meta-test set accuracy over cumulative number of few-shot episodes. We additionally report the degree of loss in backward transfer (catastrophic forgetting, CF) in the legend. This is calculated for each method as the average drop in accuracy on the first two tasks at the end of training (lower is better; U.B.: upper bound).

The adaptive capacity of our nonparametric method allows it to spawn clusters that specialize to newly observed tasks. Accordingly, even if the overall capacity is lower than that of the comparable non-uniform parametric method, our method achieves similar or better generalization, at any given training iteration. More importantly, specialization allows our method to better modulate information transfer as the clusters are better differentiated. Consequently, each cluster does not account for many assignments from more than a single task distribution throughout training. Therefore, we observed a significantly slower rate of degradation of the MSE on previous task distributions as new tasks are introduced. This is especially evident from the performance on the first task in Fig. 6.6.

Continual few-shot classification

Next, we consider an evolving variant of the *miniImageNet* few-shot classification task. In this variant, one of a set of artistic filters are applied to the images during the meta-training procedure to simulate a changing distribution of few-shot classification tasks. For the experiment in Figure 8 and Figure 9 we first train using images with a “blur” filter (Fig. 6.5b) for 7500 iterations, then with a “night” filter (Fig. 6.5c) for another 7500 iterations, and finally with a “pencil” filter (Fig. 6.5d). Additional details on the experimental setup can be found in Appendix B.1.

Results: Meta-test accuracy. In Figure 9, we report the evolution of the meta-test accuracy for two variants of our nonparametric meta-learner in comparison to the parametric

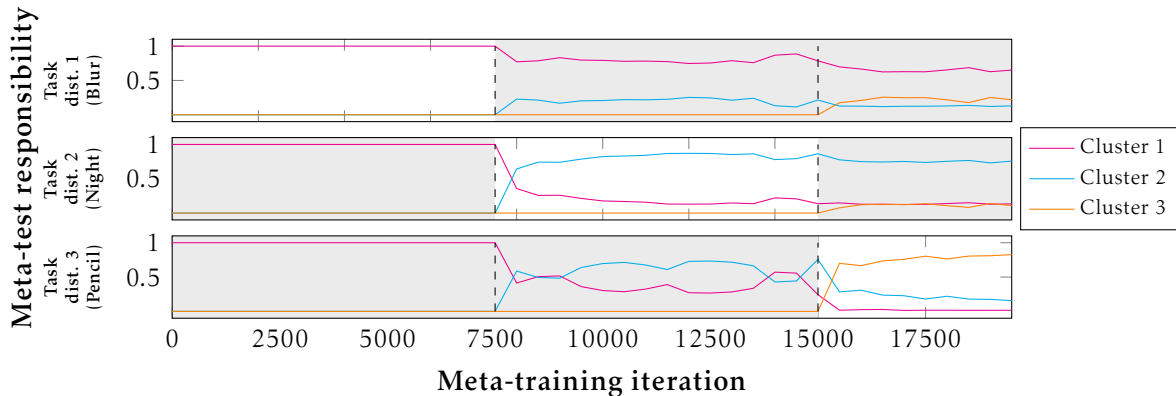


Figure 6.9: Task-specific per-cluster meta-test responsibilities $\gamma^{(\ell)}$ for both active and unspawned clusters. Higher responsibility implies greater specialization of a particular cluster (color) to a particular task distribution (row).

baselines introduced at the start of the section, *high-capacity baselines*. The *task-agnostic* variant is the core algorithm described in previous sections, as used for the regression tasks. The *task-aware* variant augments the core algorithm with a cool-down period that prevents over-spawning for the duration of a training phase. This requires some knowledge of the duration which is external to the meta-learner, thus the *task-aware* nomenclature (note that this does not correspond to a true oracle, as we do not enforce spawning of a cluster; see Appendix B.1 for further details).

It is clear from Figure 8 that neither of our algorithms suffer from catastrophic forgetting to the same degree as the parametric baselines. In fact, at the end of training, both of our methods outperform all the parametric baselines on the first and second task.

Results: Specialization. Given the higher capacity of the parametric baselines and the inherent degree of similarity between the filtered *miniImageNet* task distributions (unlike the regression tasks in the previous section), the parametric baselines perform better on each task distribution while during its active phase. However, they quickly suffer from degradation once the task distribution shifts. Our approach does not suffer from this phenomenon and can handle non-stationarity owing to the credit assignment of a single task distribution to a specialized cluster. This specialization is illustrated in Figure 9, where we track the evolution of the average cluster responsibilities on the meta-test dataset from each of the three *miniImageNet* few-shot classification tasks. Each cluster is specialized so as to acquire the majority of a single task distribution’s test set assignments, despite the degree of similarity between tasks originating from the same source (*miniImageNet*). We observed this difficulty with the non-monotone improvement of parametric clustering as a function of components in Section 4.

6.7 Related work

Meta-learning. In this work, we show how changes to the hierarchical Bayesian model assumed in meta-learning (Grant et al. 2018, Fig. 1(a)) can be realized as changes to a meta-learning algorithm. In contrast, follow-up approaches to improving the performance of meta-learning algorithms (*e.g.*, Lee and Choi 2018; Finn, Xu, et al. 2018; Gordon et al. 2019) do not change the underlying probabilistic model; what differs is the inference procedure to infer values of the global (shared across tasks) and local (task-specific) parameters; for example, Gordon et al. (2019) consider feedforward conditioning while Finn, Xu, et al. (2018) employ variational inference. Due to consolidation into one set of global parameters shared uniformly across tasks, none of these methods inherently accommodate heterogeneity or non-stationarity.

Continual learning. Techniques developed to address the catastrophic forgetting problem in continual learning, such as elastic weight consolidation (EWC) (Kirkpatrick et al. 2017), synaptic intelligence (SI) (Zenke et al. 2017), variational continual learning (VCL) (Nguyen et al. 2017), and online Laplace approximation (Ritter, Botev, et al. 2018) require access to an explicit delineation between tasks that acts as a catalyst to grow model size, which we refer to as *task-aware*. In contrast, our nonparametric algorithm tackles the *task-agnostic* setting in which the meta-learner recognizes a latent shift in the task distribution and adapts accordingly.

6.8 Conclusion

Meta-learning is a source of learned inductive bias. Occasionally, this inductive bias is harmful because the experience gained from solving a task does not transfer. In this chapter, we present an approach that allows a probabilistic meta-learner to explicitly modulate the amount of transfer between tasks, as well as to adapt its parameter dimensionality when the underlying task distribution evolves. We formulate this as probabilistic inference in a mixture model that defines a clustering of task-specific parameters. To ensure scalability, we make use of the recent connection between gradient-based meta-learning and hierarchical Bayes (Grant et al. 2018) to perform approximate *maximum a posteriori* (MAP) inference in both a finite and an infinite mixture model. This chapter is a first step towards more realistic settings of diverse task distributions, and crucially, *task-agnostic* continual learning. The approach stands to benefit from orthogonal improvements in posterior inference beyond MAP estimation (*e.g.*, variational inference (Jordan et al. 1999), Laplace approximation (MacKay 1992a), or stochastic gradient Markov chain Monte Carlo (Metropolis and Ulam 1949)) as well as scaling up the neural network architecture.

Part III

Computational modeling of neural networks

Chapter 7

Gaussian process surrogate models

The work described in this chapter is in preparation as Michael Y. Li, Erin Grant, and Thomas L Griffiths (2022). “Gaussian process surrogate models for neural networks”. In: eprint: [arXiv](#). An earlier version of this work appeared as Michael Y. Li, Erin Grant, and Thomas L Griffiths (2021). “Meta-learning inductive biases of learning systems with Gaussian processes”. In: *Proceedings of the NeurIPS Workshop on Meta-Learning*.

7.1 Introduction

Deep learning systems are ubiquitous in machine learning but sometimes exhibit unpredictable and often undesirable behavior when deployed in real-world applications (Geirhos, Jacobsen, et al. 2020; D’Amour et al. 2020). This gap between idealized and real-world performance stems from a lack of principles guiding the design of deep learning systems. Instead, deep learning practitioners often rely upon a set of heuristic design decisions that are inadequately tied to a system’s behavior (Dehghani et al. 2021), driving calls for explainability, transparency, and interpretability of deep learning systems (Lipton 2016; Doshi-Velez and Kim 2017; Samek et al. 2017) especially as these systems are more widely applied in everyday life (Bommasani et al. 2021).

Machine learning is not unique in seeking to understand a complex system whose inputs and outputs are observable but whose internal processes are opaque—this challenge occurs across the empirical sciences and engineering. An explanatory tool that is foundational across these disciplines is that of *modeling*, that is, representing a complex and opaque system with a simpler one that is more amenable to interpretation.* Modeling makes precise assumptions about how a system may operate while abstracting away details that are irrelevant for a given level of understanding or a given downstream use case. These properties are valuable for a framework for understanding deep learning as they are in other scientific and engineering disciplines.

As the popularity of deep learning has grown, a number of proposals have been made for modeling these systems. Numerous mathematical models of deep learning have been developed (Roberts et al. 2022), and some surprising phenomena, such as adversarial examples (Szegedy et al. 2014), have been captured with a mathematical analysis (Ilyas, Santurkar, et al. 2019). However, existing mathematical models, which are limited to well-understood mathematical tools, are unable to capture the properties of machine learning systems as applied in practice (Nakkiran 2021). Beyond mathematical models, localized models have aimed to explain the predictions of machine learning systems on a per-example basis (Ribeiro et al. 2016; Koh and Liang 2017; Zhou et al. 2022; Ilyas, Park, et al. 2022), but these approaches are, by construction, only partial explanations of the behavior of the end-to-end system.

What might an alternative modeling approach—one that captures salient aspects of applied systems in a global fashion—look like? We appeal to two domains for inspiration. In engineering design, *surrogate models* (Wang and Shan 2006) emulate the input-output behavior of a complex physical system, allowing practitioners to simulate effects that are consequential for design or analysis without relying on costly or otherwise prohibitive queries from the system itself. In cognitive science, *cognitive models* (Sun 2008; McClelland 2009) describe how unobservable mental processes such as memory or attention produce the range of people’s observed behaviors. Both domains abstract away internal details,

*Though some architectural components of a deep learning system are commonly referred to as a *model*—as in “neural network model”—we use *modeling* to refer to the methodology of idealizing a complex system as a simpler one.

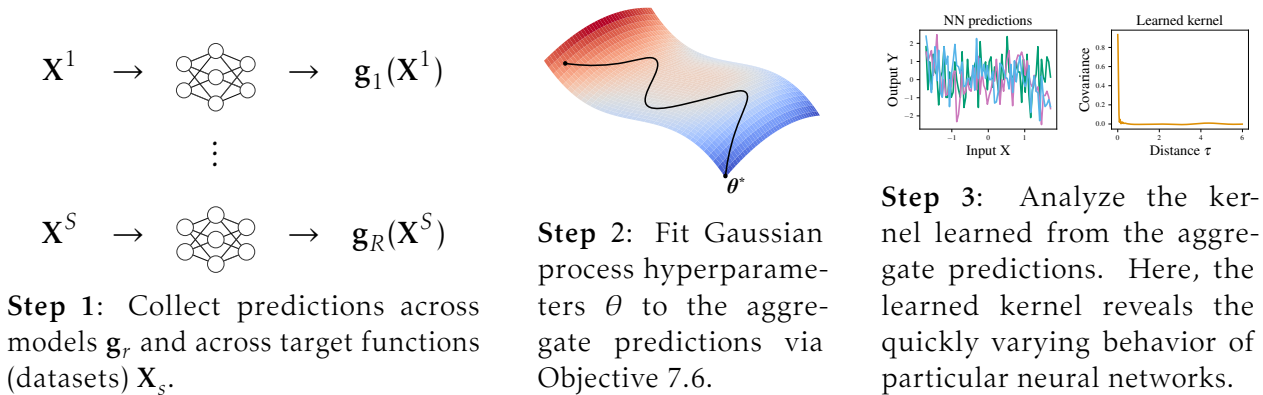


Figure 7.1: Outline of the surrogate modeling approach. We learn a Gaussian process surrogate model for a neural network family applied to a task family by learning kernel hyperparameters from aggregated neural network predictions across datasets. We interpret the learned kernel to derive insights into the properties of the neural network family; for example, biases towards particular frequencies (see Section 7.4), or expected generalization behavior on a new dataset (see Section 7.4).

such as real-world constraints on a physical system or neural circuitry in the brain, instead treating the target process or system as a *black box*. At the same time, both surrogate and cognitive models are constructed to replicate the end-to-end behavior of the target system and thus are complete where localized explanations are not.

We explore an analogous approach to investigate deep learning systems by constructing *surrogate models for neural networks*. We first must choose an appropriate family of surrogate models. GPs are a natural choice, with appealing theoretical properties specific to the study of NNs; namely, certain limiting cases of NN architectures are realizable as GPs (Neal 1996; Li and Liang 2018; Jacot et al. 2018; Allen-Zhu et al. 2019; Du et al. 2019). However, in contrast to these analytic approaches, we aim to explore the scientific and practical utility of idealizing NNs with GPs using a *data-driven* approach to estimating the kernel functions. Separately, the learned kernel of a GP is often interpretable (Wilson and Adams 2013); we use this fact to study the prior over functions represented by a GP that accounts for observed neural network behavior in less-restricted settings. With this approach, we capture a number of known phenomena, including a bias towards low frequencies and pathological behavior at initialization, in a cohesive framework. Finally, we demonstrate the practical benefits of this framework by predicting the generalization behavior of models in an NN family.

7.2 Background

In **surrogate modeling**, we approximate a complex black-box function with a simpler surrogate model that is more amenable to interpretation. Surrogate models have many applications: In optimization, they are often used to approximate queries from expensive-

to-evaluate functions (Snoek et al. 2012; Shahriari et al. 2016; Xue, Beatson, et al. 2020); in other applications, surrogate models have been used to gain insight into large physical systems, such as the global fluxes of energy and heat over the earth’s surface (Camps-Valls et al. 2015).

Cognitive models have been used by cognitive scientists since the 1950s to gain insight into another black box—the human mind (Newell et al. 1958). Bayesian models of cognition, in particular, offer a way to describe the inductive biases of learning systems in the form of a prior distribution (Griffiths et al. 2010). As deep NNs have become more prevalent in machine learning, researchers have started to use methodologies from cognitive science to interrogate otherwise opaque models (Ritter, Barrett, et al. 2017; Geirhos, Rubisch, et al. 2019; Hawkins et al. 2020). The success of these efforts suggests that other methods from cognitive science—namely, cognitive modeling—may be applicable to machine learning systems.

Gaussian processes (GPs; Rasmussen and Williams 2006) are probabilistic models that specify a distribution over functions. A GP models any *finite* set of N observations as a multivariate Gaussian distribution on \mathbb{R}^D , where the n th point is interpreted as the function value, $f(\mathbf{x}_n)$, at the input point \mathbf{x}_n . GPs are fully characterized by a mean function $m(\mathbf{x})$, usually taken to be degenerate as $m(\mathbf{x}) = \mathbf{0}, \forall \mathbf{x}$, and a positive-definite kernel function $k(\mathbf{x}, \mathbf{x}')$ that gives the covariance between $f(\mathbf{x})$ and $f(\mathbf{x}')$ as a function of \mathbf{x} and \mathbf{x}' .

Formally, let \mathbf{X} be a matrix of inputs and \mathbf{y} be a vector of output responses. Due to the marginalization properties of the Gaussian distribution, the posterior predictive distribution of a GP for a new input \mathbf{x}_* , conditioned on dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ and assuming centered Gaussian observation noise with variance σ^2 , is Gaussian with closed-form expressions for the mean and variance:

$$\mathbb{E}[f(\mathbf{x}_*) | \mathcal{D}] = m(\mathbf{x}_*) + \mathbf{k}_*^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} (\mathbf{y} - m(\mathbf{x}_*)) \quad (7.1)$$

$$\mathbb{V}[f(\mathbf{x}_*) | \mathcal{D}] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_* \quad (7.2)$$

where \mathbf{K} is the $N \times N$ Gram matrix of pairwise covariances, $k(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{k}_* = [k(\mathbf{x}_1, \mathbf{x}_*), \dots, k(\mathbf{x}_N, \mathbf{x}_*)]^T$.

The kernel function k specifies the prior on what kind of functions might be represented in observed data; for example, it can express expectations about smoothness or periodicity. Parametric kernels have hyperparameters θ that affect this prior and thus the posterior predictive. These kernel hyperparameters can be adapted to the properties of a dataset, thus defining a prior over functions that is appropriate for that context. GP kernel hyperparameters are typically learned via gradient-based optimization to maximize the GP marginal likelihood, $p(\mathbf{y} | \mathbf{X})$. Again due to properties of the GP, this marginal likelihood has the closed-form expression:

$$\log p(\mathbf{y} | \mathbf{X}) = -\frac{1}{2} \mathbf{y}^T (\mathbf{K}_\theta + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{K}_\theta + \sigma_n^2 \mathbf{I}| - \frac{n}{2} \log 2\pi, \quad (7.3)$$

We write the Gram matrix as \mathbf{K}_θ to indicate that it depends on kernel hyperparameters via a particular parameterization. In this work, we make use of two kernel parameterizations:

the **Matérn kernel** (MK; Matérn 1960) and the **spectral mixture kernel** (SMK; Wilson and Adams 2013). Specifically, following Snoek et al. (2012), we use the automatic relevance determination (ARD) 5/2 MK, given by:

$$k(\mathbf{x}, \mathbf{x}') = \theta_0 \left(1 + \sqrt{5r^2(\mathbf{x}, \mathbf{x}') + \frac{5}{3}r^2(\mathbf{x}, \mathbf{x}')} \right) \exp \left\{ -5\sqrt{5r^2(\mathbf{x}, \mathbf{x}')} \right\} \quad r^2(\mathbf{x}, \mathbf{x}') = \sum_{d=1}^D (x_d - x'_d)^2 / \theta_d^2, \quad (7.4)$$

where each θ_d is the lengthscale parameter for dimension d , which captures how smoothly the function varies along that dimension. The SMK is derived by modeling the spectral density of a kernel as a scale-location mixture of Gaussians and computing the Fourier transform of the mixture (Wilson and Adams 2013), giving:

$$k(\tau) = \sum_{q=1}^Q w_q \cos(2\pi^2 \tau^T \mu_q) \prod_{p=1}^P \exp \left\{ -2\pi^2 \tau_p^2 v_q^{(p)} \right\}. \quad (7.5)$$

Here, $k(\tau)$ gives the covariance between function values $f(\mathbf{x})$ and $f(\mathbf{x}')$ whose corresponding input values \mathbf{x} and \mathbf{x}' are a distance τ apart. For a Q -component spectral mixture, $w = \{w_i\}_{i=1}^Q$ are scalar mixture weights, and $\mu_i \in \mathbb{R}^P$ and $v_i \in \mathbb{R}^P$ are component-wise Gaussian means and variances, respectively. Appendix B.3 details how the hyperparameters of the MK and the SMK control the respective priors on functions.

7.3 Learning a Gaussian process surrogate model from neural network predictions

In this section, we detail the goals and approach of the surrogate modeling framework. In brief, our approach involves collecting neural network predictions across a set of neural network models and across a set of datasets, and estimating GP kernel hyperparameters from these predictions by maximizing the marginal likelihood across model-and-dataset pairs; see Fig. 7.1 for a schematic.

Formal framework

Our goal is to capture shared properties among a family of neural networks models \mathcal{F} as applied to a family of datasets \mathcal{D} . Here, a model family \mathcal{F} is a set of neural networks $\{\mathbf{g}_0, \dots, \mathbf{g}_R\}$ that share in design choices (*e.g.*, architecture, training procedure, random initialization scheme) but differ in quantities that are randomized prior to or during training (*e.g.*, parameter initializations).[†] Similarly, a dataset family \mathcal{D} is a set of datasets

[†]We consider both untrained and trained neural networks, where an untrained network is a special case of a trained network with the number of training iterations at 0; we thus describe the framework only for trained networks.

```

hyperparameters: model family  $\mathcal{F}$ ,
                    dataset family  $\mathcal{D}$ ,
                    model-dataset count  $T$ ,
                    GP parameterization  $\theta$ 
// Step 1 in Fig. 7.1
for  $t \in 1 \dots T$  do
    Sample a model,  $\mathbf{g}_{r_t} \sim \text{Unif}(\mathcal{F})$ 
    Sample a dataset,  $\mathcal{D}_{s_t} \sim \text{Unif}(\mathcal{D})$ 
    Train the model,  $\mathbf{g}_{r_t}^{\text{fit}} \leftarrow \text{train}(\mathbf{g}_{r_t}, \mathcal{D}_{s_t}^{\text{train}})$ 
    Evaluate  $\mathbf{g}_{r_t}^{\text{fit}}(\mathcal{D}_{s_t}^{\text{eval}})$ 
end
// Step 2 in Fig. 7.1
Optimize Objective 7.6 for  $\theta^*$ 
// Step 3 in Fig. 7.1
Analyze  $\theta^*$  via  $P_{\theta^*}$ 

```

Algorithm 7.2: Training and evaluation of the GP surrogate model described in Section 7.3.

$\{\mathcal{D}_0, \dots, \mathcal{D}_S\}$ that share some underlying structure as in multi-task and meta-learning settings (Caruana 1997; Hospedales et al. 2020). We consider supervised learning, in which each dataset consists of inputs and targets, $\mathcal{D} = (\mathbf{X}, \mathbf{y})$. Importantly, we fit surrogate model parameters θ to a *behavioral dataset* of the model family evaluated on the dataset family, and not the ground truth datasets themselves.

Data. We construct a component of the surrogate model training dataset as follows: We sample a model index r and a dataset index s . The corresponding dataset is split into a training set and an evaluation set, $\mathcal{D}_s = \mathcal{D}_s^{\text{train}} \cup \mathcal{D}_s^{\text{eval}}$. The corresponding model \mathbf{g}_r is fit the training set $\mathcal{D}_s^{\text{train}} = (\mathbf{X}_s^{\text{train}}, \mathbf{y}_s^{\text{train}})$ according to the training procedure specified by the choice of model family \mathcal{F} , producing $\mathbf{g}_r^{\text{fit}}$. We then collect the predictions of the trained model on the evaluation set, $\mathbf{g}_r^{\text{fit}}(\mathbf{X}_s^{\text{eval}})$, to produce the component $(\mathbf{X}_s^{\text{eval}}, \mathbf{g}_r^{\text{fit}}(\mathbf{X}_s^{\text{eval}}))$ consisting of the *ground truth inputs* paired with the *neural network behavioral targets* from the evaluation set. We aggregate the ground truth inputs and the neural network behavioral targets across pairs to produce the *surrogate model training dataset*, $((\mathbf{X}_{s_1}^{\text{eval}}, \mathbf{g}_{r_1}^{\text{fit}}(\mathbf{X}_{s_1}^{\text{eval}})), \dots, (\mathbf{X}_{s_T}^{\text{eval}}, \mathbf{g}_{r_T}^{\text{fit}}(\mathbf{X}_{s_T}^{\text{eval}})))$.

Surrogate model. We fit the GP using type-II maximum likelihood estimation. Let $P_\theta(\mathbf{g}^{\text{fit}}(\mathbf{X}^{\text{eval}}) | \mathbf{X}^{\text{eval}})$ be the GP marginal likelihood of the dataset component $(\mathbf{X}^{\text{eval}}, \mathbf{g}^{\text{fit}}(\mathbf{X}^{\text{eval}}))$ under a GP with kernel hyperparameters θ , as given in Eq. (7.3). We fit the surrogate model jointly across model-and-task pairs in the surrogate model training dataset by

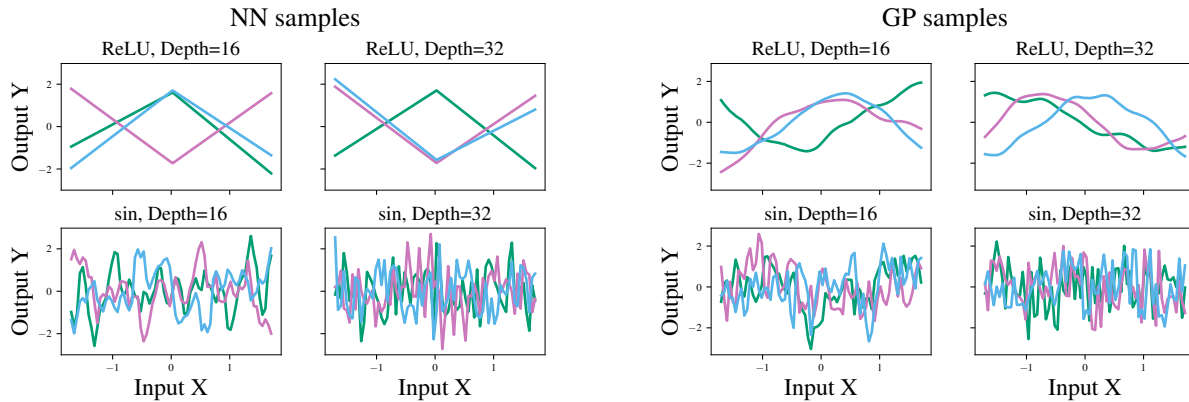


Figure 7.3: Demonstration: Comparing learned GP priors with NN priors. Samples from GP prior (**right**) with kernel hyperparameters inferred from the predictions of NN families (**left**). GPs are flexible enough to capture properties of each NN family; for example, the samples from the learned GP prior reflect the quickly varying behavior of the 32-layer sinusoidal NNs and the increasing-decreasing behavior of rectifier NNs.

maximizing the joint marginal likelihood with respect to θ :

$$\max_{\theta} \prod_{(r,s)} P_{\theta}(\mathbf{g}_r^{\text{fit}}(\mathbf{X}_s^{\text{eval}}) | \mathbf{X}_s^{\text{eval}}). \quad (7.6)$$

By optimizing Objective 7.6, we encourage the kernel hyperparameters θ to capture the implicit prior distribution over functions induced by the models in the family \mathcal{F} as applied to the datasets in the family \mathcal{D} . Algorithm 7.2 gives the complete surrogate model training and evaluation process.

Why use (GP) surrogate models for NNs?

By estimating a prior over functions for a neural network family directly from neural network behavior, we aim to capture shared properties that determine the model family’s behavior on data, *i.e.*, **the model family’s inductive biases**. There is strong evidence that the inductive biases of neural networks (*e.g.*, invariances and equivariances, Markovian assumptions, compositionality) and not just data, play an important role in their performance (Poggio, Mhaskar, et al. 2017; Tiño et al. 2004; Lin and Tegmark 2017; Fukushima 2004; Werbos 1988). Moreover, deep NNs are highly overparametrized models that can nevertheless generalize well, prompting interest in implicit regularization mechanisms that bias NNs towards learning simpler solutions (Soudry et al. 2018; Poggio, Kawaguchi, et al. 2018; Neyshabur, Bhojanapalli, et al. 2017). More broadly, the extrapolation behavior of any learning machine is underdetermined by data alone and therefore depends on its inductive biases (Mitchell 1980).

GPs, in particular, offer several advantages as surrogate models of NNs. Firstly, GPs **are flexible models that are also often interpretable** in the sense that the learned hy-

perparameters can provide insights into properties of the datasets on which they are trained (Wilson and Adams 2013). As an example, many covariance functions have separate lengthscales for each input dimension. An inverse lengthscale captures an input dimension’s “importance;” in Section 7.4, we demonstrate that we can use these lengthscales to predict generalization behavior, suggesting that the GP surrogate representation is practically useful in automating model selection.

Secondly, the use of GP surrogate models is also motivated by the **theoretical connections between GPs and NNs**. Neal (1996) showed that a prior over the parameters of certain single-layer multi-layer perceptrons (MLPs) converges to a GP as the MLP’s width approaches infinity, and recent works (Lee, Bahri, et al. 2017; Matthews et al. 2018; Novak, Xiao, Bahri, et al. 2019; Garriga-Alonso et al. 2019; Yang 2019) have extended this correspondence to deep MLPs and more modern NN architectures. Connections between GPs and NNs can provide insight because they transform the priors implicit in NNs designs into explicit priors expressed through a GP. However, our strategy to derive such a connection differs from this prior theoretical work that derives analytic kernels for limiting cases of NNs—we take an empirical approach by learning GP kernels directly from the predictions of arbitrary classes of finite NNs.

Lastly, **GPs have a tractable marginal likelihood**. Probabilistic models allow us to express inductive biases in the form of an explicit prior distribution, but the marginal likelihood is intractable for most complicated Bayesian models. In contrast, for GPs, the marginal likelihood has a closed form expression, which means that we can optimize it directly instead of resorting to approximations.

Demonstration: Comparing learned GP priors with NN priors

We briefly demonstrate the surrogate modeling framework of Section 7.3. As a simple sanity check, we verify that GP surrogates learned from varying NN families exhibit meaningful variation in behavior. To do this, we learn GP priors from varying NN families and compare the learned priors with the NN families.

NN hyperparameters. We consider ensembles of 50 randomly initialized NNs with ReLU or sin activations and 16 or 32 hidden layers of 128 hidden units each. We randomly initialize the weights about zero with weight variance $\sigma_w^2 = 1.5$ and bias variance $\sigma_b^2 = 0.05$.

GP surrogate. For each ensemble, we learn the hyperparameters of a randomly initialized SMK with $Q = 10$ mixture components by optimizing Objective 7.6 for 350 iterations with batch gradient descent and the adaptive momentum (Adam) optimizer (Kingma and Ba 2015) with a learning rate $\eta = 0.1$. We choose the kernel hyperparameters with the highest objective value across three random initializations.

Results. We plot NN predictions and samples from the learned GP priors in Fig. 7.3. The learned GP captures the periodicity of the sinusoidal neural networks (sinusoidal NNs),

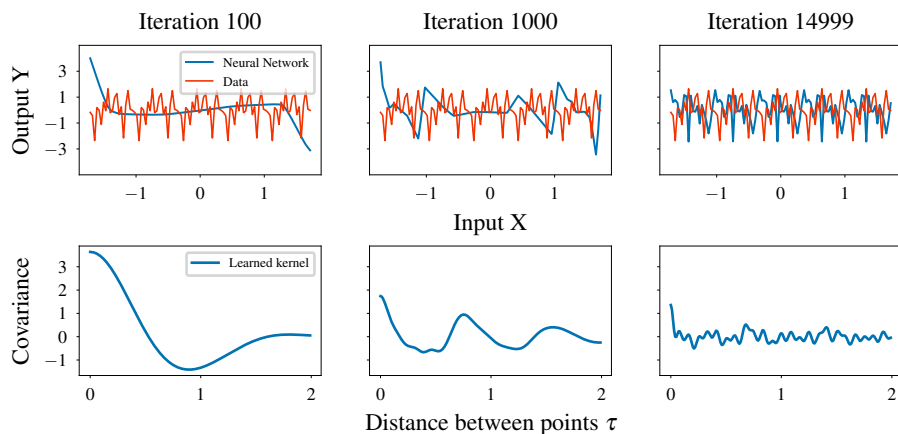


Figure 7.4: Capturing spectral bias in neural networks. (Top) Neural network predictions as training progresses on the sum-of-sines target function described in Section 7.4. (Bottom) Spectral mixture kernel fit to neural network predictions as training progresses. The kernel reveals a spectral bias for this neural network family, with the range of spectral frequencies expressed in the kernel increasing with the number of iterations of training.

and partially captures the increasing-decreasing behavior of rectifier NNs about a cusp; though, due to the SMK parameterization, it cannot capture the discontinuity at the cusp. The GP also captures differences in depths for the sinusoidal NNs: The GP prior samples for the 32-layer networks are quickly varying, indicating shorter lengthscales have been learned. Taken together, the results of this demonstration show that GP surrogates can capture certain NN behavior.

7.4 Experiments

We provide a series of demonstrations of the value of the approach of Section 7.3. Each experiment aims to investigate the properties of one or more neural network families, specified by **neural network (NN) hyperparameters**, as evaluated on one or more dataset families, parameterized as **target functions**, by analyzing the corresponding **Gaussian process (GP) surrogate model**. In Section 7.4, we capture previously established NN phenomena, while in Section 7.4, we predict NN generalization behavior.

Reproduction: Capturing spectral bias in NNs

Rahaman et al. (2019) demonstrated that deep rectifier NNs exhibit *spectral bias*, the preference to learn lower frequencies in the target function before higher frequencies. To demonstrate this, the authors studied the Fourier spectrum of rectifier NNs fit to a sum of sinusoidal functions of varying frequencies. In this section, we take an alterna-

tive approach: We learn kernels from NN predictions at various stages of training and demonstrate that the evolution of these learned kernels captures the spectral bias.

NN hyperparameters. As in Rahaman et al. (2019), we train an NN with 6 hidden layers of 256 units and ReLU activations using full-batch gradient descent with Adam and a learning rate of $\eta = 3 \times 10^{-4}$.

Target function. The target functions are sums of sine functions with frequencies in $(5, 10, \dots, 45, 50)$ and phases drawn from $U(0, 2\pi)$, evaluated at 200 points evenly spaced between $[0, 1]$, as in Rahaman et al. (2019).

GP surrogate. We learn the parameters of a spectral mixture kernel (SMK) with $Q = 10$ mixture components by optimizing Objective 7.6 with Adam for 350 iterations with a learning rate of $\eta = 0.1$. Since the marginal likelihood of the SMK is multi-modal in its frequency parameters, we repeat this optimization for three different random initializations of the kernel parameters and choose the hyperparameters with the largest marginal likelihood value (the value of Objective 7.6). We randomly initialize the length-scales v_i by sampling from a truncated normal distribution whose variance depends on the maximum distance between input points. We set the signal variances w to the variance of the target function values divided by the number of mixture components. The frequency hyperparameters of the SMK are sometimes initialized by sampling from a uniform distribution whose upper limit is the Nyquist frequency (Wilson and Adams 2013); since this target function’s largest frequency is smaller than the Nyquist frequency, we instead set a smaller frequency as the upper limit.

Results. Fig. 7.4 displays the NN predictions and the kernel of the corresponding GP surrogate at different iterations of NN training. The kernel function, which is given in Eq. (7.5), reflects how the similarity between function values varies with the distance between their input points.[‡] The structure of the learned kernel reflects the properties of the NN family: Initially, the learned kernel only captures low frequencies in the NN’s predictions—reflected in the long period of the kernel—consistent with the spectral bias of Rahaman et al. (2019). However, as training progresses, the periodicity of the learned kernel reflects both low and high frequencies.

Reproduction: Depth pathologies in randomly initialized NNs

Hyperparameter selection in NNs is not always theoretically grounded. Many recent studies thus characterize how different hyperparameter choices (*e.g.*, depth, width) affect the properties of NNs at random initialization (Schoenholz et al. 2017; Yang 2019; Xiao

[‡]Since the SMK is a stationary covariance function, we graph against the distance between input points rather than the absolute value of the input points themselves.

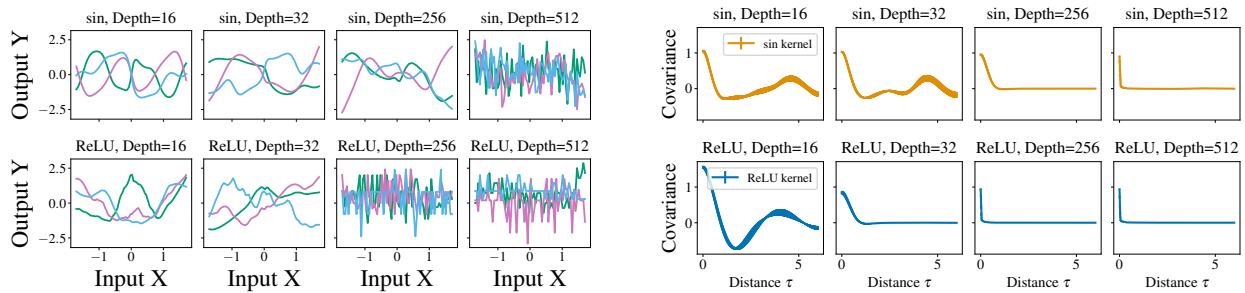


Figure 7.5: Depth pathologies in randomly initialized neural networks. Predictions of neural networks (**left**) from neural network families of different activations (**rows**) and varying depths (**columns**); mean and standard error of the covariance of the corresponding surrogate model kernels (**right**). The covariance is aggregated across 10 kernels learned from 10 different 50-member neural network ensembles from a given family. Greater depth results in kernels with shorter lengthscales, with this pathology emerging earlier in rectifier NNs; this result is consistent with prior work on pathologies of deep neural networks.

et al. 2018). Towards that end, recent work showed that increasing depth could actually induce pathologies in randomly initialized NNs (Labatie 2019; Duvenaud, Rippel, et al. 2014). For example, Duvenaud, Rippel, et al. (2014) proved that increasing depth in a certain class of infinitely wide NNs produces functions with ill-behaved derivatives. As a result, these functions are quickly varying in the input space.

We empirically study a similar pathology—quick variation in input space—that emerges in randomly initialized, finite-width, finite-depth NNs. To do this, we fit GP surrogates to randomly initialized NN ensembles of varying depths and activation functions and inspect how the learned kernels change with depth. If NNs exhibit this pathology, the learned covariance will decay sharply with distance.

NN hyperparameters. We consider families of NNs of varying activation functions ($a \sin(bx + c)$) and ReLU ($\max(0, x)$) and varying depths (from 16 to 512 layers). From each family, we sample an ensemble of 50 randomly initialized NNs, each with 128 hidden units in each layer. We randomly initialize NN weights about zero with weight variance $\sigma_w^2 = 1.5$ and bias variance $\sigma_b^2 = 0.05$.

GP surrogate. We sample 10 ensembles of 50 randomly initialized NNs, and learn an SMK kernel by optimizing Objective 7.6 separately for each ensemble, running Adam (Kingma and Ba 2015) for 750 iterations with a learning rate of $\eta = 0.1$. We choose the kernel hyperparameters with the highest mean marginal likelihood among three random initializations. To ensure our results are robust across random ensembles, we consider an averaged learned kernel: Suppose we have n kernels, $k_1(\cdot), \dots, k_n(\cdot)$, learned from n different ensembles from the same family. The average learned kernel, \bar{k} , is defined as $\bar{k}(\tau) = \frac{1}{n} \sum_{i=1}^n k_i(\tau)$.

Results. Fig. 7.5 plots the average learned kernels for NN families with varying activation functions and depths, as well as the predictions of those NN families. Across both activation functions, the learned kernels reveal a pathology: For large depths, the covariance (Fig. 7.5, right) sharply decays towards zero with distance. The NN predictions (Fig. 7.5, left) explain this property of the learned kernels: At large depths, the deep NNs vary quickly in the input domain, which causes the SMK to learn short lengthscales. Interestingly, this pathology emerges at different depths for different activation functions: We see rectifier NNs exhibit this pathology with 256 layers while sinusoidal NNs exhibit this pathology with 512 layers.

Ranking NN generalization with the GP marginal likelihood

In previous sections, we demonstrated that GP surrogate models could yield insight into NN behavior. The benefits of GPs extend beyond this. Since the GP marginal likelihood has a closed form expression, many have advocated for using the marginal likelihood in model selection and as an indicator of expected generalization performance (MacKay 1992a). In this section, we leverage the learned GP surrogate to *rank NNs by their generalization error with the GP marginal likelihood*. In particular, we learn GP surrogates from different NNs at random initialization, and we then study if the marginal likelihood of the surrogates can rank the NNs by test error after training. In the following experiments with varying classes of NN families, we find that we can indeed predict test error using the marginal likelihood of the training set under the learned surrogate GP.

The idealized case: Large-width NNs

Before we consider arbitrary NN families, we check that the marginal likelihood is predictive in an idealized setting. In particular, we consider large-width NNs whose infinite-width analogs are equivalent to GPs (Lee, Bahri, et al. 2017). If the marginal likelihood is not predictive in this case in which the kernel function can be analytically determined, it is unlikely to be useful in a general setting where the kernel is learned and GPs approximate NNs priors but are not equivalent.

NN hyperparameters. We consider NNs with sin or Gauss error function (erf)[§] activations and 2 hidden layers of 1024 units each. We randomly initialize the weights about zero with weight variance $\sigma_w^2 = 1.5$ and bias variance $\sigma_b^2 = 0.05$. We train an ensemble of 50 randomly initialized NNs from each family using full-batch (vanilla) gradient descent with learning rates of $\eta \in \{0.01, 0.1\}$.

Target function. The target function is $\sin(0.5x)$.

[§]Here, erf is defined as $a \operatorname{erf}(bx) + c$, where $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$.

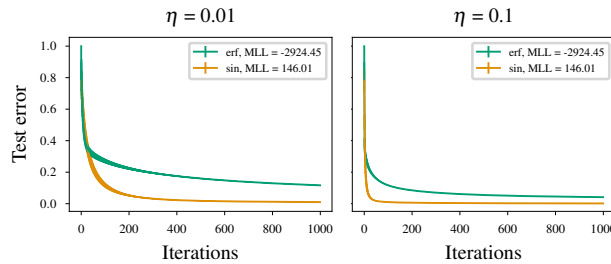


Figure 7.6: Ranking generalization from MLL in large-width NNs. Mean and standard error of the test MSE of large-width sinusoidal and erf NNs trained with learning rates $\eta = 0.01$ (left) and $\eta = 0.1$ (right) on the target function of Section 7.4. The MLL of the target function under the surrogate model corresponding to the limiting kernel for each model family is shown in the legend. Consistent with expectations, the model family whose surrogate assigns higher MLL to the target function achieves lower test error for both values of η .

GP surrogate. We do not learn a kernel from NN predictions as in previous sections. Instead, we use the kernels corresponding to the infinite width analogs of the NNs using the neural-tangents package (Novak, Xiao, Hron, et al. 2020).

Results. Fig. 7.6 compares the performance of these NN families along with the marginal likelihood of the target function under the surrogate model. The performance (mean-squared error (MSE) on the test set) is averaged across each ensemble of NNs. The marginal log-likelihood (MLL) of the target function is higher for the better-performing NN family.

Small width neural networks and learning the kernel

In the previous experiment, we showed that the marginal likelihood could be predictive when we consider large-width NNs and when we use a corresponding, analytically derived kernel. Is the marginal likelihood predictive when we consider smaller-width NNs and when we learn the kernel empirically?

NN hyperparameters. We consider ensembles of width 16, depth 4 NNs from two families: NNs with sin activations and NNs with ReLU activations. We randomly initialize weights about zero with weight variance $\sigma_w^2 = 1.5$ and bias variance $\sigma_b^2 = 0.05$. We train an ensemble of 50 randomly initialized NNs from each family on the target functions using full-batch gradient descent with a learning rate of $\eta = 0.1$.

Target function. The target function families mirror the NN model families: We collect predictions from randomly initialized, width 16, depth 4 NNs with sin or ReLU activations. These target functions are a useful sanity check, as the inductive biases of the model families are perfectly suited for a target function family.

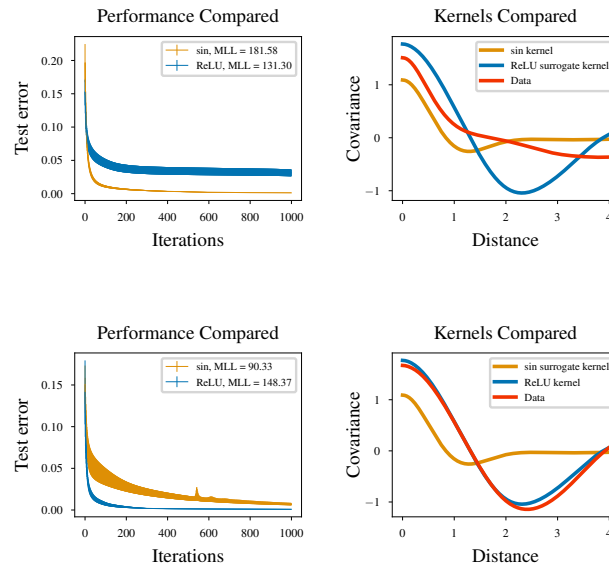


Figure 7.7: Ranking generalization from MLL in small-width NNs. Mean and standard error of test MSE (**left**) of small-width sinusoidal and rectifier NN ensembles on sin (**top**) and ReLU (**bottom**) target function families, with the target function MLL under the surrogate learned from each model family in the legend. Covariance (**right**) of surrogate kernels alongside data kernels learned from the sin (**top**) and ReLU (**bottom**) target function families. Even in the small-width regime and when the kernel is learned, the model family whose surrogate assigns a higher MLL to the target function attains lower error (**left**); the surrogate kernel learned from the better-performing model family better matches the data kernel (**right**).

GP surrogate. For each ensemble, we learn the hyperparameters of an SMK with $Q = 5$ mixture components by optimizing Objective 7.6 across the ensemble. To optimize, we randomly initialize the kernel hyperparameters and run Adam for 250 iterations with a learning rate of $\eta = 0.1$. We initialize the frequency parameters by sampling from a uniform distribution whose upper limit is the Nyquist frequency. We choose the kernel hyperparameters with the highest objective value across three random initializations.

Results. In Fig. 7.7, we compare the performances of the two NN families on the two target function families. We also display the kernels learned from NN behavior (*sin surrogate kernel* or *ReLU surrogate kernel*) and learned from the target function family (*data kernel*) directly. Across both experiments, the MLL averaged across the target function family of the better-performing NN family is higher. In general, the structure of a learned kernel reflects the properties of the learned GP prior, and so we can compare kernels to assess similarity between target function and NN families. We see that the data kernel provides a better qualitative match to the kernel of the better-performing model family.

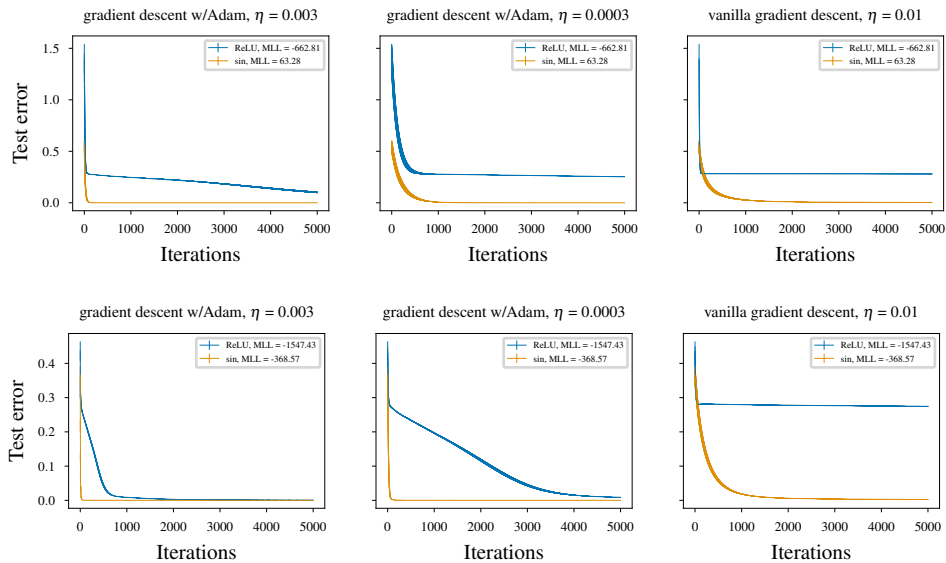


Figure 7.8: Ranking generalization performance from MLL across different learning algorithms and architectures. Each panel displays mean and standard error of test MSE of an NN family trained on the target function $\sin(0.5x)$ with noise; legend displays MLL of the training data under the surrogate for one of two NN families: 1-layer (256 hidden units) sinusoidal or rectifier NNs (**top**); 3-layer (256 hidden units) sinusoidal or rectifier NNs (**bottom**). NNs are trained with batch gradient descent with Adam (learning rates $\eta = 0.0003$, $\eta = 0.0003$) or vanilla batch gradient descent ($\eta = 0.01$). Across architectures and learning algorithms, the NN family whose surrogate assigns higher MLL to the target function achieves lower test error.

Systematic study of various learning rates and architectures

In this last experiment on ranking generalization performance, we establish that Gaussian process surrogates reliably rank performance across a range of learning rates and gradient descent algorithms.

NN hyperparameters. We consider ensembles of randomly initialized NNs with sin or ReLU activations and 1 or 3 hidden layers with 256 hidden units in each layer. We randomly initialize the weights about zero with weight variance $\sigma_w^2 = 1.5$ and bias variance $\sigma_b^2 = 0.05$. We train 50 randomly initialized NNs from each family using either vanilla full-batch gradient descent with a constant learning rate of $\eta = 0.01$, or Adam (Kingma and Ba 2015) using learning rates of $\eta \in \{0.0003, 0.003\}$.

Target function. We consider a target function of $\sin(0.5x)$.

GP surrogate. For each ensemble, we learn the hyperparameters of an SMK with $Q = 5$ mixture components by optimizing Objective 7.6 across the ensemble. To optimize, we

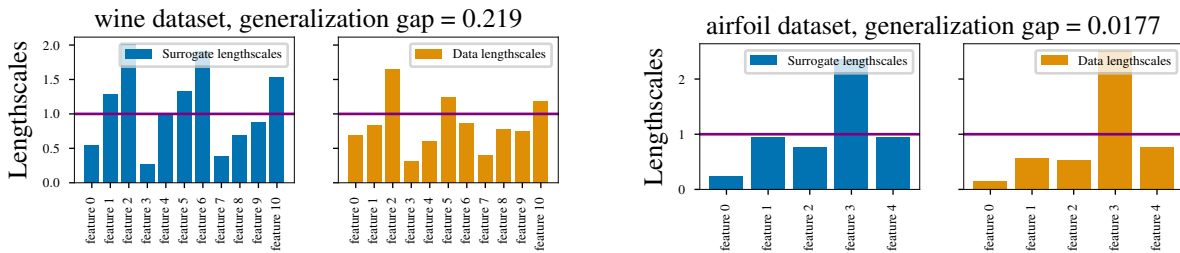


Figure 7.9: Qualitative connection between lengthscale profile discrepancy and generalization gap. Each subfigure compares normalized lengthscales learned from neural network predictions on validation set (*i.e.*, surrogate lengthscales) after training and normalized lengthscales learned from training data (*i.e.*, data lengthscales). A lengthscale greater than 1 indicates an “unimportant” feature. The title indicates the UCI dataset and generalization gap defined in Fig. 7.10. Data and surrogate lengthscales for some features are different (*e.g.*, features 1, 4, 6), reflected in a high generalization gap (**left**). Data and surrogate lengthscales for the same features are generally similar, reflected in a low generalization gap (**right**). This suggests a connection between the generalization gap and discrepancy between surrogate and data lengthscales.

randomly initialize the kernel hyperparameters and run Adam for 250 iterations with a learning rate of $\eta = 0.1$. We choose the kernel hyperparameters with the highest objective value across three random initializations. To randomly initialize the frequency parameters, we uniformly sample from the real-valued interval $(0, 25]$.

Results. In Fig. 7.8, we find that the marginal likelihood of the better-performing NN family is higher. The marginal likelihood depends on the diagonal noise σ_n^2 added to the Gram matrix (Eq. (7.3)). We find that our results are robust across three levels of this diagonal noise ($10^{-3}, 10^{-4}, 10^{-5}$). These results suggest we can rank these NN families when they are not in the asymptotic regime and when we learn the kernel, in contrast to Section 7.4, as well as when *a priori* no model family should perform better, unlike Section 7.4.

Predicting the NN generalization gap with the GP marginal likelihood

In the previous section, we predicted generalization using kernels learned from randomly initialized NNs. However, some design choices do not affect NN properties at random initialization but may still strongly influence generalization (*e.g.*, learning algorithm). Motivated by this, we characterize trained NN properties on the *validation set* and compare these properties to the training data. We focus on the validation set because it is more informative of extrapolation. If the NN extrapolates well, its predictions on the validation set should be “similar” in some sense to the dataset. On the other hand, significant discrepancies could indicate poor extrapolation. This intuition motivates our analysis.

In particular, we learn a kernel from the training data and a kernel from NN predictions on a validation set. We then quantitatively compare these kernels by computing

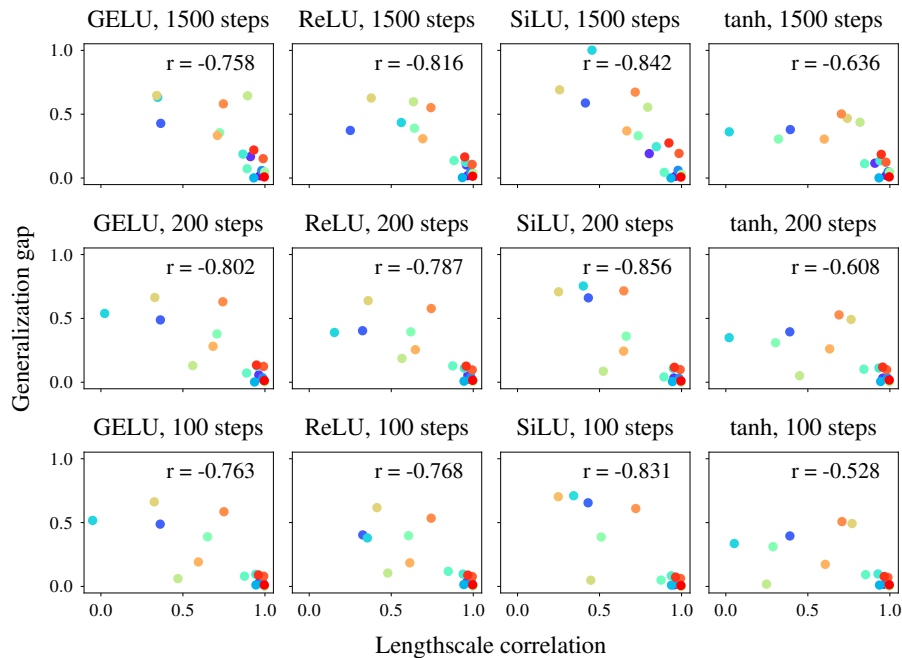


Figure 7.10: Inverse relationship between generalization error and lengthscale correlation on UCI datasets. Each point represents the lengthscale correlation (between surrogate and data lengthscales) and the generalization gap for a neural network ensemble to which the surrogate model is fit, on a single UCI dataset. Each panel corresponds to a particular neural family; see Section 7.4 for details about hyperparameters of these families, including architectures. Across datasets and architectures, a larger lengthscale correlation (*i.e.*, higher similarity between the data and surrogate representations) corresponds to a lower generalization gap (*i.e.*, better extrapolation).

a metric we describe in more detail later. We find that a lower similarity between these kernels correlates with a larger *generalization gap* (*i.e.*, poorer extrapolation), defined as the difference between test error and training error (*e.g.*, Jiang et al. 2020).

NN hyperparameters. We train ensembles of randomly initialized NNs with sigmoid-weighted linear unit (SiLU) (Elfwing et al. 2018), Gaussian error linear unit (GELU) (Hendrycks and Gimpel 2016), ReLU (Fukushima 1975; Nair and Hinton 2010), or hyperbolic tangent (tanh) activations, and two layers of 128 hidden units. We use the LeCun normal initialization with a scale of 1.5 (LeCun, Bottou, et al. 2012). We train 25 NNs with full-batch gradient descent using Adam with a learning rate of $\eta = 0.003$. We want to assess if our approach can distinguish between NNs with similar training behavior but varying generalization performance. We train NNs either for a maximum number of iterations, a hyperparameter, or until training error reaches zero.

Target functions. We consider a set of naturalistic regression tasks from the UC Irvine Machine Learning Repository (UCI) dataset (Dua and Graff 2017), spanning a range of dataset sizes and input dimensions. We split each of the datasets into a 72/8/20 train/validation/test split. Both the data input and output are standardized by mean-centering and dividing by the standard deviation dimension-wise so that the target values and each dimension of the data input have near zero mean and unit variance. We subsample 2,000 datapoints for datasets with more than 2,000 datapoints, as in Simpson et al. (2021) and Liu, Sun, et al. (2020).

GP surrogate. We learn a *data kernel* directly from the training dataset. We also learn a *surrogate kernel* from NN predictions on the validation set. In both cases, we use the Matérn kernel (MK) since the SMK can struggle for higher-dimensional inputs. We learn a separate lengthscale for each input dimension (*i.e.*, feature) of the data. We denote the lengthscales for a kernel as its *lengthscale profile*. We call the data kernel’s lengthscales the *data lengthscales* and the surrogate kernel’s lengthscales the *surrogate lengthscales*. To quantify the mismatch between NN validation predictions and the training data, we consider the *correlation in lengthscale profiles across features*. This is the correlation between the data and surrogate lengthscales.

Results. Fig. 7.9 gives intuition for our more general result in Fig. 7.10. For two UCI datasets, we compare the data lengthscales and the surrogate lengthscales for a two-layer GELU NN. The vertical axis corresponds to (normalized) learned lengthscales for each input dimension.[¶] When the generalization gap is small, the data kernel and surrogate kernel are similar; the same features have similar lengthscales (Fig. 7.9, right). When the generalization gap is large, the data kernel and surrogate kernel have discrepancies. For example, the surrogate lengthscales for features 1 and 6 are larger than 1, but the data lengthscales for feature 1 and 6 are smaller than 1 (Fig. 7.9, left).

In Fig. 7.10, we summarize our results across different architectures, datasets, and maximum training iterations. We display the generalization gap against the correlation in lengthscale profiles across features. The similarity in lengthscale profiles negatively correlates with generalization gap across a range of architectures and max iterations. The Pearson correlation coefficients range from -0.856 to -0.528. In Appendix B.3, we additionally demonstrate that these results are insensitive to outlier datasets by performing a dataset-sensitivity analysis.

[¶]For this visualization, we divide the learned lengthscale for each dimension by the difference between the maximum feature value and minimum feature value for each dimension. By doing so, we can interpret a lengthscale that is much greater than 1 as suggesting that the NN predictions do not vary much along that dimension.

7.5 Discussion

In this paper, we illustrated the potential of modeling neural networks with Gaussian process surrogates. We empirically characterized phenomena in neural networks by interpreting kernels learned directly from neural network predictions, capturing the spectral bias of deep rectifier networks (Section 7.4) and pathological behavior in deep, randomly initialized neural networks (Section 7.4). We further demonstrated that Gaussian process surrogates could predict neural network generalization by ranking test error performance by marginal (Section 7.4) and by quantifying the generalization gap via a surrogate-data kernel discrepancy (Section 7.4). Taken together, these results suggest that Gaussian process surrogates may be a valuable empirical tool for investigating deep learning, and future work could aim to use this framework to complement existing approaches to interpretability (*e.g.*, Ribeiro et al. 2016) and extrapolation (*e.g.*, Xu, Zhang, et al. 2021).

We note a couple of limitations of our current study. First, though the framework is in principle applicable to broader settings, we restricted this first exploration to regression tasks and feed-forward neural network architectures. A broader study of more architectures on more types of tasks would be challenging due to the need to scale Gaussian processes but potentially rewarding, as characterizing properties of neural networks as used in practice is a significant open problem with far-reaching implications (Sejnowski 2020). Second, we learn point estimates of kernel hyperparameters (type II maximum likelihood; Gelman et al. 2013). Although this is standard, we could infer the posterior over hyperparameters using Markov chain Monte Carlo (MCMC) or variational inference (Lalchand and Rasmussen 2020; Murray and Adams 2010; Simpson et al. 2021) to perform a fully Bayesian analysis. We also could explore a richer set of kernels, such as compositional kernels (Duvenaud, Lloyd, et al. 2013). These directions are exciting avenues for future work.

Part IV

Chapter 8

Conclusion

Modern machine learning systems, including deep-learning models, provide new flexibility to build models and solve problems in applied and scientific domains. However, this flexibility comes with its own cost, as it is now increasingly difficult to understand the assumptions that underlie these systems. Understanding implicit inductive biases of machine learning systems is therefore crucial for scientists interested in using machine learning as a modeling tool in the service of scientific insights; this understanding also benefits machine learning practitioners interested in improving interpretability and predictability in applied systems. This dissertation has advanced *cognitive analyses* as a means of addressing this challenge, and has developed new connections between paradigms in cognitive science and machine learning (Part II) that enable behavioral (Part I) and computational (Part III) studies of machine learning systems. In this final section, I highlight some ensuing directions for future work.

Collecting richer behavioral data from both humans and machines. As argued in Part I, extrapolation behavior in carefully designed probe tasks is much more revealing of inductive biases than simple metrics like accuracy. As another example, recent work (Langlois et al. 2021) operationalized notions of visual selectivity that are explicitly comparable between cognitive and machine learning systems in order to understand whether machine learning systems and humans use similar image regions to make classification decisions. These works are a start to capturing richer notions of human behavior than performance or accuracy. In turn, these richer notions can be used as a point of comparison for machine learning systems—or as an explicit objective to bring machine learning systems closer to human cognition—and as a starting point for a computational model of a cognitive phenomenon. Future work has the opportunity to develop richer experimental protocols that probe specific facets of inductive biases in both machine learning and human cognitive systems.

Complementing behavioral and black-box analyses of machine learning systems. Examining inductive biases in machine learning systems on the basis of behavior alone allows us to avoid the simplifying assumptions present in much theoretical work investigating implicit inductive bias. However, the behavioral and black-box approaches advanced in Part I and Part III lose the ability to make use of detailed knowledge about the design specification of the system as well as internal representations that are available for analysis (*e.g.*, in the context of neural networks, intermediate-layer activations, or the effect of targeted perturbations). Future work has the opportunity to make use of more detailed knowledge about the machine learning system under investigation in order to analyze its inductive biases more precisely. Understanding machine learning systems at this level of analysis—closer to neuroscience than higher-level cognition—would allow us to render explicit assumptions that impact the use of machine learning systems as computational models in scientific applications, including cognitive science.

Re-centering inductive bias as a design principle. Because of the fundamental relationship between inductive bias and extrapolation, we can often guarantee extrapolation behavior in machine learning systems by designing or determining the inductive biases that are at play. Much classical and recent work in machine learning has aimed to control such inductive biases explicitly in the design process by using, for example, object-centric models to express physical constraints (Chang et al. 2017) or graph neural networks to express relational constraints (Van Steenkiste et al. 2018). However, current machine learning systems are incredibly complex, and there are factors beyond targeted design choices that control inductive biases. Future work can make progress towards a *causal* understanding of how design decisions correspond to inductive bias specification—for example, how a particular neural network architecture might increase or decrease the propensity to memorize individual datapoints as characterized in Chapter 3. This understanding has consequences for designing robust and reliable systems in science and engineering.

Bibliography

- Abadi, Martín et al. (2016). “Tensorflow: A system for large-scale machine learning”. In: *USENIX Symposium on Operating Systems Design and Implementation*, pp. 265–283.
- Abbott, Joshua T, Joseph L Austerweil, and Thomas L Griffiths (2012). “Constructing a hypothesis space from the Web for large-scale Bayesian word learning”. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*.
- Adebayo, Julius, Michael Muelly, Harold Abelson, and Been Kim (2022). “Post hoc explanations may be ineffective for detecting unknown spurious correlation”. In: *Proceedings of the International Conference on Learning Representations*.
- Allen, Scott W and Lee R Brooks (1991). “Specializing the operation of an explicit rule”. In: *Journal of experimental psychology: General* 120.1, p. 3.
- Allen-Zhu, Zeyuan, Yuanzhi Li, and Zhao Song (2019). “A convergence theory for deep learning via over-parameterization”. In: *Proceedings of the International Conference on Machine Learning*, pp. 242–252.
- Alspach, Daniel and Harold Sorenson (1972). “Nonlinear Bayesian estimation using Gaussian sum approximations”. In: *IEEE Transactions on Automatic Control* 17.4, pp. 439–448.
- Amari, Shun-Ichi (1998). “Natural gradient works efficiently in learning”. In: *Neural Computation* 10.2, pp. 251–276.
- Andreas, Jacob (2019). “Measuring compositionality in representation learning”. In: *arXiv preprint arXiv:1902.07181*.
- Andreas, Jacob, Marcus Rohrbach, Trevor Darrell, and Dan Klein (2016). “Neural module networks”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 39–48.
- Andrychowicz, Marcin, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, and Nando de Freitas (2016). “Learning to learn by gradient descent by gradient descent”. In: *Advances in Neural Information Processing Systems*, pp. 3981–3989.
- Arjovsky, Martin, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz (2019). “Invariant risk minimization”. In: *arXiv preprint arXiv:1907.02893*.
- Arora, Sanjeev, Nadav Cohen, Wei Hu, and Yuping Luo (2019). “Implicit regularization in deep matrix factorization”. In: *Advances in Neural Information Processing Systems* 32.

- Ashby, F Gregory and Leola A Alfonso-Reese (1995). “Categorization as probability density estimation”. In: *Journal of mathematical psychology* 39.2, pp. 216–233.
- Ashby, F Gregory and James T Townsend (1986). “Varieties of perceptual independence”. In: *Psychological Review* 93.2, p. 154.
- Bakker, Bart and Tom Heskes (2003). “Task clustering and gating for Bayesian multitask learning”. In: *Journal of Machine Learning Research* 4.May, pp. 83–99.
- Baron-Cohen, Simon (1989). “The autistic child’s theory of mind: A case of specific developmental delay”. In: *J. of Child Psychology & Psychiatry* 30.2, pp. 285–297.
- Baron-Cohen, Simon, Alan M Leslie, and Uta Frith (1985). “Does the autistic child have a “theory of mind”?” In: *Cognition* 21.1, pp. 37–46.
- Barrett, David, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap (2018). “Measuring abstract reasoning in neural networks”. In: *Proceedings of the International Conference on Machine Learning*, pp. 511–520.
- Bauer, Matthias, Mateo Rojas-Carulla, Jakub Bartłomiej Świkatkowski, Bernhard Schölkopf, and Richard E Turner (2017). “Discriminative k-shot learning using probabilistic models”. In: *arXiv preprint arXiv:1706.00326*.
- Baxter, Jonathan (1997). “A Bayesian/information theoretic model of learning to learn via multiple task sampling”. In: *Machine Learning* 28.1, pp. 7–39.
- (2000). “A model of inductive bias learning”. In: *Journal of Artificial Intelligence Research* 12.1, pp. 149–198.
- Beery, Sara, Grant Van Horn, and Pietro Perona (2018). “Recognition in terra incognita”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 456–473.
- Bengio, Samy, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei (1992). “On the optimization of a synaptic learning rule”. In: *Proceedings of the Conference on Optimality in Artificial and Biological Neural Networks*. University of Texas, pp. 6–8.
- Bengio, Yoshua, Samy Bengio, and Jocelyn Cloutier (1991). “Learning a synaptic learning rule”. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. Vol. 2. Seattle.
- Bernardo, J.M. and A.F.M. Smith (2006). *Bayesian theory*. Wiley Series in Probability and Statistics. John Wiley & Sons Canada, Limited.
- Besag, Julian (1986). “On the statistical analysis of dirty pictures”. In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 259–302.
- Bishop, Christopher M (1995). “Regularization and complexity control in feed-forward networks”. In: *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pp. 141–148.
- Blei, David M, Michael I Jordan, et al. (2006). “Variational inference for Dirichlet process mixtures”. In: *Bayesian analysis* 1.1, pp. 121–143.
- Bommasani, Rishi et al. (2021). “On the opportunities and risks of foundation models”. In: *arXiv: 2108.07258*.
- Bottou, Léon and Vladimir Vapnik (1992). “Local learning algorithms”. In: *Neural Computation* 4.6, pp. 888–900.

- Bretherton, Inge and Marjorie Beeghly (1982). “Talking about internal states: The acquisition of an explicit theory of mind”. In: *Developmental Psychology* 18.6, p. 906.
- Brown, Gavin, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar (2020). “When is memorization of irrelevant training data necessary for high-accuracy learning?” In: *arXiv preprint arXiv:2012.06421*.
- Campero, Andres, Andrew Francl, and Joshua B Tenenbaum (2017). “Learning to learn visual object categories by integrating deep learning with hierarchical Bayes”. In: *CogSci*.
- Camps-Valls, Gustau et al. (2015). “Ranking drivers of global carbon and energy fluxes over land”. In: *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, pp. 4416–4419.
- Carey, Susan (1978). “The child as word learner”. In: *Linguistic Theory and Psychological Reality*. Ed. by Morris Halle, Joan Bresnan, and George Miller. MIT Press, pp. 264–293.
- Caruana, Rich (1997). “Multitask learning”. In: *Machine Learning* 28, pp. 41–75.
- (1998). “Multitask learning”. In: *Learning to learn*. Springer, pp. 95–133.
- Caruana, Richard (1993). “Multitask Learning: A Knowledge-Based Source of Inductive Bias”. In: *Proceedings of the International Conference on Machine Learning*.
- Chang, Michael B, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum (2017). “A compositional object-based approach to learning physical dynamics”. In: *Proceedings of the International Conference on Learning Representations*.
- Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton (2020). “A simple framework for contrastive learning of visual representations”. In: *Proceedings of the International Conference on Machine Learning*. Pmlr, pp. 1597–1607.
- Chomsky, Noam (1980). “Rules and representations”. In: 3.1, pp. 1–15.
- Collins, Anne GE and Michael J Frank (2013). “Cognitive control over learning: Creating, clustering, and generalizing task-set structure”. In: *Psychological review* 120.1, p. 190.
- D’Amour, Alexander et al. (2020). “Underspecification presents challenges for credibility in modern machine learning”. In: *arXiv: 2011.03395*.
- Dasgupta, Ishita, Demi Guo, Samuel J Gershman, and Noah D Goodman (2020). “Analyzing machine-learned representations: A natural language case study”. In: *Cognitive Science* 44.12, e12925.
- Daumé III, Hal (2009). “Bayesian multitask learning with latent hierarchies”. In: *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 135–142.
- Dehghani, Mostafa, Yi Tay, Alexey A Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals (2021). “The benchmark lottery”. In: *arXiv: 2107.07002*.
- Deleu, Tristan and Yoshua Bengio (2018). “The effects of negative adaptation in model-agnostic meta-learning”. In: *arXiv preprint arXiv:1812.02159*.
- Dempster, Arthur P, Nan M Laird, and Donald B Rubin (1977). “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38.

- Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). “ImageNet: A large-scale hierarchical image database”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. Ieee, pp. 248–255.
- Doshi-Velez, Finale and Been Kim (2017). “Towards a rigorous science of interpretable machine learning”. In: arXiv: 1702.08608.
- Du, Simon S, Xiyu Zhai, Barnabas Poczos, and Aarti Singh (2019). “Gradient descent provably optimizes over-parameterized neural networks”. In: *Proceedings of the International Conference on Learning Representations*.
- Dua, Dheeru and Casey Graff (2017). *UCI Machine Learning Repository*.
- Duvenaud, David, James Lloyd, Roger Grosse, Joshua Tenenbaum, and Ghahramani Zoubin (2013). “Structure discovery in nonparametric regression through compositional kernel search”. In: *Proceedings of the International Conference on Machine Learning*.
- Duvenaud, David, Dougal Maclaurin, and Ryan Adams (2016). “Early stopping as nonparametric variational inference”. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*. Cadiz, Spain, pp. 1070–1077.
- Duvenaud, David, Oren Rippel, Ryan P. Adams, and Zoubin Ghahramani (2014). “Avoiding pathologies in very deep networks”. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*.
- Dziugaite, Gintare Karolina, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M Roy (2020). “In search of robust measures of generalization”. In: *Advances in Neural Information Processing Systems* 33, pp. 11723–11733.
- Edwards, Harrison and Amos Storkey (2017). “Towards a Neural Statistician”. In: *Proceedings of the International Conference on Learning Representations*. Toulon, France.
- Elfwing, Stefan, Eiji Uchibe, and Kenji Doya (2018). “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning”. In: *Neural Networks* 107, pp. 3–11.
- Fei-Fei, Li et al. (2003). “A Bayesian approach to unsupervised one-shot learning of object categories”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. Ieee, pp. 1134–1141.
- Feldman, Vitaly (2020). “Does learning require memorization? A short tale about a long tail”. In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 954–959.
- Feldman, Vitaly and Chiyuan Zhang (2020). “What neural networks memorize and why: Discovering the long tail via influence estimation”. In: *Advances in Neural Information Processing Systems* 33.
- Fellbaum, Christiane (1998). *WordNet*. Wiley Online Library.
- Finn, Chelsea, Pieter Abbeel, and Sergey Levine (2017). “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”. In: *Proceedings of the International Conference on Machine Learning*. Sydney, Australia.
- Finn, Chelsea, Kelvin Xu, and Sergey Levine (2018). “Probabilistic model-agnostic meta-learning”. In: *Advances in Neural Information Processing Systems*, pp. 9516–9527.

- Fisher, Ronald Aylmer (1925). "Theory of statistical estimation". In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 22. 5. Cambridge University Press, pp. 700–725.
- Fukushima, Kunihiro (1975). "Cognitron: A self-organizing multilayered neural network". In: *Biological Cybernetics* 20, pp. 121–136.
- (2004). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36, pp. 193–202.
- Gao, Jing, Wei Fan, Jing Jiang, and Jiawei Han (2008). "Knowledge transfer via multiple model local structure mapping". In: *Proceedings of the 14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. Acm, pp. 283–291.
- Garnelo, Marta and Murray Shanahan (2019). "Reconciling deep learning with symbolic artificial intelligence: Representing objects and relations". In: *Current Opinion in Behavioral Sciences* 29. Artificial Intelligence, pp. 17–23.
- Garriga-Alonso, Adrià, Carl Edward Rasmussen, and Laurence Aitchison (2019). "Deep convolutional networks as shallow Gaussian processes". In: *Proceedings of the International Conference on Learning Representations*.
- Geirhos, Robert, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann (2020). "Shortcut learning in deep neural networks". In: *Nature Machine Intelligence* 2, pp. 665–673.
- Geirhos, Robert, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel (2019). "ImageNet-trained CNNs are biased towards texture; Increasing shape bias improves accuracy and robustness". In: *Proceedings of the International Conference on Learning Representations*.
- Gelman, Andrew, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin (2013). *Bayesian data analysis*. Chapman and Hall/CRC.
- (2014). *Bayesian data analysis*. Vol. 2. Chapman and Hall.
- Gershman, Samuel J and David M Blei (2012). "A tutorial on Bayesian nonparametric models". In: *Journal of Mathematical Psychology* 56.1, pp. 1–12.
- Gershman, Samuel J, David M Blei, and Yael Niv (2010). "Context, learning, and extinction". In: *Psychological review* 117.1, p. 197.
- Ghahramani, Zoubin and Matthew J Beal (2000). "Variational inference for Bayesian mixtures of factor analysers". In: *Advances in neural information processing systems*, pp. 449–455.
- Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 249–256.
- Golub, Gene H. and Charles F. Van Loan (1983). *Matrix computations*. The Johns Hopkins University Press.
- Golubeva, Anna, Guy Gur-Ari, and Behnam Neyshabur (2020). "Are wider nets better given the same number of parameters?" In: *Proceedings of the International Conference on Learning Representations*.

- Goodman, Noah D et al. (2006). “Intuitive theories of mind: A rational approach to false belief”. In: *Proceedings of Cog. Sci.* Pp. 1382–1387.
- Gordon, Jonathan, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard Turner (2019). “Meta-learning probabilistic inference for prediction”. In: *Iclr*.
- Grant, Erin, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths (2018). “Recasting gradient-based meta-learning as hierarchical Bayes”. In: *Proceedings of the International Conference on Learning Representations*.
- Greff, Klaus, Sjoerd van Steenkiste, and Jürgen Schmidhuber (2017). “Neural expectation maximization”. In: *Advances in Neural Information Processing Systems*, pp. 6694–6704.
- Griffiths, Thomas L, Nick Chater, Charles Kemp, Amy Perfors, and Joshua B Tenenbaum (2010). “Probabilistic models of cognition: Exploring representations and inductive biases”. In: *Trends in Cognitive Sciences* 14.8, pp. 357–364.
- Gupta, Sunil, Dinh Phung, and Svetha Venkatesh (2013). “Factorial multi-task learning: a Bayesian nonparametric approach”. In: *Proceedings of the International Conference on Machine Learning*, pp. 657–665.
- Haixiang, Guo, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing (2017). “Learning from class-imbalanced data: Review of methods and applications”. In: *Expert Systems with Applications* 73, pp. 220–239.
- Hariharan, Bharath and Ross Girshick (2017). “Low-shot visual object recognition”. In: *Proceedings of the International Conference on Computer Vision*.
- Hawkins, Robert, Takateru Yamakoshi, Thomas L Griffiths, and Adele Goldberg (2020). “Investigating representations of verb bias in neural language models”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 4653–4663.
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 770–778.
- Henaff, Mikael, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun (2017). “Tracking the World State with Recurrent Entity Networks”. In: *Proceedings of the International Conference on Learning Representations*.
- Hendrycks, Dan, Steven Basart, et al. (2021). “The many faces of robustness: a critical analysis of out-of-distribution generalization”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349.
- Hendrycks, Dan and Kevin Gimpel (2016). “Gaussian error linear units (GELUs)”. In: arXiv: 1606.08415.
- Hermann, Katherine L, Ting Chen, and Simon Kornblith (2019). “The origins and prevalence of texture bias in convolutional neural networks”. In: *arXiv preprint arXiv:1911.09071*.
- Heskes, Tom (1998). “Solving a huge number of similar tasks: A combination of multi-task learning and a hierarchical Bayesian approach”. In: *Proceedings of the International Conference on Machine Learning*, pp. 233–241.
- Hinton, Geoffrey E and James A Anderson (1981). *Parallel models of associative memory*. Lawrence Erlbaum Associates.

- Hinton, Geoffrey E and Ruslan R Salakhutdinov (2006). “Reducing the dimensionality of data with neural networks”. In: *Science* 313.5786, pp. 504–507.
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural Computation* 9.8, pp. 1735–1780.
- Hochreiter, Sepp, A Younger, and Peter Conwell (2001). “Learning to learn using gradient descent”. In: *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pp. 87–94.
- Hospedales, Timothy M, Antreas Antoniou, Paul Micaelli, and Amos J Storkey (2020). “Meta-learning in neural networks: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Hudson Kam, Carla L and Elissa L Newport (2005). “Regularizing unpredictable variation: The roles of adult and child learners in language formation and change”. In: *Language learning and development* 1.2, pp. 151–195.
- Hughes, Michael C, Emily Fox, and Erik B Sudderth (2012). “Effective split-merge monte carlo methods for nonparametric models of sequential data”. In: *Advances in neural information processing systems*, pp. 1295–1303.
- Ilyas, Andrew, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry (2022). “Datamodels: Understanding predictions with data and data with predictions”. In: *Proceedings of the International Conference on Machine Learning*, pp. 9525–9587.
- Ilyas, Andrew, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry (2019). “Adversarial examples are not bugs, they are features”. In: *Advances in Neural Information Processing Systems*.
- Ioffe, Sergey and Christian Szegedy (2015). “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *Proceedings of the International Conference on Machine Learning*, pp. 448–456.
- Jacot, Arthur, Franck Gabriel, and Clément Hongler (2018). “Neural tangent kernel: Convergence and generalization in neural networks”. In: *Advances in Neural Information Processing Systems*. Vol. 31.
- Jain, Sonia and Radford M Neal (2004). “A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model”. In: *Journal of computational and Graphical Statistics* 13.1, pp. 158–182.
- Jia, Yangqing, Joshua T Abbott, Joseph L Austerweil, Thomas Griffiths, and Trevor Darrell (2013). “Visual concept learning: Combining machine vision and Bayesian generalization on concept hierarchies”. In: *Advances in Neural Information Processing Systems*, pp. 1842–1850.
- Jiang, Yiding et al. (2020). *NeurIPS 2020 Competition: Predicting generalization in deep learning*.
- Johnson, Carl Nils and Henry M Wellman (1980). “Children’s developing understanding of mental verbs: Remember, know, and guess”. In: *Child Development*, pp. 1095–1102.
- Johnson, Justin, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick (2017). “CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning”. In: pp. 2901–2910.

- Jordan, Michael I, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul (1999). “An introduction to variational methods for graphical models”. In: *Machine Learning* 37.2, pp. 183–233.
- Kapoor, Sayash and Arvind Narayanan (2022). “Leakage and the reproducibility crisis in ml-based science”. In: *arXiv preprint arXiv:2207.07048*.
- Karpathy, Andrej, Justin Johnson, and Li Fei-Fei (2015). “Visualizing and understanding recurrent networks”. In: *Proceedings of the International Conference on Learning Representations, Workshop Track*.
- Kemp, Charles, Amy Perfors, and Joshua B Tenenbaum (2007). “Learning overhypotheses with hierarchical Bayesian models”. In: *Developmental Science* 10.3, pp. 307–321.
- Kingma, Diederik P and Jimmy Ba (2015). “Adam: A method for stochastic optimization”. In: *Proceedings of the International Conference on Learning Representations*.
- Kingma, Diederik P and Jimmy Ba (2014). “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980*.
- Kirkpatrick, James et al. (2017). “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the National Academy of Sciences* 114.13, pp. 3521–3526.
- Koch, Gregory (2015). *Siamese neural networks for one-shot image recognition*.
- Koh, Pang Wei and Percy Liang (2017). “Understanding black-box predictions via influence functions”. In: *Proceedings of the International Conference on Machine Learning*, pp. 1885–1894.
- Labatie, Antoine (2019). “Characterizing well-behaved vs. pathological deep neural networks”. In: *Proceedings of the International Conference on Machine Learning*.
- Lake, Brenden and Marco Baroni (2018a). “Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks”. In: *Proceedings of the International Conference on Machine Learning*. Pmlr, pp. 2873–2882.
- Lake, Brenden M and Marco Baroni (2018b). “Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks”. In: *Proceedings of the International Conference on Learning Representations, Workshop Track*.
- Lake, Brenden M, Ruslan Salakhutdinov, Jason Gross, and Joshua B Tenenbaum (2011). “One-shot learning of simple visual concepts”. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Boston, Massachusetts.
- Lake, Brenden M, Ruslan Salakhutdinov, and Joshua B Tenenbaum (2015). “Human-level concept learning through probabilistic program induction”. In: *Science* 350.6266, pp. 1332–1338.
- Lake, Brenden M, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman (2018). “Building machines that learn and think like people”. In: *Behavioral and Brain Sciences* 40.
- Lalchand, Vidhi and Carl Edward Rasmussen (2020). “Approximate inference for fully Bayesian Gaussian process regression”. In: *Proceedings of the Symposium on Advances in Approximate Bayesian Inference*. Proceedings of Machine Learning Research. Pmlr.
- Landau, Barbara, Linda B Smith, and Susan S Jones (1988). “The importance of shape in early lexical learning”. In: *Cognitive development* 3.3, pp. 299–321.

- Langlois, Thomas A, Charles H Zhao, Erin Grant, Ishita Dasgupta, Thomas L Griffiths, and Nori Jacoby (2021). “Passive attention in artificial neural networks predicts human visual selectivity”. In: *Advances in Neural Information Processing Systems*.
- Laplace, Pierre Simon (1986). “Memoir on the probability of the causes of events”. In: *Statistical Science* 1.3, pp. 364–378.
- Lawrence, Neil D and John C Platt (2004). “Learning to learn with the informative vector machine”. In: *Proceedings of the International Conference on Machine Learning*, p. 65.
- Le Lan, Charline and Laurent Dinh (2021). “Perfect density models cannot guarantee anomaly detection”. In: *Entropy* 23.12, p. 1690.
- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep learning”. In: *Nature* 521.7553, pp. 436–444.
- LeCun, Yann, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller (2012). “Efficient backprop”. In: *Neural Networks: Tricks of the Trade*. Ed. by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. Springer Berlin Heidelberg, pp. 9–48.
- Lee, Jaehoon, Yasaman Bahri, Roman Novak, Samuel Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein (2017). “Deep neural networks as Gaussian processes”. In: *Proceedings of the International Conference on Learning Representations*.
- Lee, Yoonho and Seungjin Choi (2018). “Gradient-based meta-learning with learned layer-wise metric and subspace”. In: *Proceedings of the International Conference on Machine Learning*.
- Li, Ke and Jitendra Malik (2017a). “Learning to Optimize”. In: *Proceedings of the International Conference on Learning Representations*. Toulon, France.
- (2017b). “Learning to Optimize Neural Nets”. In: *Proceedings of the International Conference on Machine Learning*. Sydney, Australia.
- Li, Yuanzhi and Yingyu Liang (2018). “Learning overparameterized neural networks via stochastic gradient descent on structured data”. In: *Advances in Neural Information Processing Systems*.
- Lin, Dahua (2013). “Online learning of nonparametric mixture models via sequential variational approximation”. In: *Advances in Neural Information Processing Systems*, pp. 395–403.
- Lin, Henry W. and Max Tegmark (2017). “Why does deep and cheap learning work so well?” In: *Journal of Statistical Physics* 168, pp. 1223–1247.
- Lipton, Zachary C (2016). “The mythos of model interpretability”. In: arXiv: 1606.03490.
- Liu, Sulin, Xingyuan Sun, Peter J Ramadge, and Ryan P. Adams (2020). “Task-agnostic amortized inference of Gaussian process hyperparameters”. In: *Advances in Neural Information Processing Systems*.
- Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang (2015). “Deep Learning Face Attributes in the Wild”. In: *Proceedings of the International Conference on Computer Vision*.
- Maas, Andrew, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts (2011). “Learning word vectors for sentiment analysis”. In: *Proceedings of*

- the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150.
- MacKay, David (1992a). “A practical Bayesian framework for backpropagation networks”. In: *Neural Computation* 4.3, pp. 448–472.
- MacKay, David JC (1992b). “The evidence framework applied to classification networks”. In: *Neural Computation* 4.5, pp. 720–736.
- Mansinghka, Vikash K, Tejas D Kulkarni, Yura N Perov, and Josh Tenenbaum (2013). “Approximate bayesian image interpretation using generative probabilistic graphics programs”. In: *Advances in Neural Information Processing Systems*, pp. 1520–1528.
- Marcus, Gary (2018). “Deep learning: A critical appraisal”. In: *arXiv preprint arXiv:1801.00631*.
- Markman, Ellen M (1989). *Categorization and naming in children: Problems of induction*. MIT Press.
- Martens, James (2014). “New insights and perspectives on the natural gradient method”. In: *arXiv preprint arXiv:1412.1193*.
- (2016). *Second-order optimization for neural networks*.
- Martens, James and Roger Grosse (2015). “Optimizing neural networks with Kronecker-factored approximate curvature”. In: *Proceedings of the International Conference on Machine Learning*, pp. 2408–2417.
- Matérn, Bertil (1960). *Spatial variation*. Vol. 36. Springer Science & Business Media.
- Matthews, Alexander G de G, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani (2018). “Gaussian process behaviour in wide deep neural networks”. In: *Proceedings of the International Conference on Learning Representations*.
- McClelland, James L (2009). “The place of modeling in cognitive science”. In: *Topics in Cognitive Science* 1.1, pp. 11–38.
- Mehta, Harsh, Ashok Cutkosky, and Behnam Neyshabur (2020). “Extreme memorization via scale of initialization”. In: *arXiv preprint arXiv:2008.13363*.
- Meltzoff, Andrew N, Alison Gopnik, and Betty M Repacholi (1999). “Toddlers’ understanding of intentions, desires and emotions: Explorations of the dark ages”. In: *Developing theories of intention: Social understanding and self control*. Ed. by PD Zelazo, JW Astington, and DR Olson. Erlbaum, pp. 17–41.
- Metropolis, Nicholas and Stanislaw Ulam (1949). “The Monte Carlo method”. In: *Journal of the American statistical association* 44.247, pp. 335–341.
- Miller, George A. (1995). “Wordnet: a lexical database for english”. In: *Commun. ACM* 38.11, pp. 39–41. ISSN: 0001-0782.
- Milligan, Karen, Janet Wilde Astington, and Lisa Ain Dack (2007). “Language and theory of mind: meta-analysis of the relation between language ability and false-belief understanding”. In: *Child develop.* 78.2, pp. 622–646.
- Mishra, Nikhil, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel (2018). “A Simple Neural Attentive Meta-Learner”. In: *Proceedings of the International Conference on Learning Representations*.
- Mitchell, Margaret, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru (2019). “Model

- cards for model reporting”. In: *Proceedings of the conference on fairness, accountability, and transparency*, pp. 220–229.
- Mitchell, Tom M (1980). *The need for biases in learning generalizations*. Tech. rep.
- Müller, Peter and David Rios Insua (1998). “Issues in Bayesian analysis of neural network models”. In: *Neural Computation* 10.3, pp. 749–770.
- Murray, Iain and Ryan P Adams (2010). “Slice sampling covariance hyperparameters of latent Gaussian models”. In: *Advances in Neural Information Processing Systems*.
- Nagarajan, Vaishnavh and J Zico Kolter (2019). “Uniform convergence may be unable to explain generalization in deep learning”. In: *Advances in Neural Information Processing Systems* 32.
- Naik, Devang K and RJ Mammone (1992). “Meta-neural networks that learn by learning”. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 437–442.
- Nair, Vinod and Geoffrey E Hinton (2010). “Rectified linear units improve restricted Boltzmann machines”. In: *Proceedings of the International Conference on Machine Learning*.
- Nakkiran, Preetum (2021). “Towards an empirical theory of deep learning”. PhD thesis. Harvard University.
- Neal, Radford M (1996). “Priors for infinite networks”. In: *Bayesian Learning for Neural Networks*. Springer, pp. 29–53.
- (2000). “Markov chain sampling methods for Dirichlet process mixture models”. In: *Journal of computational and graphical statistics* 9.2, pp. 249–265.
- (2012). *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media.
- Neal, Radford M and Geoffrey E Hinton (1998). “A view of the EM algorithm that justifies incremental, sparse, and other variants”. In: *Learning in graphical models*. Springer, pp. 355–368.
- Nelson, Katherine (2007). *Young minds in social worlds: Experience, meaning, and memory*. Harvard University Press.
- Newell, Allen, John Calman Shaw, and Herbert A Simon (1958). “Elements of a theory of human problem solving.” In: *Psychological Review* 65.3, p. 151.
- Neyshabur, Behnam, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro (2017). “Exploring generalization in deep learning”. In: *Advances in Neural Information Processing Systems*.
- Neyshabur, Behnam, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro (2019). “Towards understanding the role of over-parametrization in generalization of neural networks”. In: *Proceedings of the International Conference on Learning Representations*.
- Nguyen, Cuong V, Yingzhen Li, Thang D Bui, and Richard E Turner (2017). “Variational continual learning”. In: *arXiv preprint arXiv:1710.10628*.
- Nosofsky, Robert M, Steven E Clark, and Hyun Jung Shin (1989). “Rules and exemplars in categorization, identification, and recognition”. In: *Journal of Experimental Psychology: Learning, memory, and cognition* 15.2, p. 282.

- Novak, Roman, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-dickstein (2019). “Bayesian deep convolutional networks with many channels are Gaussian processes”. In: *Proceedings of the International Conference on Learning Representations*.
- Novak, Roman, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A. Alemi, Jascha Sohl-Dickstein, and Samuel S. Schoenholz (2020). “Neural tangents: Fast and easy infinite neural networks in Python”. In: *Proceedings of the International Conference on Learning Representations*.
- O’Laughlin, Claire and Paul Thagard (2000). “Autism and coherence: A computational model”. In: *Mind & Language* 15.4, pp. 375–392.
- Pascanu, Razvan and Yoshua Bengio (2014). “Revisiting natural gradient for deep networks”. In: *Proceedings of the International Conference on Learning Representations*.
- Perez, Luis and Jason Wang (2017). “The effectiveness of data augmentation in image classification using deep learning”. In: *arXiv preprint arXiv:1712.04621*.
- Perner, Josef, Susan R Leekam, and Heinz Wimmer (1987). “Three-year-olds’ difficulty with false belief: The case for a conceptual deficit”. In: *British Journal of Developmental Psychology* 5.2, pp. 125–137.
- Peterson, Joshua C, Joshua T Abbott, and Thomas L Griffiths (2018). “Evaluating (and improving) the correspondence between deep neural networks and human representations”. In: *Cognitive Science* 42.8, pp. 2648–2669.
- Peterson, Joshua C and Thomas L Griffiths (2017). “Evidence for the size principle in semantic and perceptual domains”. In: *arXiv preprint arXiv:1705.03260*.
- Peterson, Joshua C, Paul Soulos, Aida Nematzadeh, and Thomas L Griffiths (2018). “Learning hierarchical visual representations in deep neural networks using hierarchical linguistic labels”. In: *arXiv preprint arXiv:1805.07647*.
- Plato (1976). *Meno*. Hackett Publishing.
- Poggio, Tomaso A., Kenji Kawaguchi, Qianli Liao, Brando Miranda, Lorenzo Rosasco, Xavier Boix, Jack Hidary, and Hrushikesh N. Mhaskar (2018). “Theory of deep learning III: Explaining the non-overfitting puzzle”. In: *arXiv: 1801.00173*.
- Poggio, Tomaso A., Hrushikesh Narhar Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao (2017). “Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review”. In: *International Journal of Automation and Computing* 14, pp. 503–519.
- Pothos, Emmanuel M (2005). “The rules versus similarity distinction”. In: *Behavioral and Brain Sciences* 28.1, p. 1.
- Premack, David and Guy Woodruff (1978). “Does the chimpanzee have a theory of mind?” In: *Behavioral and Brain Sci.* 1.4, pp. 515–526.
- Quine, Willard Van Orman (1960). “Word and object, 1960”. In: *Le mot et la chose*, pp. 1977–2000.
- Raghu, Maithra, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein (2017). “On the expressive power of deep neural networks”. In: *international conference on machine learning*. Pmlr, pp. 2847–2854.

- Raghunathan, Aditi, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang (2020). “Understanding and mitigating the tradeoff between robustness and accuracy”. In: *arXiv preprint arXiv:2002.10716*.
- Rahaman, Nasim, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville (2019). “On the spectral bias of neural networks”. In: *Proceedings of the International Conference on Machine Learning*.
- Raina, Rajat, Andrew Y Ng, and Daphne Koller (2006). “Constructing informative priors using transfer learning”. In: *Proceedings of the International Conference on Machine Learning*, pp. 713–720.
- Rasmussen, Carl E. and Christopher K.I. Williams (Jan. 2006). *Gaussian processes for machine learning*. Adaptive Computation and Machine Learning. MIT Press.
- Rasmussen, Carl Edward (2000). “The infinite Gaussian mixture model”. In: *Advances in neural information processing systems*, pp. 554–560.
- (2003). “Gaussian processes in machine learning”. In: *Summer school on machine learning*. Springer, pp. 63–71.
- Ravi, Sachin and Hugo Larochelle (2017). “Optimization as a model for few-shot learning”. In: *Proceedings of the International Conference on Learning Representations*.
- Raykov, Yordan P, Alexis Boukouvalas, and Max A Little (2016). “Simple approximate MAP inference for Dirichlet processes mixtures”. In: *Electronic Journal of Statistics* 10.2, pp. 3548–3578.
- Razin, Noam and Nadav Cohen (2020). “Implicit regularization in deep learning may not be explainable by norms”. In: *Advances in neural information processing systems* 33, pp. 21174–21187.
- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). ““Why should I trust you?”: Explaining the predictions of any classifier”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144.
- Rips, Lance J (1989). “Similarity, typicality, and categorization”. In:
- Ritter, Hippolyt, Aleksandar Botev, and David Barber (2018). “Online structured Laplace approximations for overcoming catastrophic forgetting”. In: *Advances in Neural Information Processing Systems*, pp. 3738–3748.
- Ritter, Samuel, David G T Barrett, Adam Santoro, and Matt M Botvinick (2017). “Cognitive psychology for deep neural networks: A shape bias case study”. In: *Proceedings of the International Conference on Machine Learning*, pp. 2940–2949.
- Robbins, Herbert and Sutton Monro (1951). “A stochastic approximation method”. In: *The annals of mathematical statistics* 22.3, pp. 400–407.
- Roberts, Daniel A., Sho Yaida, and Boris Hanin (2022). *The principles of deep learning theory*. <https://deeplearningtheory.com>. Cambridge University Press. arXiv: 2106.10165.
- Rosch, Eleanor, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem (1976). “Basic objects in natural categories”. In: *Cognitive Psychology* 8.3, pp. 382–439.

- Rosenstein, Michael T, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich (2005). "To transfer or not to transfer". In: *Proceedings of the NeurIPS Workshop on Transfer Learning*. Vol. 898, pp. 1–4.
- Rosnay, Marc and Claire Hughes (2006). "Conversation and theory of mind: Do children talk their way to socio-cognitive understanding?" In: *British Journal of Developmental Psychology* 24.1, pp. 7–37.
- Rothman, Adam J, Elizaveta Levina, and Ji Zhu (2010). "Sparse multivariate regression with covariance estimation". In: *Journal of Computational and Graphical Statistics* 19.4, pp. 947–962.
- Rumelhart, David E, James L McClelland, PDP Research Group, et al., eds. (1986). *Parallel distributed processing*. MIT Press.
- Rumelhart, DE and JL McClelland (1986). "On learning the past tenses of English verbs". In: *Parallel distributed processing: explorations in the microstructure of cognition, vol. 2: psychological and biological models*, pp. 216–271.
- Russakovsky, Olga et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252.
- Russell, Bertrand (1948). *Human knowledge: Its scope and limits*. New York, USA: Simon and Schuster.
- Sagawa, Shiori, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang (2020). "Distributionally robust neural networks". In: *Proceedings of the International Conference on Learning Representations*.
- Sagawa, Shiori, Aditi Raghunathan, Pang Wei Koh, and Percy Liang (2020). "An investigation of why overparameterization exacerbates spurious correlations". In: *Proceedings of the International Conference on Machine Learning*. Pmlr, pp. 8346–8356.
- Salakhutdinov, Ruslan, Joshua Tenenbaum, and Antonio Torralba (2012). "One-shot learning with a hierarchical nonparametric bayesian model". In: *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pp. 195–206.
- Salakhutdinov, Ruslan, Joshua B Tenenbaum, and Antonio Torralba (2013). "Learning with hierarchical-deep models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8, pp. 1958–1971.
- Samek, Wojciech, Thomas Wiegand, and Klaus-Robert Müller (2017). "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models". In: arXiv: 1708.08296.
- Santoro, Adam, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap (2016). "Meta-learning with memory-augmented neural networks". In: *Proceedings of the International Conference on Machine Learning*, pp. 1842–1850.
- Santos, Reginaldo J. (1996). "Equivalence of regularization and truncated iteration for general ill-posed problems". In: *Linear Algebra and its Applications* 236.15, pp. 25–33.
- Schmidhuber, Jürgen (1987). *Evolutionary principles in self-referential learning*.
- (1992). "Learning to control fast-weight memories: An alternative to dynamic recurrent networks". In: *Neural Computation* 4.1, pp. 131–139.

- Schmidt, Lauren A (2009). "Meaning and compositionality as statistical induction of categories and constraints". PhD thesis. Massachusetts Institute of Technology.
- Schoenholz, Samuel S., Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein (2017). "Deep information propagation". In: *Proceedings of the International Conference on Learning Representations*.
- Sculley, David (2010). "Web-scale k-means clustering". In: *Proceedings of the 19th international conference on World wide web*. Acm, pp. 1177–1178.
- Sejnowski, Terrence J. (2020). "The unreasonable effectiveness of deep learning in artificial intelligence". In: *Proceedings of the National Academy of Sciences* 117, pp. 30033–30038.
- Shahriari, Bobak, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas (2016). "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104, pp. 148–175.
- Shepard, Roger N (1987). "Toward a universal law of generalization for psychological science". In: *Science* 237.4820, pp. 1317–1323.
- Shepard, Roger N and Jih-Jie Chang (1963). "Stimulus generalization in the learning of classifications". In: *Journal of Experimental Psychology* 65.1, p. 94.
- Simpson, Fergus, Ian Davies, Vidhi Lalchand, Alessandro Vullo, Nicolas Durrande, and Carl Edward Rasmussen (2021). "Kernel identification through transformers". In: *Advances in Neural Information Processing Systems*.
- Sjöberg, Jonas and Lennart Ljung (1995). "Overtraining, regularization and searching for a minimum, with application to neural networks". In: *International Journal of Control* 62.6, pp. 1391–1407.
- Slaughter, Virginia and Alison Gopnik (1996). "Conceptual coherence in the child's theory of mind: Training children to understand belief". In: *Child development* 67.6, pp. 2967–2988.
- Smith, Edward E and Steven A Sloman (1994). "Similarity-versus rule-based categorization". In: *Memory & Cognition* 22.4, pp. 377–386.
- Smith, Samuel L, Benoit Dherin, David GT Barrett, and Soham De (2021). "On the origin of implicit regularization in stochastic gradient descent". In: *Proceedings of the International Conference on Learning Representations*.
- Snell, Jake, Kevin Swersky, and Richard S Zemel (2017). "Prototypical Networks for Few-shot Learning". In: *Advances in Neural Information Processing Systems*.
- Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams (2012). "Practical Bayesian optimization of machine learning algorithms". In: *Advances in Neural Information Processing Systems*.
- Sorenson, Harold W and Daniel L Alspach (1971). "Recursive Bayesian estimation using Gaussian sums". In: *Automatica* 7.4, pp. 465–479.
- Soudry, Daniel, Elad Hoffer, and Nathan Srebro (2018). "The implicit bias of gradient descent on separable data". In: *Proceedings of the International Conference on Learning Representations*.

- Srivastava, Nitish and Ruslan R Salakhutdinov (2013). “Discriminative transfer learning with tree-based priors”. In: *Advances in Neural Information Processing Systems*, pp. 2094–2102.
- Sukhbaatar, Sainbayar, Jason Weston, Rob Fergus, et al. (2015). “End-to-end memory networks”. In: *Advances in Neural Information Processing Systems*, pp. 2440–2448.
- Sun, Ron (2008). *The Cambridge handbook of computational psychology*. Cambridge University Press.
- Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus (2014). “Intriguing properties of neural networks”. In: *Proceedings of the International Conference on Learning Representations*.
- Tank, Alex, Nicholas Foti, and Emily Fox (2015). “Streaming variational inference for Bayesian nonparametric mixture models”. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 968–976.
- Tenenbaum, Joshua B (1999). *A Bayesian framework for concept learning*.
- Tenenbaum, Joshua B and Thomas L Griffiths (2001). “Generalization, similarity, and Bayesian inference”. In: *Behavioral and Brain Sciences* 24.04, pp. 629–640.
- Thrun, Sebastian and Joseph O’Sullivan (1996). “Discovering structure in multiple learning tasks: The TC algorithm”. In: *Proceedings of the International Conference on Machine Learning*. Vol. 96, pp. 489–497.
- Thrun, Sebastian and Lorien Pratt (1998). *Learning to learn*. Kluwer Academic Publishers.
- Tiño, Peter, Michal Cernanský, and Lubica Benusková (2004). “Markovian architectural bias of recurrent neural networks”. In: *IEEE Transactions on Neural Networks* 15, pp. 6–15.
- Triantafillou, Eleni, Richard Zemel, and Raquel Urtasun (2017). “Few-Shot Learning Through an Information Retrieval Lens”. In: *Advances in Neural Information Processing Systems*.
- Triona, Lara M, Amy M Masnick, and Bradley J Morris (2002). “What does it take to pass the false belief task? An ACT-R model”. In: *Proceedings of Cog. Sci.* P. 1045.
- Turing, Alan M (1950). “Computing machinery and intelligence”. In: *Mind*.
- Van Loan, Charles F (2000). “The ubiquitous Kronecker product”. In: *Journal of computational and applied mathematics* 123.1, pp. 85–100.
- Van Overwalle, Frank (2010). “Infants’ teleological and belief inference: A recurrent connectionist approach to their minimal representational and computational requirements”. In: *NeuroImage* 52.3, pp. 1095–1108.
- Van Steenkiste, Sjoerd, Michael Chang, Klaus Greff, and Jürgen Schmidhuber (2018). “Relational neural expectation maximization: unsupervised discovery of objects and their interactions”. In: *Proceedings of the International Conference on Learning Representations*.
- Vinyals, Oriol, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. (2016). “Matching networks for one shot learning”. In: *Advances in Neural Information Processing Systems*, pp. 3630–3638.

- Wan, Jing et al. (2012). “Sparse Bayesian multi-task learning for predicting cognitive outcomes from neuroimaging measures in Alzheimer’s disease”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. Ieee, pp. 940–947.
- Wang, G Gary and Songqing Shan (May 2006). “Review of metamodeling techniques in support of engineering design optimization”. In: *Journal of Mechanical Design* 129.4, pp. 370–380.
- Welling, Max and Kenichi Kurihara (2006). “Bayesian K -means as a “maximization-expectation” algorithm”. In: *Proceedings of the 2006 SIAM international conference on data mining*. Siam, pp. 474–478.
- Werbos, Paul J. (1988). “Generalization of backpropagation with application to a recurrent gas market model”. In: *Neural Networks* 1, pp. 339–356.
- Weston, Jason, Antoine Bordes, Sumit Chopra, and Tomas Mikolov (2016). “Towards AI-complete question answering: A set of prerequisite toy tasks”. In: *Iclr*.
- Wichrowska, Olga, Niru Maheswaranathan, Matthew W Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando de Freitas, and Jascha Sohl-Dickstein (2017). “Learned Optimizers that Scale and Generalize”. In: *Proceedings of the International Conference on Machine Learning*. Sydney, Australia.
- Wilson, Andrew G and Ryan P Adams (2013). “Gaussian process kernels for pattern discovery and extrapolation”. In: *Proceedings of the International Conference on Machine Learning*.
- Xian, Yongqin, Christoph H Lampert, Bernt Schiele, and Zeynep Akata (2018). “Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly”. In: *IEEE transactions on pattern analysis and machine intelligence* 41.9, pp. 2251–2265.
- Xiao, Lechao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S. Schoenholz, and Jeffrey Pennington (2018). “Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks”. In: *Proceedings of the International Conference on Machine Learning*.
- Xu, Fei and Joshua B Tenenbaum (2007). “Word learning as Bayesian inference”. In: *Psychological Review* 114.2, pp. 245–272.
- Xu, Keyulu, Mozhi Zhang, Jingling Li, Simon Shaolei Du, Ken-Ichi Kawarabayashi, and Stefanie Jegelka (2021). “How neural networks extrapolate: From feedforward to graph neural networks”. In: *Proceedings of the International Conference on Learning Representations*.
- Xue, Tianju, Alex Beatson, Sigrid Adriaenssens, and Ryan P. Adams (2020). “Amortized finite element analysis for fast PDE-constrained optimization”. In: *Proceedings of the International Conference on Machine Learning*. Vol. 119, pp. 10638–10647.
- Xue, Ya, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram (2007). “Multi-task learning for classification with dirichlet process priors”. In: *Journal of Machine Learning Research* 8. Jan, pp. 35–63.
- Yang, Greg (2019). “Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation”. In: arXiv: 1902.04760.

- Ye, Haotian, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang (2021). “Towards a theoretical framework of out-of-distribution generalization”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan.
- Youngblade, Lise M and Judy Dunn (1995). “Individual differences in young children’s pretend play with mother and sibling: Links to relationships and understanding of other people’s feelings and beliefs”. In: *Child Development* 66.5, pp. 1472–1492.
- Yu, Kai, Volker Tresp, and Anton Schwaighofer (2005). “Learning Gaussian processes from multiple tasks”. In: *Proceedings of the International Conference on Machine Learning*, pp. 1012–1019.
- Zeiler, Matthew D and Rob Fergus (2014). “Visualizing and understanding convolutional networks”. In: *European conference on computer vision*, pp. 818–833.
- Zenke, Friedemann, Ben Poole, and Surya Ganguli (2017). “Continual learning through synaptic intelligence”. In: *arXiv preprint arXiv:1703.04200*.
- Zhang, Yi and Jeff G Schneider (2010). “Learning multiple tasks with a sparse matrix-normal penalty”. In: *Advances in Neural Information Processing Systems*, pp. 2550–2558.
- Zhang, Yongyue, Michael Brady, and Stephen Smith (2001). “Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm”. In: *IEEE transactions on medical imaging* 20.1, pp. 45–57.
- Zhang, Yu and Dit-Yan Yeung (2014). “A regularization approach to learning task relationships in multitask learning”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 8.3, p. 12.
- Zhou, Yilun, Marco Tulio Ribeiro, and Julie Shah (2022). “ExSum: From local explanations to model understanding”. In: *Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Appendix A

Mathematical derivations

A.1 Recasting meta-learning as hierarchical Bayes

Simultaneous Diagonalization

Lemma 1. *Suppose \mathbf{A} and \mathbf{B} are n -by- n real symmetric matrices, and furthermore that \mathbf{B} is positive definite. Then there exists a nonsingular \mathbf{X} such that both $\mathbf{X}^\top \mathbf{A} \mathbf{X}$ and $\mathbf{X}^\top \mathbf{B} \mathbf{X}$ are diagonal.*

Proof. Let $\mathbf{Q}^\top \mathbf{B} \mathbf{Q} = \text{diag}(b_1, \dots, b_n)$ be the spectral decomposition of \mathbf{B} , where $b_i > 0$ for all i since \mathbf{B} is positive definite. Set $\mathbf{P} = \mathbf{Q} \cdot \text{diag}(1/\sqrt{b_1}, \dots, 1/\sqrt{b_n})$, and let $(\mathbf{PZ})^\top \mathbf{A} \mathbf{PZ} = \text{diag}(a_1, \dots, a_n)$ be the Schur decomposition of $\mathbf{P}^\top \mathbf{A} \mathbf{P}$.

Set $\mathbf{X} = \mathbf{PZ}$. Then $\mathbf{X}^\top \mathbf{A} \mathbf{X}$ is diagonal as a rewriting of the Schur decomposition. Furthermore, noting that $\mathbf{Z}^\top \mathbf{Z} = \mathbb{I}$ and $\mathbf{P}^\top \mathbf{B} \mathbf{P} = \mathbb{I}$, it can be easily verified that $\mathbf{X}^\top \mathbf{B} \mathbf{X}$ is diagonal:

$$\mathbf{X}^\top \mathbf{B} \mathbf{X} = (\mathbf{PZ})^\top \mathbf{B} \mathbf{PZ} = \mathbf{Z}^\top \mathbf{P}^\top \mathbf{B} \mathbf{PZ} = \mathbf{Z}^\top \mathbf{Z} = \mathbb{I}.$$

□

The nonsingular matrix \mathbf{X} in Lemma 1 can be determined by computing the Cholesky factorization $\mathbf{B} = \mathbf{G} \mathbf{G}^\top$, then computing $\mathbf{C} = \mathbf{G}^{-1} \mathbf{A} \mathbf{G}^{-\top}$, then computing the Schur decomposition $\mathbf{Q}^\top \mathbf{C} \mathbf{Q} = \text{diag}(a_1, \dots, a_n)$, and finally setting $\mathbf{X} = \mathbf{G}^{-\top} \mathbf{Q}$ Golub and Van Loan (Algorithm 8.7.1 in 1983). However, it should be noted that the matrix \mathbf{X} does not uniquely ensure that both $\mathbf{X}^\top \mathbf{A} \mathbf{X}$ and $\mathbf{X}^\top \mathbf{B} \mathbf{X}$ are diagonal. In particular, replacing \mathbf{B} with a suitable nonnegative definite convex combination of \mathbf{A} and \mathbf{B} provides another matrix \mathbf{X} such that both $\mathbf{X}^\top \mathbf{A} \mathbf{X}$ and $\mathbf{X}^\top \mathbf{B} \mathbf{X}$ are diagonal Golub and Van Loan (Theorem 8.7.1 in 1983).

Early Stopping as Regularization

Given the linear system $\hat{\mathbf{y}} = \mathbf{X} \phi$, we may consider the regularized linear least squares problem

$$\min \left(\|\mathbf{X} \phi - \mathbf{y}\|_{\mathbf{P}}^2 + \|\phi - \mathbf{a}\|_{\mathbf{Q}}^2 \right) \quad (\text{A.1})$$

where \mathbf{a} is a vector in \mathbb{R}^n , and \mathbf{P} and \mathbf{Q} are symmetric positive definite matrices with norms defined by $\|\mathbf{z}\|_{\mathbf{P}} = \mathbf{z}^\top \mathbf{P}^{-1} \mathbf{z}$ and similarly for \mathbf{Q} . Note that if \mathbf{P} and \mathbf{Q} are variance-covariance matrices, then Problem (A.1) is a standard statistical regularization problem. Additionally, consider a convergent iterative method of the form

$$\phi_{(k)} = \phi_{(k-1)} + \mathbf{M} \mathbf{X}^\top \mathbf{P}^{-1} (\mathbf{y} - \mathbf{X} \phi_{(k-1)}) \quad (\text{A.2})$$

where \mathbf{M} is a nonsingular matrix.

Santos (1996) shows that every solution to Equation (A.1) is an iterate given by Equation (A.2) and vice versa. We state this equivalence result in the following two theorems and refer the reader to Santos (1996) for the complete proof.

Theorem 2 (3.1 in Santos (1996)). *The solution to the regularized problem in Equation (A.1) has an equivalent truncated iterative solution of the form Equation (A.2). In particular, for a positive iteration index k_0 , there exists a matrix \mathbf{M} such that $\phi_{(k_0)}$ given by Equation (A.2) solves Equation (A.1).*

Proof. Since \mathbf{Q} and $\mathbf{X}^T \mathbf{P}^{-1} \mathbf{X}$ are symmetric and \mathbf{Q}^{-1} is positive definite, we can simultaneously diagonalize them. Thus, there exists a nonsingular matrix \mathbf{A} such that $\mathbf{A}^T \mathbf{Q}^{-1} \mathbf{A} = \text{diag}(1/q_1, \dots, 1/q_n)$ and $\mathbf{A}^T \mathbf{X}^T \mathbf{P}^{-1} \mathbf{X} \mathbf{A} = \text{diag}(p_1, \dots, p_n)$ with $q_i > 0$ and $p_i \geq 0$ for $i = 1, \dots, n$. Consequently,

$$\mathbf{A}^{-1} \mathbf{Q} \mathbf{X}^T \mathbf{P}^{-1} \mathbf{X} \mathbf{A} = \mathbf{A}^{-1} \mathbf{Q} \mathbf{A}^{-T} \mathbf{A}^T \mathbf{X}^T \mathbf{P}^{-1} \mathbf{X} \mathbf{A} = \text{diag}(p_1 q_1, \dots, p_n q_n) .$$

Given a truncation index k_0 , let

$$\mathbf{M} = \mathbf{A} \text{diag}(\lambda_1, \dots, \lambda_n) \mathbf{A}^T ,$$

where $\lambda_i = (1/p_i)[1 - (1 + p_i q_i)^{-1/k_0}]$ if $p_i \neq 0$ and 1 otherwise.

□

Theorem 3 (3.2 in Santos (1996)). *Every truncated-iterative solution of the form Equation (A.2), where \mathbf{M} is a symmetric positive definite matrix, is the solution of a regularized problem of the form Equation (A.1); i.e., for every k and matrices \mathbf{M} and \mathbf{P} , there exists a matrix \mathbf{Q} such that $\phi_{(k)}$ in Equation (A.2) solves Equation (A.1).*

By taking $\mathbf{P} = \mathbb{I}$ and $\mathbf{M} = \text{diag}(\eta/2, \dots, \eta/2)$ in Theorem (3), we are assured that for each iterate k of the standard steepest descent update rule

$$\begin{aligned} \phi_{(k)} &= \phi_{(k-1)} - \eta \mathbf{X}^T (\mathbf{X} \phi_{(k-1)} - \mathbf{y}) \\ &= \phi_{(k-1)} - \eta \nabla_{\phi} [\|\mathbf{X} \phi - \mathbf{y}\|_2^2] (\phi_{(k-1)}) \end{aligned}$$

and for each \mathbf{a} that there exists a matrix \mathbf{Q} such that $\phi_{(k)}$ solves

$$\min \left((\mathbf{X} \phi - \mathbf{y})^T (\mathbf{X} \phi - \mathbf{y}) + (\phi - \mathbf{a})^T \mathbf{Q}^{-1} (\phi - \mathbf{a}) \right) . \quad (\text{A.3})$$

Thus, gradient descent with early stopping at iteration k_0 is equivalent to regularization of the corresponding linear least squares problem.

Regularization as MAP Estimation

We may identify Equation (A.3) as a negative log posterior minimization problem. In particular, assuming a conditional Gaussian likelihood over \mathbf{y} with identity covariance

and a Gaussian prior over ϕ with mean \mathbf{a} and covariance \mathbf{Q} and subsequently dropping all terms that do not depend on ϕ gives

$$\begin{aligned} \operatorname{argmin}_{\phi} -\log p(\phi | \mathbf{X}, \mathbf{y}) &= \operatorname{argmin}_{\phi} (-\log p(\mathbf{y} | \mathbf{X}, \phi) - \log p(\phi)) \\ &= \operatorname{argmin}_{\phi} (-\log \mathcal{N}(\mathbf{y}; \mathbf{X}\phi, \mathbb{I}) - \log \mathcal{N}(\phi; \mathbf{a}, \mathbf{Q})) \\ &= \operatorname{argmin}_{\phi} \left(\frac{1}{2}(\mathbf{y} - \mathbf{X}\phi)^{\top} \mathbb{I}^{-1}(\mathbf{y} - \mathbf{X}\phi) + \frac{1}{2}(\phi - \mathbf{a})^{\top} \mathbf{Q}^{-1}(\phi - \mathbf{a}) \right) \end{aligned}$$

which admits the same minimizer as Equation (A.3). Thus, the solution to the regularized problem is also the *maximum a posteriori* estimate of the given probabilistic formulation.

The Form of the Prior in the Linear Case

It is instructive to investigate the form of the prior imposed by early stopping in the linear case. If \mathbf{O} is an orthogonal $D \times D$ matrix such that $\mathbf{O}^{\top} \mathbf{X}^{\top} \mathbf{X} \mathbf{O} = \operatorname{diag}((\lambda_1, \dots, \lambda_D))$, then, using the proof of Theorem (3) in (Santos 1996), $\phi_{(k)}$ in Equation (A.2) is the solution of the problem

$$\operatorname{argmin}_{\phi} \left(\|\mathbf{y} - \mathbf{X}\phi\|^2 + \gamma \|\mathbf{D}^{-1/2} \mathbf{O}^{\top} \phi\|^2 \right), \quad (\text{A.4})$$

where $\mathbf{D} = \operatorname{diag}((\mu_i))$ for

$$\mu_i = \begin{cases} \frac{1}{\lambda_i} \left[\frac{1}{(1-\eta\lambda_i)^k} - 1 \right] & \text{if } \lambda_i \neq 0 \\ 1 & \text{otherwise.} \end{cases}$$

Rewritten as the problem

$$\operatorname{argmin}_{\phi} -\log p(\phi | \mathbf{X}, \mathbf{y}),$$

(A.4) corresponds to taking a Gaussian prior with mean $\boldsymbol{\mu} = \mathbf{0}$ and covariance $\boldsymbol{\Sigma} = \gamma^{-1} \mathbf{O} \mathbf{D} \mathbf{O}^{\top}$.

Since \mathbf{O} orthogonally diagonalizes $\mathbf{X}^{\top} \mathbf{X}$, \mathbf{O} is the matrix of eigenvectors of $\mathbf{X}^{\top} \mathbf{X}$ and the λ_i are the corresponding eigenvalues. Therefore, the regularization term can be understood as a Mahalanobis distance of which eigenvalues of the covariance, $\mathbf{X}^{\top} \mathbf{X}$, are rescaled to $(\gamma \lambda_i)^{-1} [(1 - \eta \lambda_i)^{-k} - 1]$. In particular, if the step size and all eigenvalues are small (*i.e.*, $\eta \lambda_i < 1$), the new eigenvalues grow exponentially in the number of steps k , proportional to the regularization coefficient γ .

Assuming the design matrix, \mathbf{X} , is centered and full rank, we can understand the above as defining a data-dependent prior covariance whose width in the directions of greatest variation in the data (*i.e.*, the eigenvectors of the unscaled sample covariance matrix $\mathbf{X}^{\top} \mathbf{X}$ with the largest eigenvalues, also known as the principal eigenvectors) grows exponentially in the number of gradient descent steps k . Interpreted as a form of regularization, the prior thus favors dimensions of extreme variation in the data by more lightly penalizing the magnitude of the weights in these principal component directions.

The Form of the Prior in the General Case

A central motivation behind MAML is that the learned weight initialization is one or a few gradient steps away from an optimal parameter setting to solve any given task sampled from the task distribution. Thus, for a general nonlinear model, we can understand the behavior of fast adaptation via a quadratic approximation in the neighborhood of an optimal parameter setting ϕ^* .

Consider the second-order Taylor series expansion of the objective \mathcal{L} about a minimum ϕ^* :

$$\tilde{\mathcal{L}}(\phi) = \mathcal{L}(\phi^*) + \frac{1}{2}(\phi - \phi^*)^\top \mathbf{H}(\phi^*)(\phi - \phi^*) \quad (\text{A.5})$$

where the first order derivative vanishes, \mathbf{H} is the Hessian of \mathcal{L} and we omit the dependence on the data (\mathbf{X}, \mathbf{y}) of \mathcal{L} and \mathbf{H} . * Taking the gradient of Equation (A.5) gives the steepest descent update

$$\phi_{(k)} = \phi_{(k-1)} - \eta \mathbf{H}(\phi^*)(\phi_{(k-1)} - \phi^*) .$$

By writing $\phi_{(k)} = [\mathbf{I} - \eta \mathbf{H}(\phi^*)]\phi_{(k-1)} - \eta \mathbf{H}(\phi^*)\phi^*$ we can identify this as a recursive definition in order to solve for $\phi_{(k)}$ in terms of $\phi_{(0)}$ and ϕ^* :

$$\phi_{(k)} = [\mathbb{I} - \eta \mathbf{H}(\phi^*)]^k \phi_{(0)} - \eta \mathbf{H}(\phi^*)\phi^* \sum_{j=0}^{k-1} [\mathbb{I} - \eta \mathbf{H}(\phi^*)]^j$$

By an argument analogous to the proof of Theorem (3) in Santos (1996), $\phi_{(k)}$ is then a solution to

$$\min_{\phi} \left(\tilde{\mathcal{L}}(\phi) + \|\phi - \phi_{(0)}\|_{\mathbf{Q}}^2 \right) \quad (\text{A.6})$$

for $\mathbf{Q} = \mathbf{O}\mathbf{\Lambda}^{-1}((\mathbb{I} - \eta\mathbf{\Lambda})^{-k} - \mathbb{I})\mathbf{O}^\top$, a rescaling of the eigenvalues of $\mathbf{H}(\phi^*) = \mathbf{O}\mathbf{\Lambda}\mathbf{O}^\top$. In particular, the rescaling occurs so that, for each i ,

$$\lambda_i \leftarrow \frac{(1 - \eta\lambda_i)^{-k} - 1}{\lambda_i} .$$

which can be further rewritten as

$$\phi_{(k)} = \phi_{(0)} - \eta \mathbf{H}(\phi^*)(\phi_{(0)} - \phi^*) - \eta \mathbf{O}\mathbf{\Lambda}^{-1} [\mathbb{I} - \eta\mathbf{\Lambda}]^{-k} \mathbf{O}(\phi_{(0)})$$

Therefore, early stopping after a single or a few steps of fast adaptation under a quadratic approximation to the loss corresponds to a regularization that penalizes directions of high curvature of the loss more lightly.

*The second-order expansion is sometimes also called the Newton approximation because the assumption that $\tilde{\mathcal{L}}(\phi)$ is a valid local approximation of $\mathcal{L}(\phi)$ is the central motivation behind Newton's method.

Alternatively, if we take a probabilistic interpretation of the regularization, so that \mathbf{Q}^{-1} is the covariance of a Gaussian prior over the parameters, then we see that the eigenvalues of the covariance are rescaled such that, for each i ,

$$\lambda_i \leftarrow \frac{\lambda_i}{(1 - \eta \lambda_i)^{-k} - 1} .$$

Under this formulation, early stopping corresponds to MAP inference with a Gaussian prior with exponentially high variance in directions of high curvature of the loss.

Appendix B

Additional experimental results

B.1 Nonparametric priors for non-stationarity

Extended related work

Multi-task learning. Rosenstein et al. (2005) demonstrated that negative transfer can worsen generalization performance, and avoidance of negative transfer has motivated much work on hierarchical Bayes in transfer learning and domain adaptation (*e.g.*, Lawrence and Platt 2004; Yu et al. 2005; Gao et al. 2008; Daumé III 2009; Wan et al. 2012). Closest to our proposed approach is early work on hierarchical Bayesian multi-task learning with neural networks that places a prior only on the output layer (Heskes 1998; Bakker and Heskes 2003; Salakhutdinov et al. 2013; Srivastava and Salakhutdinov 2013). In contrast, we place a nonparametric prior on the full set of neural network weights. Furthermore, none of these approaches were applied to the episodic training setting of meta-learning. Similar to our point estimation procedure, Heskes (1998) and Srivastava and Salakhutdinov (2013) propose training a mixture model over the output layer weights of a neural network using MAP inference. However, these approaches do not scale well to all the layers in a network as performing full passes on the dataset for inference of the full set of weights is computationally intractable in general.

Clustering. Incremental or stochastic clustering was considered in the EM setting in Neal and Hinton (1998). and in the K -means setting in Sculley (2010). Lin (2013) conducted online learning of a nonparametric mixture model using sequential variational inference. A key distinction between our work and these approaches is that we leverage the connection between empirical Bayes in a hierarchical model and gradient-based ML (Grant et al. 2018) to use a MAML-like (Finn, Abbeel, et al. 2017) objective as a log posterior surrogate. This allows our algorithm to make use of a scalable stochastic gradient descent optimizer instead of alternating a maximization step with an inference pass over the full dataset (*c.f.*, Srivastava and Salakhutdinov 2013; Bauer et al. 2017).

Our approach is also distinct from recent work on gradient-based clustering (Greff et al. 2017) since we employ the episodic batching of Vinyals et al. (2016). This can be a challenging setting for a clustering algorithm, as the assignments need to be computed using, for example, $K = 1$ examples per class in the 1-shot setting.

Contrasting the batch and stochastic settings. In the stochastic setting, access to past data is unavailable, and so none of the standard algorithms and heuristics for inference in nonparametric models are applicable Jain and Neal (*e.g.*, 2004) and Hughes et al. (2012). In particular, our proposed algorithm does not refine the cluster assignments of previously observed points by way of multiple expensive passes over the whole dataset.

In contrast, we incrementally infer model parameters and add components during episodic training based on noisy estimates of the gradients of the marginal log-likelihood. Moreover, we avoid the need to preserve task assignments, which is potentially harmful due to stale parameter values, since the task assignments in our framework are meant to be easily reconstructed on-the-fly using the E-STEP with updated parameters $\theta^{(0)}, \dots, \theta^{(L)}, G$.

Maximum a posteriori estimation as iterated conditional modes. Due to the high-dimensionality of the parameter set of neural networks, we consider a mode estimation procedure based on iterated conditional modes (ICM) (Besag 1986; Zhang, Brady, et al. 2001; Welling and Kurihara 2006; Raykov et al. 2016) that can leverage gradient computation instead of the expensive process of Gibbs sampling. iterated conditional modes (ICM) is a greedy strategy that iteratively maximizes the full conditional distribution for each variable (*i.e.*, computes the MAP estimate), instead of sampling from the conditional as is done in Gibbs sampling (Welling and Kurihara 2006). This leads to a fast point-estimation of the DPMM parameters in which we only need to track the means of the cluster priors.

Alternative inference procedures in probabilistic mixtures. A standard approach for estimation in latent variable models, such as probabilistic mixtures, is to represent the distribution using samples produced via some sampling algorithm. The most widely used is the Gibbs sampler (Neal 2000; Gershman and Blei 2012), which draws from the conditional distribution of each latent variable, given the others, until convergence to the posterior distribution over all the latents. However, in the setting of latent variables defined over high-dimensional parameter spaces such as those of neural network models, using a sampling algorithm such as the Gibbs sampler is prohibitively expensive (Neal 2012; Müller and Insua 1998). Instead of sampling, one can fit factorized variational distributions to the exact distribution $p(\phi, z|x) \approx q(\phi)q(z)$ (Ghahramani and Beal 2000; Blei, Jordan, et al. 2006). It should be noted that we do not claim that our method of point estimation in the DPMM is the most accurate method for posterior inference but we leave improved approximate inference extensions to future work.

The main drawback of using point estimates for a nonparametric mixture estimation is the inability to leverage the diffusion of the global prior G_0 when computing the likelihood of a new cluster. Highly concentrated parameter estimates for non-empty clusters should lead to low likelihoods for outlier tasks, whereas the diffused global prior should be better at capturing a wider variety of tasks. Nonetheless, point estimation is a necessary trade-off between computation and accuracy. To allow for a more accurate estimate of the likelihood, we experimented with simulating a normal centered at the global prior mean with a variance hyperparameter that can be annealed over time to account for increased certainty about the prior choice. We can then compare the average cluster responsibility to the threshold. Another interesting extension we experimented with was to compute the gradient for each of the samples and average over the number of samples as to approximate the expectation of the gradient under the global prior. However, we found this to be less stable than simply comparing the cluster responsibilities to the threshold.

Maximum a posteriori estimation in the Dirichlet process mixture model

From (6.4) and using a conditional mode estimate for task-specific parameters ϕ_j ,

$$\log p(\mathbf{z}_j = \ell \mid \mathbf{x}_{j:1:M}, \mathbf{z}_{1:j-1}, \boldsymbol{\theta}^{(\ell)}) \approx \begin{cases} \log n^{(\ell)} + \log p(\mathbf{x}_{j:1:M} \mid \hat{\phi}_j^{(\ell)}) + \\ \log p(\hat{\phi}_j^{(\ell)} \mid \boldsymbol{\theta}^{(\ell)}) & \text{for } \ell \leq L \\ \log \zeta + \log p(\mathbf{x}_{j:1:M} \mid \hat{\phi}_j^{(\ell)}) + \\ \log(\hat{\phi}_j^{(\ell)} \mid \boldsymbol{\theta}^{(0)}) & \text{for } \ell = L + 1. \end{cases} \quad (\text{B.1})$$

Experimental setup**Dataset details****Few-shot regression**

- Polynomial wave (Figure 6.4a):

$$y = \sum_i a_i x^{p_i}$$

and $a \sim \mathcal{U}(-5.0, 5.0)$.

- Sinusoid wave (Figure 6.4b):

$$y = a \sin(x - \phi)$$

where $\phi \sim \mathcal{U}(0, \pi)$ and $a \sim \mathcal{U}(0.1, 5.0)$.

- Sawtooth wave (Figure 6.4c):

$$y = -\frac{2a}{\pi} \arctan(\cot(\frac{x\pi}{\phi}))$$

where $\phi \sim \mathcal{U}(0, \pi)$, $a \sim \mathcal{U}(0.1, 5.0)$.

Hyperparameter choices

MiniImageNet few-shot classification. We use the same data split, neural network architecture, and hyperparameter values as in Finn, Abbeel, et al. (2017) for common components. We use $\tau = 1$ for the softmax temperature and the same initialization as Finn, Abbeel, et al. (2017) for the global prior G_0 . We determine an iteration number for early stopping using the validation set.

Continual few-shot regression. Our architecture is a feedforward neural network with 2 hidden layers with ReLU nonlinearities, each of size 40. We use a meta-batch size of 10 tasks (both for the inner updates and the meta-gradient updates) for 5-shot regression. Our nonparametric algorithm starts with a single cluster ($L_0 = 1$ in Algorithm 6.3). In these experiments, we set the spawning threshold $\epsilon = 0.95T/(L + 1)$, with L the number of non-empty clusters and T the size of the meta-batch. We use the mean-squared error for each task as the inner loop and meta-level objectives.

Continual few-shot *miniImageNet* classification. We use the same data split, neural network architecture, and hyperparameter values as in Finn, Abbeel, et al. (2017) for common components. We use a meta-batch size of 4 tasks, start with a single cluster, and set the spawning threshold to the same formula as in Appendix B.1. We use the multi-class cross entropy error for each task as the inner loop and meta-level objectives. More details on the practical implementation for image datasets of the nonparametric algorithm can be found in Appendix B.1.

Practical and implementational details

Task-aware vs. task-agnostic

Since a cluster is not well-tuned immediately after its creation, we consider a cool-down period after the spawning of each new cluster where we do not consider the creation of new clusters for a fixed number of iterations, and we freeze the updating of existing clusters for a same number of iterations. This allows the newly created cluster to take enough gradient updates in order to move from its global prior initialization, allowing it to sufficiently differentiate from the global prior.

This experimental paradigm also allows us to approximate the *task-aware* algorithms of prior work Kirkpatrick et al. (e.g., 2017), Zenke et al. (2017), Nguyen et al. (2017), and Ritter, Botev, et al. (2018) which require access to an explicit delineation between tasks that acts as a catalyst to grow model size. For the *task-aware* nonparametric mixture results reported in the experiments, we set this cool-down period to be exactly the length of the training phase for the appropriate dataset; therefore, clusters which are not meant to be specialized for the active dataset are not updated. In contrast, the *task-aware* results consider a cool-down period of $1k$ iterations, which is less than 15% of the active period for each dataset. Extensions to this fixed cool-down period could consider the rate of learning in the active cluster in order to detect when the new component has been sufficiently fit to the new task.

Practical extensions to the nonparametric algorithm

The penalty term of $\log n^{(\ell)}$ or $\log \zeta$ is necessary to regularize the likelihood of a potential new cluster in order to limit overspawning. However, in the setting where the likelihood

is approximated by the loss function of a complex neural network, as in the case for most meta-learning applications, there is a large difference in orders of magnitude between the loss value (especially for the cross-entropy function) and the penalty term, even after a single batch of assignments. Furthermore, the classical log observation count $\log n$ term is misaligned with our stochastic setting for two reasons. First, since we do not re-evaluate over the whole dataset for every meta-learning episode, we are thus more concerned with the relative number of task assignments over recent iterations than the total number of assignments over the duration of training. Second, the number of tasks to be assigned can grow too large in the stochastic setting (e.g. $60k$ for *miniImageNet*) which exacerbates the already large difference in orders of magnitudes between the loss function and the penalty term.

Accordingly, we propose two changes; First, we compute the observation based on a moving window of fixed size (5 in the experiments). Second, we apply a coefficient, which can be tuned, to the log observation count in (6.4). This provides more flexibility to our meta-learner as it allows it to apply to any black-box function approximator which might exhibit losses of orders of magnitudes smaller than those expected of classical probabilistic models. While the moving window size and CRP penalty coefficient terms are somewhat interdependent, we propose them as a simple starting point to tune this nonparametric meta-learner beyond what is empirically explored in this paper.

Note that without such changes in the stochastic setting of meta-learning, a non-parametric algorithm would be unable to spawn a new cluster after the first handful of iterations. Even if we were to lower the threshold ϵ , multiple almost identical clusters would be spawned in the first few iterations before it would be impossible to spawn anymore. Furthermore, the clusters would be nearly identical given the small step size of a gradient update for each meta-learning episode. Finally, this would be computationally intensive since unlike the typical applications of nonparametric mixture learning where one can afford to spawn hundreds of components then prune them over the training procedure.

Thresholding

A marked difference that is not immediate from the Gibbs conditionals is the use of a threshold on the cluster responsibilities, detailed in the E-STEP in Subroutine 4, to account for noise from stochastic optimization when spawning a cluster on the basis of a single batch. This threshold is necessary for the stochastic mode estimation procedure of Algorithm 6.3, as it ensures that a new cluster’s responsibility needs to exceed a certain value before being permanently added to the set of components.

If a cluster has close to an equal share of responsibilities as compared to existing clusters after accounting for the CRP penalty $\log n^{(\ell)}$ or $\log \zeta$, it is spawned. Accordingly, this approximate inference routine still preserves the preferential attachment (“rich-get-richer”) dynamics of Bayesian nonparametrics (Raykov et al. 2016). A sequential approximation for nonparametric mixtures with a similar threshold was proposed in

Lin (2013) and Tank et al. (2015), in which variational Bayes was used instead of point estimation in a DPMM.

Pruning heuristics

None of the results reported in our experiments used a pruning heuristic as we used a rather conservative hyper parameter setting that deters overspawning. We did however explore different heuristics which could work in more general settings, especially in the presence of many more latent clusters of tasks than considered in the experimental settings in this work. One such heuristic is to prune small clusters that have received disproportionately few assignments over a certain number of past iterations. Another is to evaluate the functional similarity of two clusters by computing an odds-ratio statistic for the assignment probabilities to each cluster over a set of validation tasks. If the odds-ratio statistic is below a certain threshold, the smaller cluster can be pruned.

Estimating the CRP hyperparameters

We fixed α at the size of the meta-batch. An alternative is to place a $\Gamma(1,1)$ on the concentration parameter. Based on the likelihood, the posterior is then proportional to $p(\alpha|N, K) \propto \frac{\Gamma(\alpha)}{\Gamma(\alpha+N)} \alpha^K e^{-\alpha}$. This is not a standard distribution but Rasmussen (2000) have shown that $\log p(\alpha|N, K)$ is log-concave and methods such as L-BFGS have been used successfully in prior works. Alternatively, if we have some prior knowledge about the expected number of clusters, we can compute α based on $E[K] = \alpha \log N$. For the window-size, we considered an initial size of 20 iterations that can grow as more cluster are considered.

Implementation details

We implemented both of our parametric and nonparametric meta-learners in TensorFlow (TF) (Abadi et al. 2016). We considered 2 different settings for the M-STEP optimization:

- Train each cluster’s parameters separately based on its corresponding loss function in an alternating manner closest to the classic EM algorithm.
- Train all cluster weights simultaneously using a surrogate loss over all validation batches.

Since the latter better leverages the differentiability of softmax-clustering and performed better empirically, we used it to report all experimental results.

Nonparametric Implementation

For the nonparametric algorithm, we chose the first approach to the M-STEP by constructing separate optimizers for each cluster’s parameters. We pre-allocate a set of weights and use

a mask during training to discard the parameters of empty clusters due to the static nature of TF graphs. When the algorithm exhausts the set of pre-allocated weights, we simply construct more network weight and reinitialize our optimizers.

CRP global prior

The likelihood of a new cluster is sensitive to the choice of a base measure or prior, G_0 on the cluster hyperparameters. Our gradient-based point estimation does not make any modeling assumption on the distribution of the weights, rendering the problem of principally updating the base measure, after or during training, non-trivial. We chose to initialize all weights with zero-mean normals in the fully connected layers. For the convolutional layers, we leveraged Xavier initialization Glorot and Bengio (2010) similarly to prior work Finn, Abbeel, et al. (2017) in ML.

However, such initialization is poor in the nonparametric for most non-trivial regression or classification tasks. Therefore, in the nonparametric setting, we start with a single cluster for a fixed number of iterations. We then initialize all clusters with the weights of the first clusters. This set of weights can be considered as the mean of the base measure or global prior in our setting.

We periodically update the global prior using a uniform average of the parameters of the existing clusters. This can be done by simply averaging over the parameter of the non-empty clusters as weighted by their sizes. Note that, we found that performing weighted KDE smoothing with a small bandwidth hyperparameter to perform slightly better than the average which is to be expected for neural network parameters. The number of iterations between updates of the global prior is a hyperparameter that we tune on the validation set. It is also possible to continuously, but less frequently over time, update this global prior as more data is encountered.

B.2 Rule- and exemplar-based generalization

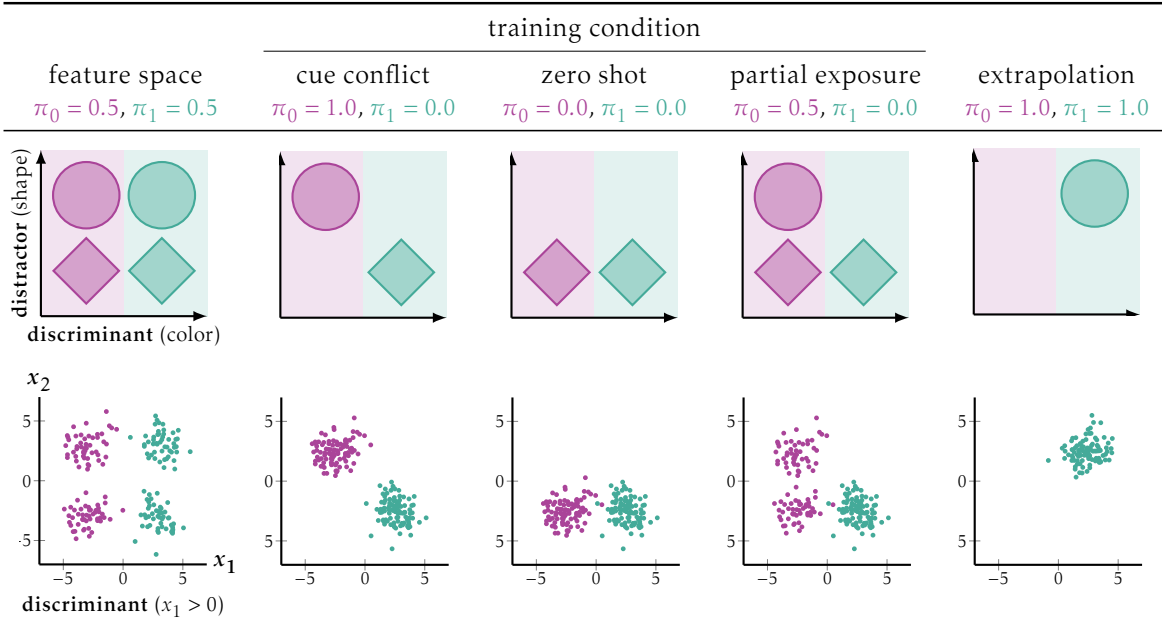


Figure B.1: We expand on Fig. 3.3 from the main text by including a realization of the abstract training conditions in the simple 2D points-in-a-plane setting. **(Top) Formalizing the illustrative experiment:** The experiment from Fig. 3.2 expressed in terms of the formalism in Section 3.3 with $\mathbf{z}_{\text{dist}} = \text{color}$ and $\mathbf{z}_{\text{disc}} = \text{shape}$. Background colors indicate true category boundary. **(Bottom)** The conditions realized via a binarization of continuous feature values. Here, the discriminant is binarized as $x_1 > 0$ and the distractor as $x_2 > 0$; this setting is further investigated in Section 3.4. Color here depicts the label but is not part of the input.

Generalizing the framework from two binary attributes to many categorical attributes

In the most general terms, we consider a setting in which each observation $\mathbf{x} \in \mathcal{X}$ is underlied by n categorical variables $z_1, \dots, z_n \in \{0, \dots, C\}$ with $C \in \mathbb{Z}_+$, henceforth *attributes* whose concatenation $\mathbf{z} = (z_1, \dots, z_n)$ determines the observable input \mathbf{x} via some mapping $g: \mathbb{Z}_{0+}^n \rightarrow \mathcal{X}$. We consider the binary classification task of fitting a model $\hat{f}: \mathcal{X} \rightarrow \{0, 1\}$ from a given model family \mathcal{F} to predict a binary label for each input. A subset of the attributes in \mathbf{z} , without loss of generality (z_0, \dots, z_i) , is taken to define the decision boundary, while the remaining attributes, z_{i+1}, \dots, z_n , are assumed to not be independently predictive of the true classification $y \in \{0, 1\}$. We therefore denote the *discriminant*, $\mathbf{z}_{\text{disc}} = (z_0, \dots, z_i)$, and the *distractor* $\mathbf{z}_{\text{dist}} = (z_{i+1}, \dots, z_n)$. For simplicity, we assume that the attributes are binary (*i.e.*, $C = 2$ and $z_i \in \{0, 1\}, \forall i$), and that the discriminant attributes must be jointly active for the classification to change from the null class $y = 0$ (*i.e.*, $y = 1 \iff \mathbf{z}_{\text{disc}} = \mathbf{1}$); the latter simplification allows us to redefine $\mathbf{z}_{\text{disc}} = z_0 \wedge \dots \wedge z_i$ and $\mathbf{z}_{\text{dist}} = z_{i+1} \wedge \dots \wedge z_n$, which is equivalent to the earlier discussion of the illustrative two-attribute case.

Training conditions expressed in terms of the joint distribution

We express the training conditions displayed in Fig. 3.3 and realized in Figure 6 in terms of the joint distribution instead of the parameters π_0, π_1 .

1. The cue-conflict condition the upper left and lower right quadrants in Figure 6 and defines the distribution of attributes as

$$\frac{p_{cc}(\mathbf{z}_{disc} = 0, \mathbf{z}_{dist} = 1) = 0.5 \quad | \quad p_{cc}(\mathbf{z}_{disc} = 1, \mathbf{z}_{dist} = 1) = 0}{p_{cc}(\mathbf{z}_{disc} = 0, \mathbf{z}_{dist} = 0) = 0 \quad | \quad p_{cc}(\mathbf{z}_{disc} = 1, \mathbf{z}_{dist} = 0) = 0.5 .}$$

2. The zero-shot condition populates the bottom left and right quadrants in Figure 6 and defines the distribution of attributes as

$$\frac{p_{zs}(\mathbf{z}_{disc} = 0, \mathbf{z}_{dist} = 1) = 0 \quad | \quad p_{zs}(\mathbf{z}_{disc} = 1, \mathbf{z}_{dist} = 1) = 0}{p_{zs}(\mathbf{z}_{disc} = 0, \mathbf{z}_{dist} = 0) = 0.5 \quad | \quad p_{zs}(\mathbf{z}_{disc} = 1, \mathbf{z}_{dist} = 0) = 0.5 .}$$

3. The partial-exposure condition populates all quadrants but the upper right in Figure 6 and defines the distribution of attributes as

$$\frac{p_{pe}(\mathbf{z}_{disc} = 0, \mathbf{z}_{dist} = 1) = 0.25 \quad | \quad p_{pe}(\mathbf{z}_{disc} = 1, \mathbf{z}_{dist} = 1) = 0}{p_{pe}(\mathbf{z}_{disc} = 0, \mathbf{z}_{dist} = 0) = 0.25 \quad | \quad p_{pe}(\mathbf{z}_{disc} = 1, \mathbf{z}_{dist} = 0) = 0.5 .}$$

CelebA results for specific model sizes

We include model-specific results, split by ResNet depth and width, in Fig. B.2. We find no systematic relationship between EvR and depth or width.

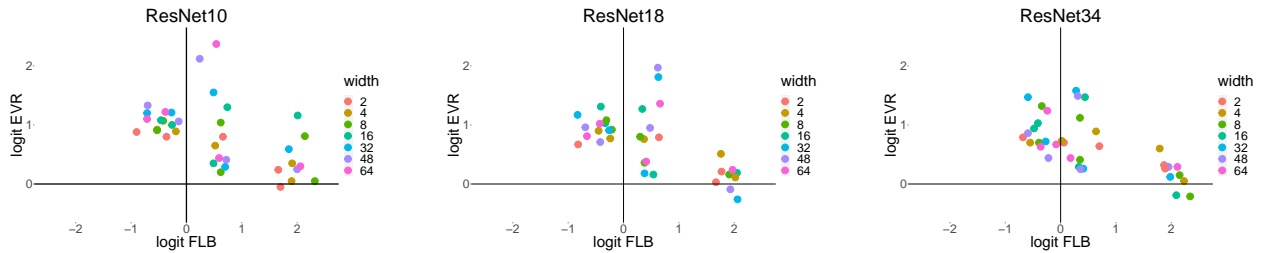


Figure B.2: CelebA EvR and FLB across feature pairs, averaged across 30 runs, split by depth and width of ResNet.

Spurious correlation underdetermines feature distributions

The partial-exposure condition ($\pi_0 = 0.5$, $\pi_1 = 0.0$) in Section 3.2 results in a spurious correlation between the discriminant \mathbf{z}_{disc} and the distractor \mathbf{z}_{dist} ($\rho = 0.58$). To examine behavior in a wider range of data settings, we vary π_0 and π_1 as described in Section 3.3, thereby also changing the degree of spurious correlation.

I. Interpolation towards zero shot. We interpolate π_0 from 0.5 towards 0.0, keeping $\pi_1 = 0.0$. This moves us closer to $\pi_0 = \pi_1 = 0.0$, where we have no exposure to $\mathbf{z}_{\text{disc}} = 1$ in training. Intuitively, we are reducing the exposure to the new distractor feature value from the partial-exposure condition.

II. Interpolation to full exposure. We interpolate π_1 from 0.0 towards 0.5, keeping $\pi_0 = 0.5$. This moves us closer to $\pi_0 = \pi_1 = 0.5$, where we have equal exposure to all quadrants in training. Here, rather than reducing the exposure to the new distractor feature value, we are equalizing the exposure to it across the discriminant dimension.

III. Interpolation with matched correlation. We report results on this in Sections 3.4 and 3.6. As also depicted in Fig. 3.4, we generate training conditions by changing π_0 and π_1 such that we follow a ρ -contour away from the partial-exposure condition ($\pi_0 = 0.5, \pi_1 = 0.0, \rho = 0.58$): solid contour in Fig. 3.4. We also match the spurious correlation across the two interpolations in Appendix B.2A and B: Fig. B.4 shows these additional ρ -contours as dashed lines.

These different interpolations are depicted in Fig. B.4a with different shapes and colors.

Generating interpolation points

We generate points along all three interpolation lines: from partial exposure towards zero shot (I); from partial exposure towards full exposure (II); and the equi-correlation line originating from partial exposure (III). The interpolating points along each line are selected to balance spurious correlation and feature exposure. In particular, we follow the following procedure:

1. We choose a point that interpolates towards full exposure. We do this by choosing a value of π_1 between 0.0 and 0.5, π^{FE} . This gives a data setting, along with a corresponding spurious correlation, ρ , computed via Eq. (3.3):

$$\pi_0 = 0.5 ; \quad \pi_1 = \pi^{\text{FE}} ; \quad \rho = \rho(0.5, \pi^{\text{FE}}) .$$

2. We generate a corresponding point that interpolates towards zero shot. Given the data setting above, we set $\pi_1 = 0.0$ and compute the π_0 to produce the same ρ as the full-exposure interpolations in Step 1. This gives the data setting:

$$\pi_0 = \pi^{\text{ZS}}(\pi^{\text{FE}}) ; \quad \pi_1 = 0.0 ; \quad \rho = \rho(\pi^{\text{ZS}}(\pi^{\text{FE}}), 0.0) = \rho(0.5, \pi^{\text{FE}}) .$$

3. Finally, we also derive the equi-correlation interpolation from the full-exposure interpolation as follows. We retain π_1 from the full-exposure condition, but recompute the

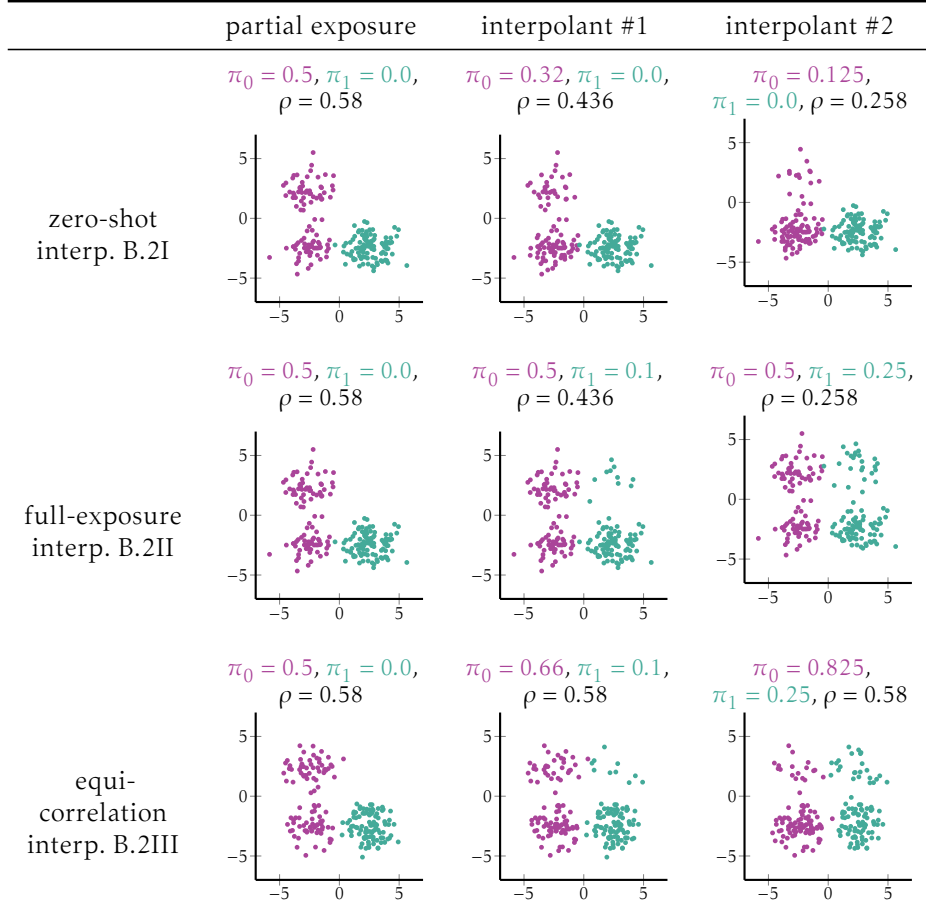


Figure B.3: We visualize several of the interpolants used for the interpolation analyses.

π_0 such that the correlation ρ matches the spurious correlation of the pure glspec ($\rho = 0.58$). This gives an additional data setting:

$$\pi_0 = \pi^{\text{EQ}}(\pi^{\text{FE}}); \quad \pi_1 = \pi^{\text{FE}}; \quad \rho = \rho(0.5, 0.0) = 0.58.$$

Note that, despite there being three different interpolation lines, the specific interpolants we use are constrained along a single degree of freedom—choosing π^{FE} (Step 1). The data settings for zero shot (Step 2) and equi-correlation (Step 3) are derived from this value.

Specific interpolation values used

For all data settings, we generate points along the interpolation lines using the procedure in Appendix B.2.

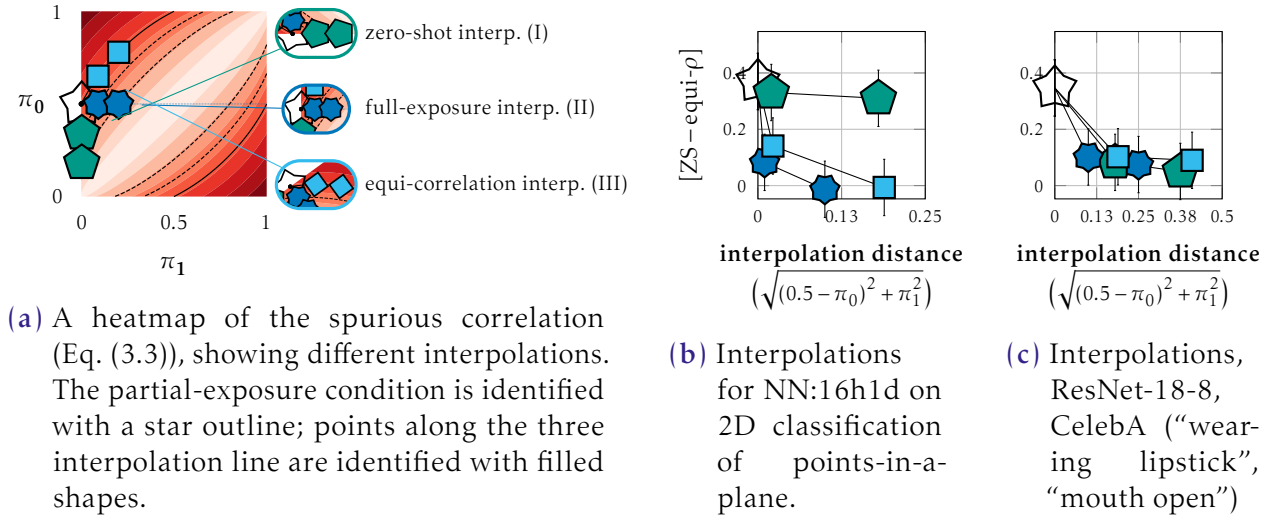


Figure B.4: Interpolations away from the PE: changes in extrapolation behavior under data distribution with the same spurious correlation as in PE, as well as different ways to change spurious correlation.

For the simple 2D classification setting, we examine two interpolants. In this simple domain, we keep the interpolation distances small, since we expect changes in extrapolation behavior even from small changes.

	interpolant 1			interpolant 2		
	π_0	π_1	ρ	π_0	π_1	ρ
interpolation to zero shot (I)	0.481	0.0	0.563	0.32	0.0	0.436
interpolation to full exposure (II)	0.5	0.01	0.563	0.5	0.1	0.436
equi-correlation interpolation (III)	0.519	0.01	0.58	0.661	0.1	0.58

For CelebA, we increase the interpolation distance to reflect the wider range of natural data distributions among feature pairs. The data these interpolation values generate is visualized as the equivalent points-in-a-plane setting in Figure 7.

	interpolant 1			interpolant 2		
	π_0	π_1	ρ	π_0	π_1	ρ
interpolation to zero shot (I)	0.32	0.0	0.436	0.125	0.0	0.258
interpolation to full exposure (II)	0.5	0.1	0.436	0.5	0.25	0.258
equi-correlation interpolation (III)	0.66	0.1	0.58	0.825	0.25	0.58

Interpolation analyses

In the 2-D classification example

In the simple setting from Section 3.4, we vary π_0, π_1 for an NN model (NN:16h1d, the NN with lowest EvR level overall). Results are in Fig. B.4b and discussed below.

EvR \neq **sensitivity to spurious correlation**. As also discussed in the main text, along the equi-correlation interpolation line, the “effective EvR” drops drastically (*i.e.*, the learner generalizes in more rule-based manner) despite no change in spurious correlation.

Implications for controlling extrapolation. Despite both having the same ρ , interpolating towards full-exposure increases the EvR more than towards zero-shot. This further supports that spurious correlation cannot fully characterize extrapolation behavior. This shows that different *ways* to reduce ρ have different effects on extrapolation, and has important implications for data manipulation methods (*e.g.*, subsampling or augmentation) that aim to directly control this ρ .

In CelebA

We see the same effects as in the linear setting: as also discussed in the main text, we see a much smaller gap to the ZS condition despite no change in spurious correlation. We don’t find clear effects distinguishing different ways to reduce spurious correlation (interpolation to zero shot (I) and interpolation to full exposure (II)).

Additional dataset details

We provides relevant statistics of each dataset, such as number of examples.

	2D	IMDb	CelebA*
dataset size (train)	75	21,215	4,000 to 40,000
dataset size (valid)	75	21,027	8,000
dataset size (test)	75	13,995	20,000
input space	\mathbb{R}^2	\mathbb{R}^{400}	$\mathbb{R}^{178 \times 218 \times 3}$

We do not use a validation set for the simple 2D classification setting and IMDb datasets, but hold out examples for a test set. For CelebA, we follow the authors’ division of images in train, validation and test splits.

As described in the main text, we subsample data to balance attributes within each training condition. For the CelebA domain, we use the following feature pairs to produce the results in Section 3.6:

*The numbers for CelebA are approximate because there are deviations in the availability of images across attribute combinations.

discriminant	distractor
mouth open	male
wearing lipstick	mouth open
male	mouth open
male	high cheekbones
male	blond hair
male	arched eyebrows

For Section 3.5 and Section 3.6, we used publicly available datasets: IMDb (Maas et al. 2011) is available at <https://ai.stanford.edu/~amaas/data/sentiment/>; CelebA (Liu, Luo, et al. 2015) at <https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>.

Additional experimental details

We use default hyperparameter settings whenever possible. For the points-in-a-plane and IMDb settings, we use 20 random seeds, which randomize the model weight initialization. For the CelebA domain, we run 30 seeds for each model configuration, and discard runs that achieve below 75% accuracy on validation set images that belong to the data conditions (quadrants) observed during training.

We report accuracy as a performance metric on each of the four quadrants depicted in Fig. 3.3 as well interpolating data settings. We additionally report measures that are the performance difference between data settings. We include a 95% confidence interval on all reported measures.

B.3 Gaussian process surrogate models

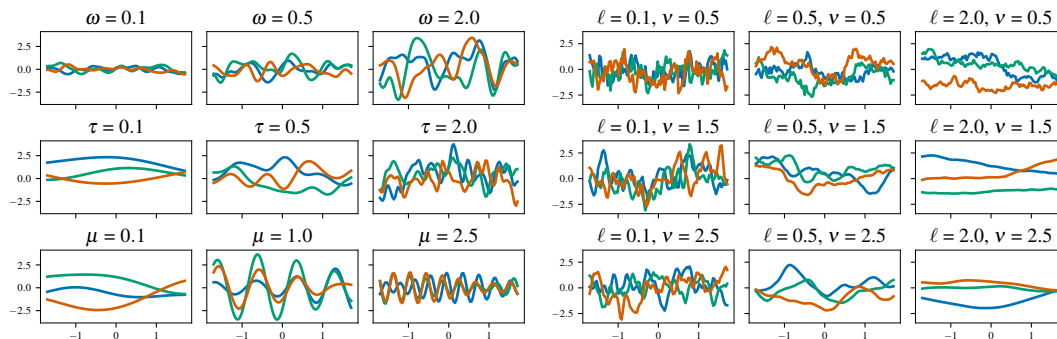


Figure B.5: Illustrating the effect of GP kernel hyperparameters on the GP prior. (Left) Samples from a GP prior with SMK with varying mixture weights ω , mixture scale τ , and mixture means μ . **(Right)** Samples from a GP prior with Matern kernel with varying ν and ℓ (lengthscale). GPs are flexible models whose properties can be controlled through hyperparameters.

Properties of the spectral mixture kernel and the Matérn kernel

We describe how the various hyperparameters of the SMK and MK kernel affect the GP prior. We begin with the spectral mixture kernel. The mixture weights w are signal variances and control the scale of the function values. The mixture means (μ) encode periodic behavior. The variances (τ) are (inverse) lengthscales, which control the smoothness. The (ARD) MK kernel has lengthscales θ , which controls the smoothness of the function with respect to each dimension. ν is another hyperparameter that also modulates smoothness, and the Matern covariance function admits a simple expression when ν is a half-integer. $\nu = 2.5$ corresponds to twice differentiable functions and $\nu = 1.5$ corresponds to once differentiable functions.

In Fig. B.5, we vary the hyperparameters of the SMK (w, μ, τ) and Matern kernels (ν, θ) and illustrate how they impact the prior over functions.

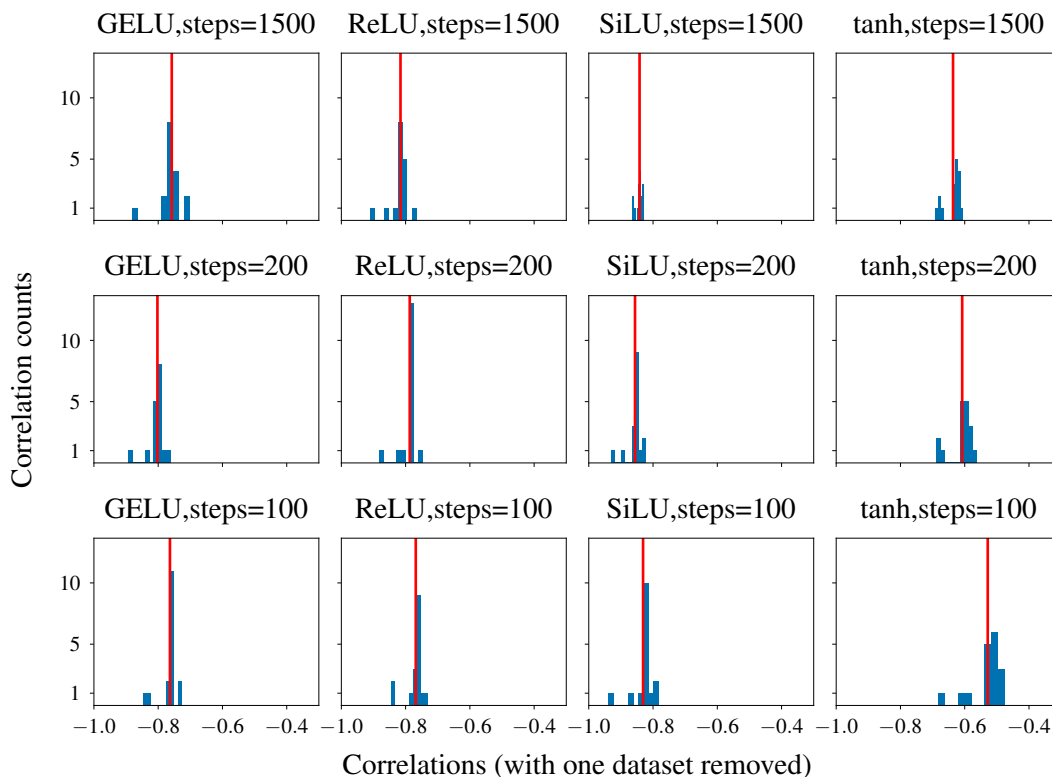


Figure B.6: Sensitivity analysis of generalization gap and lengthscale profile relationship. Each panel a histogram and mean (red line) of correlations obtained by recomputing the correlation between lengthscale profile correlation and generalization gap after removing each UCI dataset. Across datasets and architectures, even when a single dataset is removed, there remains an negative correlation between generalization gap and lengthscale profile correlation. Therefore, the inverse relationship between generalization gap and lengthscale profile correlation demonstrated in Fig. 7.10 is robust to outlier datasets.

Correlation sensitivity

We present some additional results to supplement our analysis from Section 7.4 where we demonstrated that discrepancy in lengthscale profiles between data and neural network predicts the generalization gap. Correlation can be sensitive to outliers. Does any single dataset account for the negative correlations? To answer this, we characterize how the correlation changes as a result of dropping each dataset. Specifically, for each UCI dataset, we remove that dataset and then compute the correlation between lengthscale profile correlation and generalization gap for the remaining datasets. We plot the resulting distribution of correlations in Fig. B.6. We find there is a tight spread around the correlation computed from all the UCI datasets. Importantly, when we remove any UCI dataset, we still see moderate to high negative correlations between lengthscale profile correlation and generalization gap.