THE UNIVERSITY
of EDINBURGH

Meta Learning for Few-shot Learning

Xueting Zhang

School of Informatics
The University of Edinburgh

Thesis submitted for the degree of
Doctor of Philosophy

October 2021

I would like to dedicate this thesis to
my loving parents and friends

Abstract

Few-shot learning aims to scale visual recognition to open-ended growth of new classes with limited labelled examples, thus alleviating data and computation bottleneck of conventional deep learning. This thesis proposes a meta learning (a.k.a. learning to learn), paradigm to tackle the real-world few shot learning challenges.

Firstly, we present a parameterized multi-metric based meta learning algorithm (RelationNet2). Existing metric learning algorithms are always based on training a global deep embedding and metric to support image similarity matching, but we propose a deep comparison network comprised of embedding and relation modules learning multiple non-linear distance metrics based on different levels of features simultaneously. Furthermore, images are represented as a distribution rather than vectors via learning parameterized Gaussian noise regularization, reducing overfitting and enable the use of deeper embeddings.

We next consider the fact that several recent competitors develop effective few-shot learners through strong conventional representations in combination with very simple classifiers, questioning whether "meta-learning" is necessary or highly effective features are sufficient. To defend meta-learning, we take an approach agnostic to the off-the-shelf features, and focus exclusively on meta-learning the final classifier layer. Specifically, we introduce MetaQDA, a Bayesian meta-learning extension of quadratic discriminant analysis classifier, that is complementary to advances in feature representations, leading to high accuracy and state-of-the-art uncertainty calibration performance in predictions.

Finally, we investigate the extension of MetaQDA to more generalized real-world scenarios beyond the narrow standard few-shot benchmarks. Our model achieves both many-shot and few-shot classification accuracy in generalized few-shot learning. In terms of few-shot class-incremental learning, MetaQDA is inherently suitable to novel classes growing scenarios. As for open-set recognition, we calculate the probability belonging to novel class by Bayes' Rule, maintaining high accuracy in both close-set recognition and open-set rejection.

Overall, our contributions in few-shot meta-learning advance state of the art under both accuracy and calibration metrics, explore a series of increasingly realistic problem settings, to support more researchers and practitioners in future exploration.

Declaration

I declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. Except where otherwise acknowledged, the work presented is entirely my own.

Xueting Zhang
October 2021

Acknowledgements

It is my super honor to express my deep gratitude for those who make this thesis possible. First and foremost, I am tremendously grateful to thank my supervisor Timothy M. Hospedales for his continuous support and guidance throughout my PhD, providing me freedom and instruction during my research and life. Tim is the coolest guy in my mind with his incessant enthusiasm, immense knowledge, shining personality, and absolutely charming Tango .

I would like to thank my second supervisor Dr. Oisin Mac Aodha and Dr. Charles Sutton, for their constructive suggestions and fruitful comments on my research. It is such a great opportunity to collaborate with the amazing Machine Intelligence Group (MIG) during my PhD, including Dr. Yongxin Yang, Dr. Henry Gouk, Dr. Yuting Qiang, Flood Sung, Debin Meng, etc., who contribute incredible talent, overnight discussion and crucial advice to this piece of research.

My sincere thanks also go to all my friends in the magical city Edinburgh, who tolerate my picky taste and feed my stomach and mind with generosity. Most importantly, this thesis is dedicated to my parents, family and hometown chums for all the years of your love, support and understanding throughout this adventure.

Contents

List of Figures

List of Tables

Nomenclature

AI	Artificial Intelligence
BN	Bayes Network
CL	Continual Learning
CNN	Convolutional Neural Network
DA	Domain Adaptation
DG	Domain Generalization
DCN	Deep Comparison Network
DL	Deep Learning
EA	Evolution Algorithm
FSL	Few-Shot Learning
FSCIL	Few-Shot Class Incremental Learning
FSOSR	Few-Shot Open-Set Recognition
GPU	Graphics Processing Unit
GFSL	Generalized Few Shot Learning
LR	Linear Regression
LDA	Linear Discriminant Analysis
MAML	Model Agnostic Meta Learning
ML	Meta Learning
MetaQDA	Meta Quadratic Discriminant Analysis
OSR	Open-Set Recognition
PN	Prototypical Network
RN	Relation Network
RL	Reinforcement Learning
SGD	Stochastic Gradient Descent
TL	Transfer Learning

Chapter 1

Introduction

Machine learning is an important branch of Artificial Intelligence (AI). It is the study of learning from experience (data) to train a program (model) and therefore to solve a problem (task), like humans can do. However, conventional machine learning algorithms require an abundance of labelled data to build a standard training paradigm to solve a specific task, and lack the capacity to learn from only a few instances. Meta learning aims to address this by improving the learning algorithm itself through a learning-to-learn process. This provides a mechanism to tackle some challenges in contemporary meta learning applications such as few-shot learning in computer vision. That is, how we can leverage prior knowledge to recognize new categories based on only a few labelled samples. This thesis focuses on meta-learning algorithms based on statistics and deep learning approaches, and explores a series of few-shot learning problems from simple academic benchmarks to more challenging real-world scenarios, which aims to achieve the state-of-the-art performance and provide guidelines to further research.

1.1 Background

Deep-learning based approaches have yielded great successes when applied to numerous fields, such as image classification, speech recognition, language translation and robotics [56, 149, 26]. This is partly due to the development of powerful computing resources, such as GPU clusters and distributed high-performance platforms, and enormous benchmark datasets, e.g., ImageNet [25] and MetaDatast [164]. Modern machine learning technologies aim to bridge the gap between computers and humans to achieve greater capability and autonomy, which has so far been highly successful in a variety of real-world daily-life applications, e.g., face recognition [197, 67, 50], machine translation [167, 31], search engines [120], unmanned aerial vehicles (UAVs) [2, 184] and recommendation systems [24, 178, 22].

State-of-the-art deep learning models are data-hungry and time consuming, and thus still only effective in areas where big data and computations are available. Specifically, contemporary machine learning systems require high human cost to prepare the annotation, and expensive computing resources to train the neural networks with millions of parameters [81]. For example, a shallow basic 9-layer convolutional neural network (CNN) can have 60 million parameters and 650,000 nodes, which needs to be trained on a million distinct samples. Furthermore, the popular BERT model [26] has more than 3 billion parameters, which necessarily costs 64 NVIDIA V100 GPUs training 4 days and utilizes an effective 1507 kilowatt-hours of electricity.

In principle, once given infinite data, the brute force application of deep learning is powerful enough to represent deterministic mapping from instances into a finite set of categories (e.g., ImageNet). But in reality, it is not possible to always collect abundant data for all the tasks or domains, not only due to laborious and costly annotation, but also considering the issues related to privacy, safety and ethics, etc. There are a lot of low-data regimes where the labelled data can be hard or even impossible to acquire. For example, due to potential toxicity and shortage of clinical records, drug discovery often lacks sufficient samples to detect and analyze the properties of new molecules [1]. Also, the data owner may not want to share the raw data publicly for social fairness or commercial sensitivity, and data annotation by specialists is expensive due to the requirement of expert knowledge [112, 168]. Data scarcity (e.g. for rare animal species) and annotation budgets (e.g., for medical images) create an application bottleneck for deep learning algorithms. Therefore, we need to propose more practical algorithms to relieve the burden of the requirement of large-scaled supervised data [180, 103].

Humans seem to hold the advantage of reusing their previous knowledge and extracting abstract explicit definition and implicit meanings. It is easy for a child to learn the concept of tiger after recognizing cat, and to classify the difference between a hand-written digit and a printed digit after only a few instances. However, conventional machine learning is different from humans. When facing with a new task, the classic paradigm becomes useless without leveraging prior knowledge, and may need to be stated again from scratch. Thus, generalization across tasks could enhance the essential ability to learn from a few samples, which is an important property of human intelligence.

To solve the complex, data-scarce learning problems, there are several popular approaches in the literature. Classic deep learning approaches treat each learning problem as tabula-rasa [81, 56, 66], whereas *transfer learning*, as an effective methodology, is used to enhance the learning of the target task given a source task (problem) by parameter transfer and optional fine-tuning (methodology) [119, 135]. Source and target tasks can often have different feature

spaces and data distribution. For example, in computer-aided detection problems, a CNN model pre-trained on ImageNet is required to be transferred to a specific medical image dataset such as interstitial lung disease classification [148], in order to achieve the required level of accuracy. Transfer learning sometimes also refers to a problem setting rather than a methodology. In this case, it has an overlap with few-shot learning problem setting, because both address reducing the data requirements for the target task.

Deep learning methods typically attempt to use prior knowledge as little as possible, which enabled them to achieve excellent performance when trained on large datasets. However, exploiting prior knowledge is crucial when training on small datasets. For example, *Bayesian models* hold mathematical interpretability, allowing prior knowledge to be encoded explicitly. Nevertheless, using Bayesian models by itself cannot compete with the high performance of current deep learning approaches. In this case, it is crucial to exploit the strengths of both Bayesian and deep learning methods to achieve strong few-shot learning performance.

1.2 Meta Learning

Meta-Learning aims to improve a learning algorithm over a distribution of tasks, while conventional deep learning gains knowledge from a given dataset or task. Specifically, meta learning applied to neural network further advances the frontier of deep learning to integrate joint feature, model and algorithm learning, in order to improve the performance of future tasks with both data and computing efficiency. This thesis focuses on meta-learning style approaches to improve the data efficiency in few-shot problems. The methodologies can be broken down into three main taxonomies: optimization-based methods (e.g. MAML [34, 35]), model-based methods (e.g. MANN [141]) and metric-based methods (e.g. Prototypical Network [150]). We explore both model-based and metric-based approaches to enhance the few-shot recognition performance. In this research area, there are various methods including Siamese Networks [78], Matching Networks [170], Prototypical Networks [150], Relation Networks [159], and Graph Neural Networks [40].

We propose RelationNet2 (in Chapter 3) deploying parameterized multi-metrics to achieve parallel learning of different level feature embeddings and similarity matching metrics. We also uniquely propose an amortized Bayesian meta learning approach MetaQDA (in Chapter 4). Importantly, where standard deep learning methods suffer from poor uncertainty calibration, MetaQDA also provides state-of-the-art uncertainty calibration in its predictions. As computer vision is used in more safety critical applications, proper *calibration* plays an important role in addition to the average accuracy. Models should report the confidence of predictions, allowing one to double check low-confidence decisions to avoid disastrous errors [49].

We are concerned with not only the academic but also the real-world applications of few-shot learning problems. Current few-shot benchmarks are limited to the hand-crafted episodic C-way-K-shot formulation, but are not quite suitable to the more complicated real-world challenges. In the standard few-shot learning problem setting, the source many-shot samples are only used to train the meta-learning model, before updating it with the few-shot categories. During testing, the model is then only prevented with the few-shot categories for evaluation. However, in practical applications both many-shot seen classes and few-shot unseen classes are of interest, e.g., real-time interactive vision applications for portable devices [188, 44, 93]. *Generalized few-shot learning (GFSL)* evaluates model performance on both old and novel instances at test time. This therefore requires models to solve the catastrophic forgetting problem [44, 147, 65]. Similarly, another real-world challenge is that in *few-shot incremental learning*, unseen few-shot classes could be provided incrementally in a stream rather than as a batch [131, 126, 53], such as on-device deployment in robotic scenarios. However, many popular gradient based meta learning methods (e.g., MAML [34]) cannot tackle this challenge. A conventional classification system is assumed to operate in a closed-set condition where all training (support) and testing (query) examples are from the same label space. However, we actually often encounter unseen samples from the open-ended real world, which means unknown instances would be fed into the model during the testing procedure. These should be rejected as unknown instances instead of being wrongly classified to some known classes [142, 5, 199]. *Open-Set Recognition (OSR)* requires dual functions of anomaly detection and close-set classification. The difference from generalized few-shot learning is that we could hardly get any "feature" from the "unknown open-set" to realize detection, but we need to retain both the classification and rejection performance. This thesis endeavors to propose meta learning methodologies which can be easily adapted to the above more realistic problem settings and compares these to the existing approaches to show the advantages of our model.

In this thesis, we conduct research on meta learning methods applied to few-shot learning problems. The proposed approach formulation is based on these hypotheses: (1) The class distribution could be formulated as a The main contributions are as follows: (1) We demonstrate RelationNet2, which is a non-linear parametric deep metric-based meta learner that achieves state-of-the-art few-shot learning performance. (2) We then introduce MetaQDA, a hybrid deep Bayesian approach to few-shot learning that conveniently decomposes the representation learning and classifier meta-learning problems, which can further improve the few-shot learning performance. (3) We explore the extension of MetaQDA to a variety of increasingly real-world problem scenarios including generalized, incremental, and open-set few-shot learning.

1.3 Thesis Outline

The thesis consists of 3 parts with 6 chapters, as shown in Figure 1.1:

1. Introduction

Chapter 1 We demonstrate an introduction to the research area, including our research motivation, goals and contributions, providing an overview of the thesis structure and contents.

Chapter 2 We investigate the related work and summarize the background of meta learning and few shot learning, providing an overview literature review with the terminology and taxonomy. Overall, we focus on how to use meta-learning methods to improve the few shot learning problems in different scenarios.

2. Main Work and Contributions

Chapter 3 We propose our first contribution, a simple but powerful metric-based few-shot learning algorithm named RelationNet2/DCN, comprised of embedding and relation modules learning multiple non-linear distance metrics based on different levels of features. Furthermore, image features are presented as distribution rather than vectors via learning parameterized Gaussian noise regularization. This work achieves SotA performance on various standard FSL benchmarks, and the insight is verified effectively. This work was published as an oral conference paper in IJCNN'2020.

Chapter 4 We then introduce our second contribution, an orthogonal few-shot learning direction of Bayesian shallow meta-learning classifiers, which is easy to plug in any off-the-shelf extracted features. By decomposing the representation learning and classifier meta-learning issues in few-shot visual recognition, MetaQDA enables few-shot meta-learning to benefit from future advances in supervised representation learning. Furthermore, the probabilistic approach MetaQDA outperforms the other algorithms in terms of superior uncertainty calibration. We also show that thanks to this decomposition of classifier meta-learning and representation learning, MetaQDA can draw upon features designed for multi-domain data, and also achieve state of the art cross-domain few-shot learning. This work was published as a conference paper in ICCV'2021.

Chapter 5 We move to our third contribution, which extends our Bayesian MetaQDA model to several more realistic scenarios, from generalized few-shot learning (GFSL), few-shot class incremental learning (FSCIL) to few-shot open-set recognition (FSOSR). Each problem setting is much more complicated than fixed C-way-K-shot standard paradigm. The model should keep both the many-shot and few-shot classification accuracy without forgetting, support incremental addition of new categories, and maintain high performance on both close-set recognition and open-set rejection.

3. Discussions and Future Work

Chapter 6 We provide conclusion and discussion about open challenges in meta learning, especially some valuable open questions on few shot learning, including some of the potential directions of future work, such as deep Bayesian meta-learning, multi-domain meta learning, life-long meta learning, and few-shot learning beyond object recognition.

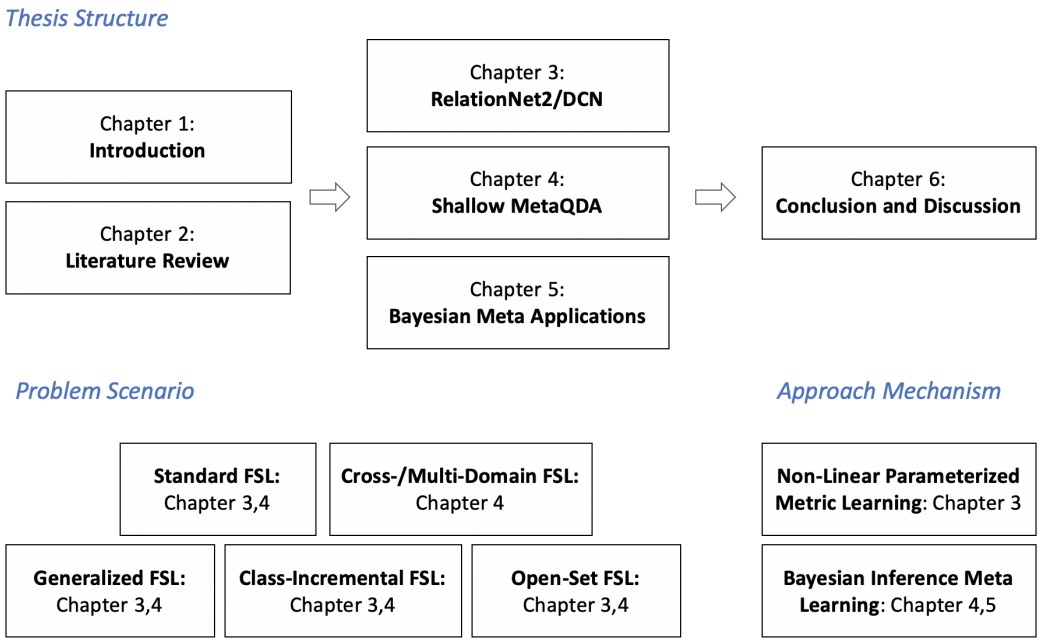


Figure 1.1 **Outline of the thesis.** The thesis deploys meta-learning approaches to various few-shot learning scenarios. We show the thesis structure from both the view of problem scenario and approach mechanism.

Chapter 2

Literature Review

This chapter provides a brief review of the underlying methodologies of meta learning and few-shot learning, along with the problem settings and applications. Specifically, we investigate related literature and summarize the contributions and limitations, not only listing the terminology and taxonomy of meta learning, but also how to solve different few-shot learning problems from academic research to various commercial and industrial applications. We then introduce more specialized details corresponding to the specific research topics in each technical chapter.

2.1 Background in Meta-Learning

In this section, we introduce the fundamental concepts and development of the research fields related to meta learning. We also propose a distribution-view formalization of meta learning, providing a general understanding from a data distribution point of view [7]. Finally, we organize the existing methods with a taxonomy reflecting the landscape of meta-learning.

2.1.1 Related Research Fields

Meta-learning first appeared in the literature in 1987, proposing a theoretical framework of self-referential learning which can learn weights and predict updates, with the fast/slow-weights corresponding to optimizer updates and inference itself [61, 144]. Recent contemporary meta-learning has a resurgence along with the advancement of deep learning algorithms and GPU computing capabilities. Gradient descent and back-propagation have been used since 2001 [136], and contemporary meta learning is introduced in [163], marking the beginning of modern meta learning.

Meta learning is often confused with some related research areas, so we explain the connections to and differences from the other fields. *Transfer Learning (TL)* [119, 193, 135] refers to a model which is trained on a source task with sufficient data to improve performance on a target task with a different label space, but meta learning deals with a high-level objective to learn the "learning algorithm" itself by much more wide meta-representation or meta-optimizer. For example, MAML [34] learns the prior by an outer optimization that evaluates how well the prior performs when helps learning a new task, instead of simply extracting the corresponding prior from the source tasks. *Domain Adaptation (DA)* and *Domain Generalization (DG)* are both domain-shift issues where the source and target tasks have different distributions reducing the model performance. DA can utilize unlabelled data from the target domain, while DG does not have any access to target domain data during the training process [119, 23]. Meta-learning methodologies can be designed to solve both DA and DG problems [89]. *Continual Learning*, a.k.a. *Lifelong Learning*, accelerates the learning of new tasks without forgetting old tasks by training on a sequence of tasks drawn from a potentially non-stationary distribution [121, 21]. However, meta-learning proposes a framework to improve lifelong learning by overcoming the difficulty of encoding the meta-objective [146, 134, 111]. *Hierarchical Bayesian Models (HBM)* are a theoretically valuable viewpoint for meta-learning, including Bayesian learning of parameters θ through a prior $p(\theta|\omega)$, where ω is a kind of meta knowledge. Instead of providing an algorithmic framework, Bayesian inference appears to provide a model for understanding the intrinsic intuition of meta learning. For example, *Latent Dirichlet Allocation* [16] uses Bayesian marginalization due to the conjugate exponential models, and a stochastic variational approach [30] calculates an approximate posterior from which a lower bound to the marginal likelihood is computed.

Moreover, contemporary neural network meta learning is a methodology framework used to solve more challenging problems, which is always conducted as an end-to-end optimization of the inner algorithm with respect to an explicitly defined meta-objective [62, 183, 100].

2.1.2 Formalization of Meta Learning

Meta-learning aims to improve the learning algorithm over multiple learning episodes, which involves a hierarchical optimization problem, namely a tuple including base algorithm, trained model and performance. *Base-learning* is the inner learning algorithm itself such as image classification or language translation, but *meta-learning* is the outer learning to update the inner algorithm with a high-level meta-objective. For example, *learning speed* of the inner algorithm refers to either training examples (sample efficiency) or optimization iterations (convergence rate), and *generalization performance* refers to the performance of the learned

model on a held out validation set. Here we depict then mathematical formalization of meta-learning.

Conventional machine learning is commonly understood to improve the model performance from scratch for each task. Given a training dataset $D = (x_1, y_1), ..., (x_N, y_N)$ as the pair of an input instance and the corresponding label, then we can train a model $\hat{y} = f_\theta(x)$ parameterized by θ, where θ is usually specific to the application, e.g. a convolutional neural network (CNN) in the case of computer vision [81] or a recurrent neural network (RNN) in the case of natural language processing [26]. The objective is

$$\theta^* = \arg\min_\theta \mathcal{L}(\mathcal{D}; \theta, \omega), \tag{2.1}$$

where $\mathcal{L}$ is the loss function to evaluate the predicted label accuracy comparing to the ground truth label, and ω is the dependency on explicit condition such as function for f or optimizer for θ. The specialization of ω determines the learning process of θ with great influence on generalization, computation cost and data efficiency.

Meta-learning prefers to learn a more general purpose across multiple learning episodes sampled from a task family, where ω is referred to as *meta-knowledge* across different tasks. In particular, it prefers to learn the ω from a distribution of tasks rather than fix the hand-crafted ω. Meta-knowledge ω is evaluated as the capability of how to learn over a distribution of tasks $p(\mathcal{T})$, where $\mathcal{T} = \{\mathcal{D}, \mathcal{L}\}$. The meta-learning process becomes

$$\omega^* = \arg\min_\omega \mathop{\mathbb{E}}_{\mathcal{T} \sim p(\mathcal{T})} \mathcal{L}(D; \omega), \tag{2.2}$$

where $\mathcal{L}(\mathcal{D}; \omega)$ indicates the performance of a model with trained ω on dataset $\mathcal{D}$.

We need to generate M training tasks from $p(\mathcal{T})$ instead of training instances. During meta-training process, we mimic the meta-testing process to provide the supervision labels to measure the meta-knowledge, so there are *support (s)* and *query (q)* set corresponding to the train and validation roles, $\mathcal{D}_{m-train} = \{(\mathcal{D}^s_{m-train}, \mathcal{D}^q_{m-train})^{(i)}\}_{i=1}^M$. Then the objective of meta-training step is

$$\omega^* = arg\max_\omega log p(\omega | \mathcal{D}_{m-train}). \tag{2.3}$$

Similarly, during meta-testing, we have N testing tasks as $\mathcal{D}_{m-test} = \{(\mathcal{D}^s_{m-test}, \mathcal{D}^q_{m-test})^{(i)}\}_{i=1}^N$, and then we could use the learned *meta-knowledge* ω to train the base model on the $i-th$ unknown target testing tasks:

$$\theta^{*(i)} = arg\max_\theta log p(\theta | \omega^*, \mathcal{D}^{s(i)}_{m-test}). \tag{2.4}$$

2.1.3 Methodology Taxonomy

We prefer to classify the existing methodologies into three axes as meta-representation, meta-optimizer, and meta-objective [62], reflecting the cutting-edge researches on meta-learning from a big-picture perspective.

Meta Representation ("What") There are several possible choices of learning meta-knowledge ω. *Parameter initialization* considers ω as the initial parameters of a neural network, e.g. MAML [34, 35] is a typical family of methods meta-learning the initial condition of the inner optimization. But the challenge whether one initial condition is limited to narrow distribution $p(\mathcal{T})$ led to other approaches with model mixtures over multiple initial conditions [138, 171]. *Optimizer* methods replace the hand-crafted optimizers, e.g. widely used stochastic gradient descent (SGD), with a learned optimizer defined by ω [129, 3, 179]. *Black-box models* train ω to learn a feed-forward mapping from the support set $\theta = g_\omega(\mathcal{D}^{train})$ with a black-box CNN or RNN [106, 27, 173]. *Metric-learning* is always applied to few-shot learning, which is an important application of meta-learning, also is what we focus on in this thesis. Making the embedding task-conditional and learning an elaborate comparison metric are widely used to enhance the performance [159, 40, 116]. *Loss-learning* aims to learn the base model task-loss $\mathcal{L}_\omega^{task}$, improving the generalization ability with less local minima, especially in self-supervised or auxiliary learning. *Architecture learning* uses reinforced learning, long short term memories (LSTMs) and evolution algorithms to learn a better architecture which can be directly applied to meta-testing.

Meta Optimizer ("How") The choice of outer (meta) optimization strategy for ω is divided into gradient-descent, reinforcement learning, and evolutionary search. *Gradient Descent* methods exploit analytical gradients of $d\mathcal{L}/d\omega$ by computing derivatives [34, 129, 36], with the challenge of second-order gradients [114], the inevitable gradient degradation and some non-differentiable operations. *Reinforcement Learning (RL)* estimates the gradient alleviating the requirements of differentiability but with tremendous computing cost. *Evolution Algorithms (EA)* also relieve the differentiability and back propagation, and are highly parallelizable and lower costly [154, 139]. However, the population size is exponentially increased and the mutation strategy is sensitive to hyperparameters, thus evolution algorithms are often applied in combination with reinforcement learning [64].

Meta Objective ("Why") The choice of meta-objective, from a bilevel optimization view, is related to the outer objective $\mathcal{L}^{meta}$, the task distribution $p(\mathcal{T})$ and the associated data-flow between inner-loop episodes and outer-loop optimization. *Many-/Few-shot episode design* could be defined by generating tasks with the labelled examples to improve the performance [36, 179, 34, 129]. *Fast adaptation* encourages fast base task learning, where the validation

loss is calculated after each inner-loop episode [4, 10, 179]. *Multi/Single-task* requires the inner-loop learning episodes drawing data from the same specific task or various tasks from a given family $p(\mathcal{T})$ with different value propositions, separately [150, 3, 183]. *Offline learning* defines the meta-optimization as an outer-loop of the inner base learner [107], while *online learning* approaches perform the meta-optimization within a single base learning episode to co-evolve with higher compute efficiency [91, 9].

2.2 Background in Few-shot Learning

In this section, we introduce few-shot learning, one of the most popular applications of meta-learning in the computer vision domain, motivated by the challenge of the long-tail distribution of image recognition. We depict the problem setting and benchmarks in this research field, by utilizing both the taxonomies and terminologies.

2.2.1 Literature Review

Few-shot learning is hard to solve by using traditional deep neural networks, whereas human intelligence can sufficiently learn a classification rule from a few labelled samples [170]. Various methodologies are proposed to overcome the challenge of overfitting in the few examples of novel classes, e.g., transfer learning, semi-supervised learning, and meta-learning approaches [135, 156, 78, 34]. However, this thesis focuses on the metric-based and model-based meta-learning methods for few-shot learning problems, which enable a deep neural network to be successfully applied to small datasets.

Specifically speaking, few-shot multi-class classification is one of the basic problems in image recognition. Metric based meta learning approaches learn the feature encoder and the distance measurement, then recognize the unknown few-shot instances by comparing to the known train images using the learned metric [159]. The intuition is that images from the same class are located closer to each other in the feature space, and different classes would be further apart from each other. A meta-learning strategy could be used for different components, namely feature embedding, class representations, and similarity measures.

From both a theoretical and practical perspective, few-shot learning has three aspects of significance: (1) A reduced reliance on large-scale training samples and relieve the annotation cost; (2) Bridging the gap between human intelligence and artificial intelligence; (3) Achieving quick deployment for real-world applications. Consequently, researchers have previously shed light on few-shot learning in the past, non-deep era (e.g., congealing algorithm [105], variational Bayesian framework [32]), and the advancement of deep-learning has recently

attracted increasing attention. Here we put emphasis on the deep-learning period beginning with [78], which incorporated a siamese convolutional neural network to learn a class-irrelevant similarity measure. It seems that the discriminative model based few-shot learning algorithms dominates in recent developments, and a meta-learning framework is widely used in this area, such as Matching Nets [170], MAML [34], meta-LSTM [129], MANN [141], MetaNet [108], Prototypical Nets [150], Relation Net [159] and etc.

2.2.2 Problem Setting

Firstly, we consider a C-way K-shot classification problem for few shot learning. There are some labelled source tasks with sufficient data, denoted as meta-train dataset $\mathcal{D}_{\text{m-train}}$, and we ultimately want to solve a new set of target tasks denoted as meta-test dataset $\mathcal{D}_{\text{m-test}}$, for which the label space is disjoint $\mathcal{D}_{m-train} \cap \mathcal{D}_{m-test} = \emptyset$. Within each episode of meta-train and meta-test process, we denote each task $\mathcal{T}$ as being composed of a support set of training examples, and a query set of testing examples. Meta-test tasks are assumed to be few-shot, so $\mathcal{D}_{\text{m-test}}$ contains a support set with C categories and K examples for each category, and the query set has K' examples for each class. Then the support set has $C \times K$ instances, and the query set has $C \times K'$ examples. The meta-training process generates tasks to mimic the meta-testing process, and the evaluation of the model is the average prediction accuracy performance of the query set. In this context, we want to learn a model during meta-training that can generalize out of the box, without fine-tuning, to learning the new categories during meta-testing. The problem setting could be illustrated as Figure 2.1.

Formally, we use x to represent the feature of input data, y to represent the label of the data, so the dataset contains of data-label pairs $\mathcal{D} = (x_i, y_i)$, and $\mathcal{X}$ and $\mathcal{Y}$ to denote the space of input data feature space and label space, respectively. We introduce the principle to model few-shot learning problems. As for one *query* sample x_j, the algorithm predicts the label $\hat{y}_j$ with the highest posterior probability of different *support* classes by the following statistical model:

$$\hat{y}_j = \arg\max_{y \in \mathcal{Y}} p(y|x_j). \tag{2.5}$$

Some few shot learning approaches endeavor to model the posterior probability $p(y|x)$ directly by learning the distribution of x belonging to different classes in the task. However, some other methods prefer to tackle the problem by Bayes' rule, then the Bayesian model becomes to:

$$\hat{y}_j = \arg\max_{y \in \mathcal{Y}} p(x_j|y)p(y). \tag{2.6}$$

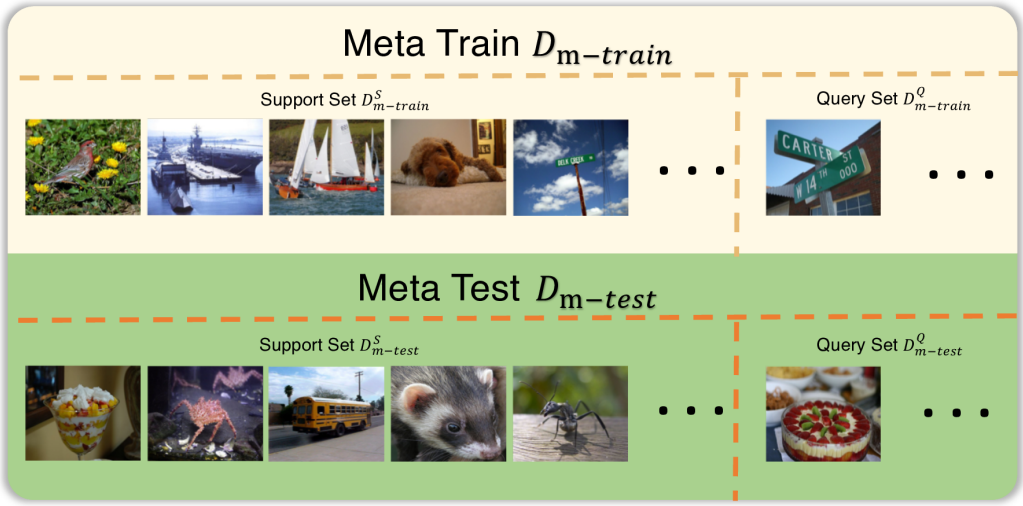


Figure 2.1 **Problem setup of few-shot learning.** The meta-test dataset has disjoint label space with the meta-train dataset, and they share the similar task generation as C-way-K-shot, which is 5-way-1-shot in this illustration.

where $p(y)$ is the prior distribution of the target class and $p(x_j|y)$ is the conditional distribution of the query instance given class y. However, $p(y)$ is assumed as uniformly distributed or calculated as the statistical frequency of different classes.

2.2.3 Benchmarks

Well-designed benchmarks motivate and spur the high development of machine learning algorithms. In terms of few-shot learning, we should design a benchmark where the learner is trained on a set of tasks, which is then required to generalize to learn on unseen tasks. Most few-shot learning studies follow the standard set-to-set task generation setting [170, 150], requiring the model should extract task-specific information from the *support set* and perform well on the *query set*. The meta-learner is trained across the tasks during meta-training process as "C-way-K-shot". There are numerous established few shot learning datasets, such as *omniglot* [170], *miniImagenet* [129], *tieredImagenet* [132], *CUB-200* [20] and *CIFAR-100* [80]. These datasets are all re-composed into smaller classification tasks with lower-way, and re-purposed to define a task definition for benchmarking the meta-training and meta-testing processes. All the dataset details are depicted in the following experiment sections.

While, it is convenient to generate enough tasks from aforementioned datasets, the model still suffers from the lack of task diversity. Even for a many-shot deep learning model, it is hard to fit a single neural network to tackle the domain-generalization scenarios. Specifically speaking, when it comes to a few-shot learner facing a novel domain without relevant available

auxiliary data, e.g., when source data is general images of *mini*ImageNet and target data is medical images from Chest X-Ray [75], the performance decreases dramatically with a domain shift, motivating some recent research on *Meta-Dataset* [164] and *cross-domain* evaluation [51]. This topic is naturally related to domain adaptation, but with the additional challenge that one is trying to learn a novel category in the target domain (unlike domain adaptation where target and source categories are usually overlapped).

2.3 Real-World Challenges

With the ubiquitous demand for machine learning systems, we have witnessed considerable progress in few shot learning, both in methodology and applications. However, challenges still exist due to the intrinsic difficulty of having sparse samples. Actually, most current few-shot learning studies are built on an ideal data hypothesis with small-scale novel few-shot task classes and large-scale labelled meta-training samples. These assumptions may not hold for some practical scenarios, such as the long-tailed phenomenon of data distribution which requires recognizing both many-shot and few-shot instances, or continual learning of incremental few-shot samples, and even open-set recognition with abnormal detection. This section extends the traditional problem setting to more challenging real-world complex scenarios and discusses the pros and cons of the existing related works.

2.3.1 Generalized Few-Shot Learning

Vanilla standard few-shot learning approaches are trained to make a prediction for pre-defined classes of a novel task, but are hardly applied to the previously seen classes in the meta-training auxiliary dataset. Naively extending them to recognize in the joint label space of auxiliary meta-training and target meta-testing data tends to result in catastrophic forgetting and/or poor scalability. However, it is very common that users are interested in recognising class concepts from both the auxiliary training dataset (with many examples available) and the novel classes (with few examples available). Such data imbalance is extremely challenging to handle. Then, generalized few-shot learning (GFSL) approaches are required to jointly recognize the many-shot and few-shot instances [147, 187].

Several augmentation based few-shot learning methods have two-stage independent learning process which firstly augments the few-shot training samples before meta-training the models, which is naturally suitable for generalized few-shot learning setting. *GcGPN* [147] combines the prototypical network [150] and *GCN* [77] to model the relationship of the few-shot new class with the many-shot old classes by defining the class as nodes and the inter-

class dependencies as edges. *CADA-VAE* [145] deploys a cross-modal embedding framework with variational auto-encoder, jointly representing the information in its latent space and training a linear softmax classifier. *FEAT* [188] develops the set-to-set class-agnostic function instantiating with transformer to realize embedding adaptation. Many other methodologies also tackle the generalized few-shot learning problems by leveraging the relationship between target task training instances.

2.3.2 Few-Shot Class-Incremental Learning

Incremental learning is commonly referred to *continual learning* and *lifelong learning*. It aims to continuously learn novel data in sequence (rather than in batch) while memorizing knowledge from previously seen tasks [130, 163]. The conventional few-shot learning paradigm can access all support examples during one single training session of the deep neural network. But practically speaking, as for the class-incremental learning scenario [166], the meta-learner should be trained in a sequence of delineated training sessions, with new classes presented at each session but the leaner should be capable of distinguishing between all classes [37]. Humans acquire and leverage knowledge from previous tasks, and integrate new knowledge with existing knowledge. This joint access not only overcomes the famous drawback of *catastrophic forgetting* but also utilizes past meta-knowledge. In a learning system, it is crucial and desirable to keep an incremental learning ability, not only due to the biological motivation, but also the potential to save on computing costs and optimize the learning process. In traditional deep learning systems, it is always essential to retrain all tasks in order to achieve better performance on both previous and novel tasks, but it is often expensive or even unavailable because of privacy or safety problems.

There are a variety of research in the literature regarding class-incremental learning problems. *iCARL* [130] firstly proposes a method for continual learning and learns strong nearest-neighbor classifiers and a data representation simultaneously, maintaining an episodic memory of previous exemplars. *EEIL* [18] utilizes external memory by adding the distillation loss term to the cross-entropy loss for end-to-end training. *NCM* [63] uses the cosine distance metric to mitigate the prediction bias in the output layer caused by the class-imbalance problem. Similarly, *BIC* [181] learns a bias-correction model to post-process the output logits to alleviate the bias between the progressively added new classes and old classes. However, in the context of few-shot learning, this thesis focuses on a more difficult scenario, few-shot class-incremental learning (FSCIL), where the number of new class training samples is limited. *TOPIC* [161] constrains the feature space of a convolutional neural network with a neural gas network, learning and preserving the topology of the features manifolded by different classes.

2.3.3 Open-Set Recognition

Traditional machine learning is based on a closed set or static environment assumption, where training and testing data are drawn from the same label and feature space. However, various realistic scenarios are usually open and non-stationary where some unseen instances would appear unexpectedly, such as unmanned driving [17, 95], medical diagnosis [52, 73], and etc. *Open-set recognition* (OSR), first formulated by Scheirer et al. [142], which aims to endow a learning system with the ability to reject unknown samples from novel classes at test time, while preserving the ability to recognize known classes. Nevertheless, it is difficult for traditional machine learning algorithms to work effectively when the open-set samples may have a high activation score for one of the known categories. Therefore, various frameworks and algorithms of open-set recognition in the computer vision area have attracted considerable attention recently [123, 58], and we analyze this challenge in the context of deep networks.

Scheirer et al. [142] proposed an extreme value parameter redistribution method for the logits generated by the classifier. The later works could be divided into discriminative and generative perspectives. *Schlachter et al.* [143] built an intra-class splitting model, where a closed-set classifier was used to split data into typical and atypical subsets, reformulating open-set recognition as a traditional classification problem. *OpenMax* [12] has been proposed as the first solution for an open-set deep neural network with the normal SoftMax layer. *G-OpenMax* [43] utilizes a generator as an alternative to SoftMax, providing explicit probability estimation over the generated unknown classes to synthesize all unknown examples as an extra class. *Neal et al.* [113] introduced counterfactual image generation, aiming to generate samples that cannot be classified into any of the seen classes, producing an extra class for classifier training. To deviate from simulating open-set classes, a class conditional generator learns a representation preserving only known-class samples [117]. It is also possible to ignore the labels and modelling known class with one-class classification [118, 124, 123], but this tends to perform poorly compared to standard approaches. Generative representation and self-supervision can also be used to enhance the performance [125].

Most methods reduce the set of unseen classes to one extra class. In actual fact, open samples can be drawn from different categories and have significant visual differences, thus the assumption that a feature extractor can map them all into a single feature space cluster is biased. It is practically difficult even if theoretically possible. Some generative approaches try to build a meta learning model allowing a cluster per seen class and label samples that do not fall into these clusters as unseen.

Few-Shot Open-Set Recognition However, few-shot recognition is a more challenging scenario than the context of large-scale classification. Obviously, the approaches based on the large-scale classifier are not suitable for solving few-shot open-set recognition (FSOSR)

problems. A classifier trained in the few-shot context is more difficult to delineate seen-class boundariessolve due to the lack of labelled data. Therefore, few-shot is a more pervasive but challenging setup for current open-set recognition research. Meta learning methods tend to capture the metric structure of the data more broadly throughout the feature space and achieve better performance on these generalization critical tasks. An episodic training mechanism randomly selects a set of novel classes per episode, producing more robust embedding to overcome the problem of overfitting to one specific class. *PEELER* [93] proposes a meta-learning-based framework for open-set recognition, combining the cross-entropy loss and a novel open-set loss to improve open-set performance on both the large-scale and few-shot settings. Gaussian embedding and Mahalanobis distance further improve the performance and robustness.

2.4 Summary

In this chapter, we have reviewed the background and several well-established algorithms of meta learning, providing a formalization and taxonomy analysis. We then discussed the significance and contributions of meta-learning to the few-shot learning problem, and also introduced the benchmarks and terminologies. Finally, we moved to more complex real-world scenarios, and provided a literature review in this cutting-edge research field.

Chapter 3

RelationNet2

In this chapter, we discuss few-shot deep learning which scales visual recognition to open ended growth of unseen new classes with limited labeled figures. One promising approach is based on metric learning, which trains a deep embedding to support image similarity matching. Our insight is that effective general-purpose matching requires discrimination with regard to features at multiple abstraction levels. We propose RelationNet2, a.k.a Deep Comparison Network (DCN) in the published paper, decomposing embedding learning into a sequence of modules paired each with a relation module. Furthermore, to reduce overfitting and enable deeper embeddings, we represent images as distribution rather than vectors via learning parameterized Gaussian noise regularization. Finally, the resulting network achieves state-of-the-art performance on both *mini*ImageNet and *tiered*ImageNet, retaining the appealing simplicity and efficiency of deep metric learning approaches.

3.1 Introduction

Few shot learning recovers a surgence recently due to the successful development of deep learning models [56, 69, 66] applied on large-scale visual recognition problems. However, the most popular deep-learning based methods treat each learning problem independently from scratch, and fail to learn efficiently from few instances while human could easily generate a new concept from a single image by building upon prior knowledge. These observations have motivated a resurgence of interest in FSL (few-shot learning) for visual recognition [170, 34, 150, 127] and beyond. Contemporary deep networks overfit in the few-shot regime – even when exploiting fine-tuning [193], data augmentation [81], or regularization [152] techniques. In contrast, 'Meta-learning' techniques extract transferable task agnostic knowledge from historical tasks and benefit sparse data learning of specific new target tasks. These take several forms: Fast adaptation methods enable sparse-data adaptation without overfitting –

via good initial conditions [34] or learned optimizers [129]. Weight synthesis approaches learn a meta-network that synthesizes recognition weights given a training set [13, 106]. Deep metric learning approaches support representation [78] and comparison [170, 150] of instances, allowing new categories to be recognized with nearest-neighbour comparison. However, existing approaches have several drawbacks including inference complexity [85, 84], architectural complexity [108], the need to fine-tune on the target problem [34], or reliance on a simple linear comparison [170, 150, 84].

We build on deep metric learning methods due to their architectural simplicity and instantaneous training of new categories. These methods use auxiliary training tasks to learn a deep image-embedding such that the embedded data becomes linearly separable [78, 170, 150]. Thus the decision is non-linear in image-space, but linear in the embedding space. For learning the target task, images are simply memorized during few-shot training. But for testing the target task, query images are matched to training examples by deep embedding and similarity comparison function. Within this paradigm, the recent Relation Network [159] achieved excellent performance by learning a non-linear comparison function. Relation Network is meta-trained to lean a deep distance metric to compare a small number of images within episodes, which is designed to simulate the few-shot setting. The model is trained end-to-end from scratch and is able to classify new images from novel classes by only computing relation scores without further updating.

Learning the embedding and non-linear comparison module jointly alleviates the reliance on the embedding's ability to generate linearly separable features. We take this idea of jointly learning an embedding and a non-linear distance metric further with the following insights. First, we introduce the notion of multiple meta-learners operating at multiple abstraction levels. Concretely we train non-linear distance metrics corresponding to each embedding module in a feature hierarchy - thus covering features from simple textures to complex parts [194]. Secondly, prior studies only use a single linear [150] or non-linear comparison [159]. To provide the inductive bias that each layer of representation should be potentially discriminative for matching, and enable better gradient propagation [69] to each relation module, we deeply supervise [87] all the relation modules. Finally, to enable deeper embedding architectures to be used without overfitting, we design each embedding module to output a feature *distribution*, thus representing each image as a distribution rather than a vector. This can be seen as an end-to-end learnable noise regularizer that performs data augmentation in semantic feature space rather than image space.

Overall our RelationNet2 (RN2) can be seen as jointly learning embedding and comparison as task agnostic meta knowledge [78, 170, 150, 159]. It makes full use of deep networks by making comparisons with the full feature hierarchy extracted by the embedding network,

and learning Gaussian noise to improve generalization. The resulting framework maintains the architecture simplicity and efficiency of other methods in this line, while providing excellent performance on both *mini*ImageNet and the more challenging *tiered*ImageNet few shot learning benchmarks.

3.2 Related Work

Contemporary approaches to deep-network few-shot learning have exploited the learning-to-learn paradigm. Auxiliary tasks are used to meta-learn some task agnostic knowledge, before exploiting this to learn the target few-sample more effectively problem. The learning-to-learn idea has a long history [162, 33, 85], but contemporary approaches typically cluster into three categories: Fast adaptation, weight synthesis, and metric-learning approaches.

3.2.1 Fast Adaptation

These approaches aim to meta-learn an optimisation process that enables base models to be fine-tuned quickly and robustly. So that a base model can be updated for sparse data target problems without extensive overfitting. Effective ideas include the simply meta-learning an effective initial condition [34], and learning a recurrent neural network optimizer to replace the standard SGD learning approach [129]. Recent extensions also include learning per-parameter learning rates [92], and accelerating fine-tuning through solving some layers in closed form [14]. Nevertheless, these methods suffer from needing to be fine-tuned for the target problem, often generating costly higher-order gradients during meta-learning process [34], and failing to scale to deeper network architectures as shown in [106]. They also suffer from a fixed parametric architecture. For example, once the MAML [34] is trained for 5-way auxiliary classification problems, it is restricted to the same for target problems without being straightforwardly generalizable to a different cardinality of classification.

3.2.2 Classifier Synthesis

Another line of work focuses on synthesising a classifier based on the provided few-shot training data [44]. An early method in this line learned a transferrable 'LearnNet' that generated convolutional weights for the base recognition network given a one-shot training example [13]. However, this was limited to binary classification. Conditional Neural Processes [41] exploited a similar idea, but in a Bayesian framework. SNAIL obtained excellent results by embedding the training set with temporal convolutions and attention [106]. Recently Qiao *et al*. proposed a method to predict classification parameters given neuron activations

[127]. In this case the global parameter prediction network is the task agnostic knowledge that is transferred from auxiliary categories. Compared to the fast adaptation approaches, these methods generally synthesize their classifier in a single pass, making them faster to train on the target problem. However learning to synthesize a full classifier does entail some complexity. This process can overfit and generalize poorly to novel target problem.

3.2.3 Deep Metric Learning

These approaches aim to learn a deep embedding that extracts robust features, allowing them to be classified directly with nearest neighbour type strategies in the embedding space. The deep embedding forms the task agnostic knowledge transferred from auxiliary to target tasks. Early work simply used *Siamese Networks* [78] to embed images, such that images of the same class are placed near each other. Matching Networks [170] defined a differentiable nearest-neighbour loss based on cosine similarity between the support set and query embedding. *Prototypical Networks* [150] provided a simpler but more effective variant of this idea where the support set instances for one class are embedded as a single prototype. Their analysis showed that this leads to a linear classifier in the embedding space. The most related method to ours is *RelationNet* [159], which extends this line of work to use a separate non-linear comparison module instead of relying entirely on the embedding networks to make the data linearly separable [78, 150, 170]. This division of labour between a deep embedding and a deep relation module improved performance in practice [159]. Our approach builds on this line of work in general and RelationNet in particular. RelationNet relied on the embedding networks to produce a *single* embedding for the relation module to compare. We argue that a general purpose comparison function should use any or all of the full feature hierarchy [194] to make matching decisions, for example matching based on colors, textures, or parts, which may be represented at different layers in an embedding network. To this end we modularise the embedding networks, and pair every embedding module with its own relation module.

3.2.4 Use of Feature Hierarchies

The general strategy of simultaneously exploiting multiple layers of a feature hierarchy has been exploited in conventional many-shot classification network [69, 153], instance recognition [19], and semantic segmentation networks [54]. However, in the context of deep-metric learning, the conventional pipeline is to extract a complete feature [42, 67]. Importantly, in contrast to prior approaches single ‘short-cut’ connection of deeper features to a classifier [54, 19], we uniquely learn a hierarchy of relation modules: One non-linear comparison function for each block of the embedding modules. Our approach is also reminiscent of

classic techniques such as spatial pyramids [86] (since each module in the hierarchy operates at different spatial resolutions) and multi-kernel learning [169] (since we learn multiple relation modules for each feature in the hierarchy). This can also be seen as the first multiple meta-learner approach for few shot learning problems.

3.2.5 Leaned Noise and Regularization

Many previous FSL models struggle with deeper backbones [106, 34]. For best performance, we would like to exploit a state-of-the-art embedding module architecture (we use SENet [66]), and also benefit from the array of comparison modules mentioned above. To enable RN2 to benefit from deep backbones without overfitting, we modify the embedding modules to output a feature distribution at each layer. Rather than generating deterministic features at a module output, we generate means and variances which are sampled in the forward pass, with back propagation relying on the reparamaterization trick. Unlike density networks [15] where such distributions are only generated at the output layer, or VAEs [76] here they are generated only once by the generator, we generate such stochastic features at *each* embedding module's output. This can be seen as an end-to-end learnable data augmentation strategy in semantic feature rather than image space. It is also complementary to standard L2/weight decay and image space augmentation techniques.

3.3 Methodology

3.3.1 Problem Definition

We consider a C-way K-shot classification problem for few shot learning. There are some labelled source tasks with sufficient data, denoted meta-train $\mathcal{D}_{\text{m-train}}$, and we ultimately want to solve a new set of target tasks denoted meta-test $\mathcal{D}_{\text{m-test}}$, for which the label space is disjoint. Within meta-train and meta-test, we denote each task as being composed of a support set of training examples, and a query set of testing examples. The meta-test tasks are assumed to be few-shot, so $\mathcal{D}_{\text{m-test}}$ contains a support set with C categories and K examples each. We want to learn a model on meta-train that can generalize out of the box, without fine-tuning, to learning the new categories in meta-test.

Episodic Training We adopt an episodic training paradigm for few-shot meta-learning. During meta-training, an episode is formed as follows: (i) Randomly select C classes from $\mathcal{D}_{\text{m-train}}$, (ii) Sample K images each class, which serve as *support set* $\mathcal{D}^{\text{S}}_{\text{m-train}} = \left\{(x_i, y_i)\right\}_{i=1}^{m}$, where $m = K * C$, (iii) For the same C classes, sample K' images each class serving as

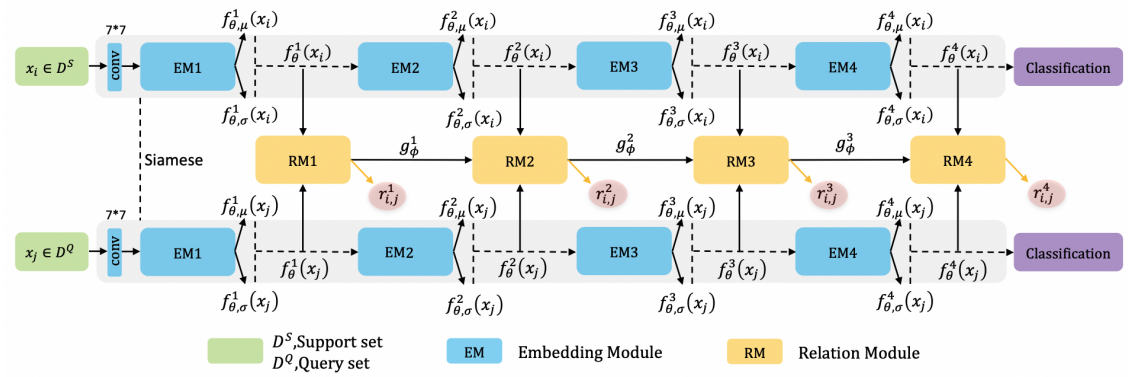


Figure 3.1 **Network architecture of RelationNet2.** There are 4 embedding modules f_θ for each embedding branch, and a set of 4 corresponding relation modules g_ϕ. Support set and query set share the same embedding network. Each embedding module outputs a feature distribution $\mathcal{N}(f_{\theta,\mu}(x), f_{\theta,\sigma}(x))$, we then randomly sample a feature $f_\theta(x)$ as the input of corresponding relation module and next embedding module.

the *query set* $D^{\mathrm{Q}}_{\text{m-train}} = \left\{(\tilde{x}_j, \tilde{y}_j)\right\}_{j=1}^{n}$, where $n = K' * C$, $D^{\mathrm{S}}_{\text{m-train}} \cap D^{\mathrm{Q}}_{\text{m-train}} = \emptyset$. The support/query distinction mimics the $D_{\text{m-test}}$/ real-time testing. Our few-shot RN2 will be trained for instance comparison using episodes constructed in this manner.

3.3.2 Model

Overview RelationNet2 (RN2) is composed of two module types: *embedding* and *relation* modules f_θ and g_ϕ, as shown in Figure 3.1. The detailed architecture will be given in Section 3.3.3. Here we choose 4 sub-modules following that SENet architecture has 4 blocks. A pair of images x_i and x_j in the support and query set are fed to embedding modules respectively. Then the multi-level embedding modules output stochastic features to the corresponding multi-level relation modules, and learn the relation score and weights for different relation modules. Finally, the RN2 learns weighted non-linear metric of few shot learning tasks.

Distribution Embedding Modules Conventionally, an embedding module (e.g., a ResNet or SENet block) outputs deterministic features. As a regularization strategy, we treat each feature output as a random variable drawn from a parameterized Gaussian distribution, for which the embedding module outputs the mean and variance. This design is illustrated in Figure 3.1. Each vth-level embedding module predicts a feature mean $f^v_{\theta,\mu}$ and a feature variance $f^v_{\theta,\sigma}$. To generate a module's output f^v_θ, we use the reparameterization trick to draw

one (or more) Gaussian random samples

$$f_\theta^\upsilon = f_{\theta,\mu}^\upsilon + \varepsilon \odot f_{\theta,\sigma}^\upsilon, \tag{3.1}$$

where ε is a standard Gaussian $\mathcal{N}(0,1)$ random samples, and $\odot$ denotes element-wise product.

Metric Hierarchy The υth-level of embedding modules produce query and support image feature maps, which are concatenated as $[f_\theta^\upsilon(x_i), f_\theta^\upsilon(x_j)]$, and then fed into the corresponding υth-level relation module for comparison. For a pair x_i and x_j at level $\upsilon - 1$, the relation module outputs a similarity feature map $g_\phi^{\upsilon-1}$. The υth-level relation module takes both the υth-level embedding output for query and support, and also the $(\upsilon-1)$th-level relation module similarity feature map as input:

$$g_\phi^\upsilon = g([f_\theta^\upsilon(x_i), f_\theta^\upsilon(x_j), g_\phi^{\upsilon-1}]). \tag{3.2}$$

The first relation module is special as it does not have a predecessor to input, and we cannot use zero-padding because 0 has a specific meaning in our context (that the similarity of the previous support and query images are the same). Thus we set $g_\phi^1 = g([f_\theta^1(x_i), f_\theta^1(x_j)])$.

Simultaneously, after an average pooling and fully connected layer denoted $q(\cdot)$, each relation module also outputs a real-valued scalar representing similarity/relation score $r_{i,j}^\upsilon$ of two images estimated at the feature level υ,

$$r_{ij}^\upsilon = q(g_\phi^\upsilon). \tag{3.3}$$

K-Shot For K-shot with $K > 1$, the embedding module outputs the average pooling of features, and all samples from the same class produce *one* feature map. Thus, the number of outputs for the υ-level relation module is C, regardless of the value of K.

Objective Function There are 2 steps to train the RelationNet2 (RN2) network. We first train the embedding network, then fix the embedding network parameters and train the relation network (run the whole RN2 consisting of embedding and relation modules, but only update the relation modules). We first train the embedding network θ as a conventional multi-class classifier for the data in $\mathcal{D}_{m-train}$ using cross entropy loss ℓ^{CE}. To leverage our distribution-embedding, we add a feature variance regularizer:

$$\theta \leftarrow \underset{\theta}{\operatorname{argmin}}\ \ell^{CE}(\theta) - \lambda \frac{1}{m} \sum_{i=1}^{m} \sigma_i, \tag{3.4}$$

where σ_i is the predicted standard deviation of each instance and m is their total number, and λ is the hyperparameter to finetune the influence of the regularizer (here is 0.01). This ensures

that feature distributions are learned, and we do not collapse to standard (zero-variance) vector embedding (our mean σ is about 0.5). This pipeline can be seen as a learnable data augmentation strategy at each level of the feature hierarchy for relation modules. Learning with these augmented features improves generalization. After embedding training, the parameters θ of embedding modules are fixed.

We next train the column of relation modules ϕ on $\mathcal{D}_{m-train}$ with an episodic strategy [170] using cross entropy loss ℓ^{CE} at each module (Figure 3.1). To weight the V relation modules, we assign a learnable attention weight $w^{v}_{c,j}$ to the calculated relation similarity score $r^{v}_{c,j}$ of each module.

$$\phi \leftarrow \underset{\phi}{\operatorname{argmin}} \sum_{c=1}^{C} \sum_{j=1}^{n} \sum_{v=1}^{V} \ell^{CE}(w^{v}_{c,j} r^{v}_{c,j}, \mathbf{1}(y_c = y_j); \phi), \tag{3.5}$$

where $j = 1 \ldots n$ refers to query samples and c refers to a batch of K support examples of class y_c in a C-way-K-shot problem. $r_{c,j}$ are the relation scores between query image j and the class y_c support images. Additionally, $w^{v}_{c,j} = \alpha^{v}(g^{v}_{c,j})$ is a sigmoid-activated fully connected layer that computes a scalar attention weight given relation feature map $g^{v}_{c,j}$, and the weights of α^{v} are included in ϕ.

Testing Strategy To evaluate our learned model on C-way-K-shot learning, we calculate the final relation score $r_{c,j}$ of one query image x_j to the images of each support class c:

$$r_{c,j} = \sum_{v=1}^{V} w^{v}_{j} r^{v}_{c,j} \tag{3.6}$$

where $r^{v}_{c,j}$ is the relation score between image j and the support images of class c at module v. Finally, the class with the highest relation score r_c is the final predicted classification. We evaluate the approach by the resulting classification accuracy.

3.3.3 Network Architecture

The RelationNet2 architecture (Figure 3.1) uses 4 embedding modules, each paired with a relation module. We explain our method with SENet for concreteness, but it can be instantiated with any backbone.

Embedding Subnetwork As shown in Table 3.1, first we use a 7×7 convolution followed by a 3×3 max-pooling, which is a common size reduction as [66]. Then, we have 4 embedding modules each composed of a number of SENet blocks. Finally, an avg-pooling and a fully-connected layer are used to produce C logit values, corresponding to C classes in $\mathcal{D}_{\text{m-train}}$. More specifically, 4 embedding modules followed the 4 SENet basic blocks composition

Output size	**Embedding**	**Embedding+** *noise*	Output size	**Relation**
112×112	conv, 7×7, 64, stride 2, padding 3			
56×56	Maxpooling 3×3, stride 2, padding 1			
56×56	$\begin{bmatrix} conv,3\times3,64 \\ conv,3\times3,64 \\ fc,[4,64] \end{bmatrix}\times3$	$\begin{bmatrix} conv,3\times3,64 \\ conv,3\times3,64 \\ fc,[4,64] \end{bmatrix}\times2$ $\begin{bmatrix} conv,3\times3,64 \\ conv,3\times3,65 \\ fc,[4,65] \end{bmatrix}\times1$	28×28	$\begin{bmatrix} conv,3\times3,128 \\ conv,3\times3,128 \\ fc,[8,128] \end{bmatrix}\times2$
28×28	$\begin{bmatrix} conv,3\times3,128 \\ conv,3\times3,128 \\ fc,[8,128] \end{bmatrix}\times4$	$\begin{bmatrix} conv,3\times3,128 \\ conv,3\times3,128 \\ fc,[8,128] \end{bmatrix}\times3$ $\begin{bmatrix} conv,3\times3,128 \\ conv,3\times3,129 \\ fc,[8,129] \end{bmatrix}\times1$	14×14	$\begin{bmatrix} conv,3\times3,384 \\ conv,3\times3,256 \\ fc,[16,256] \end{bmatrix}\times1$ $\begin{bmatrix} conv,3\times3,256 \\ conv,3\times3,256 \\ fc,[16,256] \end{bmatrix}\times1$
14×14	$\begin{bmatrix} conv,3\times3,256 \\ conv,3\times3,256 \\ fc,[16,256] \end{bmatrix}\times6$	$\begin{bmatrix} conv,3\times3,256 \\ conv,3\times3,256 \\ fc,[16,256] \end{bmatrix}\times5$ $\begin{bmatrix} conv,3\times3,256 \\ conv,3\times3,257 \\ fc,[16,257] \end{bmatrix}\times1$	7×7	$\begin{bmatrix} conv,3\times3,768 \\ conv,3\times3,512 \\ fc,[32,512] \end{bmatrix}\times1$ $\begin{bmatrix} conv,3\times3,512 \\ conv,3\times3,512 \\ fc,[32,512] \end{bmatrix}\times1$
7×7	$\begin{bmatrix} conv,3\times3,512 \\ conv,3\times3,512 \\ fc,[32,512] \end{bmatrix}\times3$	$\begin{bmatrix} conv,3\times3,512 \\ conv,3\times3,512 \\ fc,[32,512] \end{bmatrix}\times2$ $\begin{bmatrix} conv,3\times3,512 \\ conv,3\times3,513 \\ fc,[32,513] \end{bmatrix}\times1$	7×7	$\begin{bmatrix} conv,3\times3,1536 \\ conv,3\times3,512 \\ fc,[32,512] \end{bmatrix}\times1$ $\begin{bmatrix} conv,3\times3,512 \\ conv,3\times3,512 \\ fc,[32,512] \end{bmatrix}\times1$
1×1	Global average pooling, fc			

Table 3.1 **Parameters of each embedding and relation module.** Relation modules concatenate the final feature maps of both corresponding embedding modules and the previous relation module. The output size of each embedding module matches the input size of the corresponding relation module. The brackets of '*fc*' indicate the dimension of FC layers in an SE block [66].

[3,4,6,3], respectively. In original SENet paper [66], they use SE-ResNet-50, but here we use smaller backbones as SE-ResNet-34, where $(3+4+6+3)*2+2=34$. Otherwise, we follow the other setting in [66], e.g., reduction ratio $r=16$ as suggested.

Distribution Embedding Conventually, an embedding module outputs deterministic features. As explained in Section 3.3.2, each embedding module's output is split into two parts: the mean feature $f_{\theta,\mu}$ sized $[b,c,h,w]$ ([batch_size, channel, height, width]), and standard deviation (std) $f_{\theta,\sigma}$ sized $[b,1,h,w]$. We assume that every channel shares the same standard deviation (std). This means, in addition to the penultimate-to-output layer (now it is penultimate-to-mean layer), we have a new penultimate-to-std layer (with its own parameters). The motivation behind sharing stds across channels is to reduce the number of parameters in

the newly introduced layer. We also control the amount of noise added by applying Sigmoid activation to constrain the std to the range $[0, 1]$. We sample one feature vector per image in a single forward pass, but multiple samples are drawn considering the whole batch.

Relation Subnetwork As illustrated in Figure 3.1, the relation column consists of 4 serial modules, each of which has 2 SENet blocks, with a pooling and a fully-connected layer to produce the relation score. Thus the relation modules are designed as [2,2,2,2], where the SENet block architecture is the same as the one used in embedding module.

3.4 Experiments

Our RN2 is evaluated on few-shot classification using datasets: *mini*ImageNet and *tiered*ImageNet datasets. All experiments are implemented in PyTorch. Code is published on https://github.com/zhangxueting/DCN.

3.4.1 Prerequisites

Baselines We compare several state-of-the-art baselines for few-shot learning including Matching Nets [170], Meta Nets [108], Meta LSTM [129], MAML [34], Baseline++ [20], Prototypical Nets [150], Graph Neural Nets [40], Meta-SSL [132], Relation Net [159], Meta-SGD [92], TPN [96], CAVIA [198], DynamicFSL [44], SNAIL [106], AdaResNet [109], TADAM [116], MTL [156], TapNet [191], MetaOpt Net [84], PPA [127], LEO [138].

Data Augmentation We follow the standard data augmentation [160, 66, 56, 20] with random-size cropping and random horizontal flipping Input images are normalized through mean channel subtraction.

Pre-train and train The embedding branch is pre-trained by the training set and the parameters then are fixed. The validation set is used to estimate the number of early stop episodes for the relation training. Finally, both train and validation data (as per common practice [127]) are used to train the relation modules in RN2.

3.4.2 *mini*Imagenet

Dataset *mini*ImageNet has 60,000 images in consist of 100 ImageNet classes, each with 600 images [170]. Following the split in [129], the dataset is divided into a 64-class training set, 16-class validation set and a 20-class testing set.

Settings We evaluate both *5-way-1-shot* and *5-way-5-shot*, where each episode contains 5 query images for each sampled class. There are 5*5+1*5=30 images per training

Model	Embedding	*mini*Imagenet 5-way Acc.	
		1-shot	5-shot
MATCHING NETS [170]	Conv-4	43.56 ± 0.84%	55.31 ± 0.73%
META LSTM [129]	Conv-4	43.44 ± 0.77%	60.60 ± 0.71%
MAML^O [34]	Conv-4	48.70 ± 1.84%	63.11 ± 0.92%
BASELINE++ [20]	Conv-4	48.24 ± 0.75%	66.43 ± 0.63%
META NETS [108]	Conv-5	49.21 ± 0.96%	-
PROTONET [150]	Conv-4	49.42 ± 0.78%	68.20 ± 0.66%
GNN [40]	Conv-4	50.33 ± 0.36%	66.41 ± 0.63%
META SSL[132]	Conv-4	50.41 ± 0.31%	64.39 ± 0.24%
RELATION NET [159]	Conv-4	50.44 ± 0.82%	65.32 ± 0.70%
META SGD^O [92]	Conv-4	50.47 ± 1.87%	64.03 ± 0.94%
TPN [96]	Conv-4	52.78 ± 0.27%	66.59 ± 0.28%
CAVIA [198]	Conv-4	51.82 ± 0.65%	65.85 ± 0.55%
DYNAMIC FSL† [44]	Conv-4	**56.20 ± 0.86%**	**72.81 ± 0.62%**
RN2	Conv-4	**53.48 ± 0.78%**	**67.63 ± 0.59%**
BASELINE++ [20]	ResNet-18	51.87 ± 0.77%	75.68 ± 0.63%
RELATION NET [20]	ResNet-18	52.48 ± 0.86%	69.83 ± 0.68%
PROTONET [20]	ResNet-18	54.16 ± 0.82%	73.68 ± 0.65%
SNAIL [140]	ResNet-12	55.71 ± 0.99%	68.88 ± 0.92%
DYNAMIC FSL [44]	ResNet-12	55.45 ± 0.89%	70.13 ± 0.68%
ADARESNET [109]	ResNet-12	57.10 ± 0.70%	70.04 ± 0.63%
TADAM [116]	ResNet-12	58.50 ± 0.30%	76.70 ± 0.30%
MTL [156]	ResNet-12*	61.20 ± 1.80%	75.50 ± 0.80%
TAP NET [191]	ResNet-12	61.65 ± 0.15%	76.36 ± 0.10%
METAOPTNET^O [84]	ResNet-12*	**64.09 ± 0.62%**	**80.00 ± 0.45%**
RN2	ResNet-12	**63.92 ± 0.98%**	**77.15 ± 0.59%**
PPA [127]	WRN-28-10	59.60 ± 0.41%	73.74 ± 0.19%
LEO^O [138]	WRN-28-10	61.78 ± 0.05%	77.59 ± 0.12%
MAML	SENet	55.99 ± 0.99%	-
RELATION NET	SENet	57.39 ± 0.86%	-
PROTONET	SENet	51.60 ± 0.85%	-
RN2	SENet	**63.19 ± 0.87%**	**76.58 ± 0.66%**

Table 3.2 **Few-shot classification results on *mini*ImageNet.** Our model achieves excellent performance across a range of shallow and deep architectures. All accuracies are averaged over 600 test episodes and are reported with 95% confidence intervals. From top to bottom: Simple conv block embeddings to other deep embeddings (ResNet, WRN, SENet). '-': not reported. †: uses two-step optimization with added attention. ^{O}: requires gradient-based optimisation at meta-test time. *: uses a wider ResNet than standard and higher dimensional embedding.

episode/mini-batch for 5-way-1-shot experiments, and 5*5+5*5=50 images for 5-way-5-shot experiments. When it comes to 5-shot, we calculate the class-wise average feature across the support set. Thus we get 5*5*5*1=125 feature pairs as input for the relation module. For

embedding and relation module training, optimization uses SGD with momentum 0.9. The initial learning rate is 0.1, decreased by a factor of 5 every 60 epochs, and the training epoch is 200. All models are trained from scratch, using the robust RELU weight initialization [57]. We follow [20] in using 224×224 pixels crops for evaluation on ResNet and SENet, and [129] in using 84×84 images for the smaller Conv-4 backbone.

Results Following the setting of [150], when evaluating testing performance, we batch 15 query images per class in a testing episode and the accuracy is calculated by averaging over 600 randomly generated testing tasks (for both 1-shot and 5-shot scenarios). In Table 3.2, RN2 achieves excellent performance with different embedding backbones. Specifically, the accuracy of 5-way *mini*ImageNet with SENet is 63.19% and 76.58% for 1-shot and 5-shot respectively. We note that MetaOptNet [84] uses significantly more advanced regularizers than standard among the competitors (which corresponds to about 2% performance according to [84]), also requires an order of magnitude higher dimensionality of embeddings [64,160,320,640] than the other competitors [64,96,128,256]. Overall RN2's 1-shot recognition performance is state-of-the-art among methods that do not require optimisation at meta-test time (unlike, e.g., MAML [34] and MetaOptNet [84]). It is noteworthy that achieving good performance with deeper backbones is not trivially automatic as Dynamic FSL, for example fails to improve from Conv-4 to ResNet embedding. RN2's learned noise regularizer helps it to exploit a powerful SENet backbone without overfitting. Direct comparison among models is complicated by the diversity of embedding networks used in different studies, so we show the results of RN2 with each commonly used backbone in Table 3.2, e.g. Conv-4 and ResNet-12. We can see that our model performs favorably across a range of architectures.

Cross-way Testing Results Standard procedure in few-shot evaluation is to train models for the desired number of categories to discriminate at testing time. However, unlike alternatives such as MAML [34], our method is not required to match label cardinality between training and testing. We therefore evaluate 5-way trained model on 20-way testing in Table 3.3. It shows that our model outperforms the alternatives clearly despite RN2 being trained for 5-way, and the others specifically for 20-way, indicating another important aspect of RN2's flexibility and general applicability.

3.4.3 *tiered*Imagenet

Dataset *tiered*ImageNet is a larger few-shot recognition benchmark containing 608 classes (779,165 images), in which training/validation/testing categories are organized so as to ensure a larger semantic gap than those in *mini*ImageNet, thus providing a more rigorous test of generalization. This is achieved by dividing according to 34-higher-level nodes in the

Model	**Embedding**	***mini*Imagenet 20-way Acc.**	
		1-shot	5-shot
MATCHING NETS [92]	Conv-4	$17.31 \pm 0.22\%$	$22.69 \pm 0.86\%$
META LSTM^O [92]	Conv-4	$16.70 \pm 0.23\%$	$26.06 \pm 0.25\%$
MAML^O [92]	Conv-4	$16.49 \pm 0.58\%$	$19.29 \pm 0.29\%$
META SGD^O [92]	Conv-4	$17.56 \pm 0.64\%$	$28.92 \pm 0.35\%$
RN2	Conv-4	$27.56 \pm 0.24\%$	$39.56 \pm 0.81\%$
RN2	ResNet-12	$31.65 \pm 0.34\%$	$50.25 \pm 0.46\%$
RN2	SENet	$\mathbf{32.90 \pm 0.39\%}$	$\mathbf{51.37 \pm 0.39\%}$

Table 3.3 **20-way classification accuracy on *mini*ImageNet.** RN2 is trained on 5-way with different embeddings and transferred to 20-way. The results of Meta LSTM, MAML and Meta SGD are from [92].

ImageNet hierarchy [132], grouped into 20 for training (351 classes), 6 for validation (97 classes) and 8 for testing (160 classes), respectively.

Settings Similar to the setting of *mini*ImageNet, we use 5 query images per training episode. Due to the larger data size, we train embedding modules with a larger batch size 512, initial learning rate 0.3 and 100 training epochs. Other settings remain the same as *mini*ImageNet.

Results Following the former experiments, we batch 15 query images per class in each testing episode and the accuracy is calculated by averaging over 600 randomly generated testing tasks. From Table 3.4, RN2 achieves the state-of-the-art performance on the 5-way-1-shot and 5-shot tasks with comfortable margins. Again, this is state-of-the-art performance for methods that do not require optimisation at meta-testing. We note also that Meta-SSL [132] and TPN [96] are semi-supervised methods that use more information than ours, and have additional requirements such as access to the test set for transduction.

3.5 Further Analysis

In this section, we capture some further analysis to highlight the insight of our model, show the comparison of different metric learners, and give ablation study to confirm the effectiveness of the architecture design and multiple relation modules.

3.5.1 Application to Other Metric Learners

Our main insight is the value of feature comparison at multiple abstraction levels in metric learning, as well as that of learned noise regularizers for deep networks in the few-shot regime.

Model	Embedding	*tiered*Imagenet 5-way Acc.	
		1-shot	5-shot
REPTILE [96]	Conv-4	48.97%	66.47%
MAML [96]	Conv-4	51.67%	70.30%
META SSL† [132]	Conv-4	52.39 ± 0.44%	70.25 ± 0.31%
PROTO NET [96]	Conv-4	53.31%	72.69%
RELATION NET [96]	Conv-4	54.48%	71.31%
TPN† [96]	Conv-4	59.91%	73.30%
TAP NET [191]	ResNet-12	63.08 ± 0.15%	80.26 ± 0.12%
METAOPTNET^{O} [84]	ResNet-12*	**65.81 ± 0.74%**	**81.75 ± 0.53%**
RN2	Conv-4	60.58 ± 0.72%	72.42 ± 0.69 %
RN2	ResNet-12	68.58 ± 0.63%	**80.65 ± 0.91%**
RN2	SENet	**68.83 ± 0.94%**	79.62 ± 0.77%

Table 3.4 **Few-shot classification results on *tiered*ImageNet.** All accuracies are averaged over 600 test episodes and reported with 95% confidence intervals. For each task, the best-performing method is bold. †: uses additional unlabelled data for semi-supervised learning or transductive inference. ^{O}: requires gradient-based optimisation at meta-test time. *: uses a wider ResNet than standard size and higher dimensional embedding.

Model	Noise Regularization?	Deep Comparisons?	Acc.
PROTONET[150]	X	X - 1 module	51.04 ± 0.77%
PROTONET	✓	X - 1 module	51.60 ± 0.85%
PROTONET	X	✓- 4 modules	53.62 ± 0.82%
PROTONET	✓	✓- 4 modules	**54.78 ± 0.88%**
RELATIONNET[159]	X	X - 1 module	52.48 ± 0.86%
RELATIONNET	✓	X - 1 module	57.39 ± 0.86%
RN2	X	✓- 4 modules	60.57 ± 0.86%
RN2	✓	✓- 4 modules	**63.19 ± 0.87%**

Table 3.5 **Comparison of RelationNet and ProtoNet.** Multiple deep comparisons and distribution embedding of features benefit both RelationNet (learnable relation modules) and ProtoNet (fixed linear modules) few-shot architectures. Accuracies are calculated on 5-way-1-shot classification of *mini*Imagenet.

We confirm these ideas can be applied to other base metric learners. Table 3.5 shows the 5-way-1-shot results for both RelationNet [159] and ProtoNet [150] base learners controlling for these features. We can see that both architectures benefit from deep comparisons and regularizers. However the benefit is greater for RelationNet, which we attribute to the learnable non-linear relation modules. These can learn a different comparison function at each abstraction level, but are also more complex so benefit more from the additional regularization.

Model	*mini*ImageNet 5-way-1-shot Acc.
RN2 Full model	$63.19 \pm 0.87\%$
RN2-No module weight	$62.88 \pm 0.83\%$
RN2-No noise	$60.57 \pm 0.86\%$
RN2-No retrain	$60.79 \pm 0.88\%$
RN2-No retrain, No noise	$58.04 \pm 0.82\%$
RN2-No deep sup.	$58.02 \pm 0.80\%$
RN2-r_1	$52.25 \pm 0.80\%$
RN2-r_2	$58.07 \pm 0.80\%$
RN2-r_3	$60.69 \pm 0.81\%$
RN2-r_4	$58.31 \pm 0.79\%$

Table 3.6 **Ablation study to evaluate the regularization and multiple relation modules.** Accuracies are calculated on 5-way-1-shot classification of *mini*ImageNet.

3.5.2 Ablation Study

We further investigate the detailed design parameters of our method with a series of ablation studies reported in Table 3.6. **Parameterized Gaussian Noise Regularization:** Comparing RN2 and RN2-No Noise, we can see that this brings over 2% improvement. **Retraining:** The impact of retraining on the combined training and validation set is visible by comparing the entries with RN2-No Retrain, which provides a similar 2% margin, and this is complementary to the noise. **Deep Supervision:** The RN2-No Deep Sup. result shows that deep supervision is important to gain full benefit from a column of relation modules. **Module Weighting:** Our model learns the attention weight automatically during meta-training, which eliminates the need for hand-tuning. Compared to manually tuned module weights or no weights, learning weights per module helps somewhat. **Multiple Non-linear Metrics:** Table 3.6 also shows the testing accuracy with each relation module output score r_v in isolation (RN2-r_v). Each module performs competitively, but their combination clearly leads to the best overall performance, supporting our argument that multiple levels of the feature hierarchy should be used to make general purpose matching decisions. Multiple meta learner design is a creative contribution of our work. **Architecture:** Our model benefits from deeper embedding architectures (Table 3.2). It improves when going from simple convolutional blocks (used by early studies [34, 150, 159]), to ResNet [56] and SENet [66]. For fair comparison, when fixing a common ResNet-12, our model outperforms the others that do not require meta-test optimization. Moreover, when fixing a common SENet, competitors RelationNet/ProtoNet/MAML are improved, but still surpassed by our model.

3.5.3 Relation Module Analysis

A key contribution is to perform metric learning at multiple abstraction levels simultaneously via a series of paired relation and embedding modules. Relation modules are analyzed to provide insight into the complementarity.

Score-Distance Correlation We firstly check how the relation module (RM) scores relate to distances in the ImageNet hierarchy [81]. We search for $(support1, support2, query)$ category tuples where the distance $D(query, support1)$ and $D(query, support2)$ match a certain number of links, and then plot instances from these tuples query categories against the relative relation module scores $RM(q, s1)$, $RM(q, s2)$. Figure 3.2 presents scatter plots for the four relation modules where points are images and colors indicate category tuples with specified distance from the two support classes. We can see that: (1) The scores generally match ImageNet distances: The most/least similar categories (red/magenta) are usually closer to the top right/bottom left of the plot; while query categories closer to one support class are in the opposite corners (blue/yellow-green). (2) Generally higher numbered relation modules are more discriminative, separating classes with larger differences in relation score.

Score Correlation We next investigated if relation module predictions are diverse or redundant. We analyzed the correlation in their predictions by randomly picking 10,000 image pairs from *mini*ImageNet and computing the Spearman rank-order correlation coefficient [151] between each pair of relation module's scores. The results in Table 3.7, show that: (1) Many correlations are relatively low (down to 0.34), indicating that they are making diverse, non-redundant predictions; (2) Adjacent RMs have higher correlation than non-adjacent RMs, indicating that prediction diversity is related to RM position in the feature hierarchy.

Module	RM1	RM2	RM3	RM4
RM1	-	-	-	-
RM2	0.75	-	-	-
RM3	0.55	0.73	-	-
RM4	0.34	0.45	0.61	-

Table 3.7 **Spearman rank-order correlation coefficient between different relation modules.** Results show that different modules make diverse predictions.

Prediction Success by Module We know that RM predictions do not necessarily agree. But to find out if they are complementary, we made a scatter plot of the per-class accuracy of RM-1 vs RM-4 in Figure 3.3. We can see that many categories lie on the diagonal, indicating that RM-1 and RM-4 get them right equally often. However there are some categories *below* the diagonal, indicating that RM-1 gets them right more often than RM-4. Examples include both stereotyped and fine-grained categories such as 'hourglass' and 'African hunting dog'. These below diagonal elements confirm the value of using deeper features in metric learning.

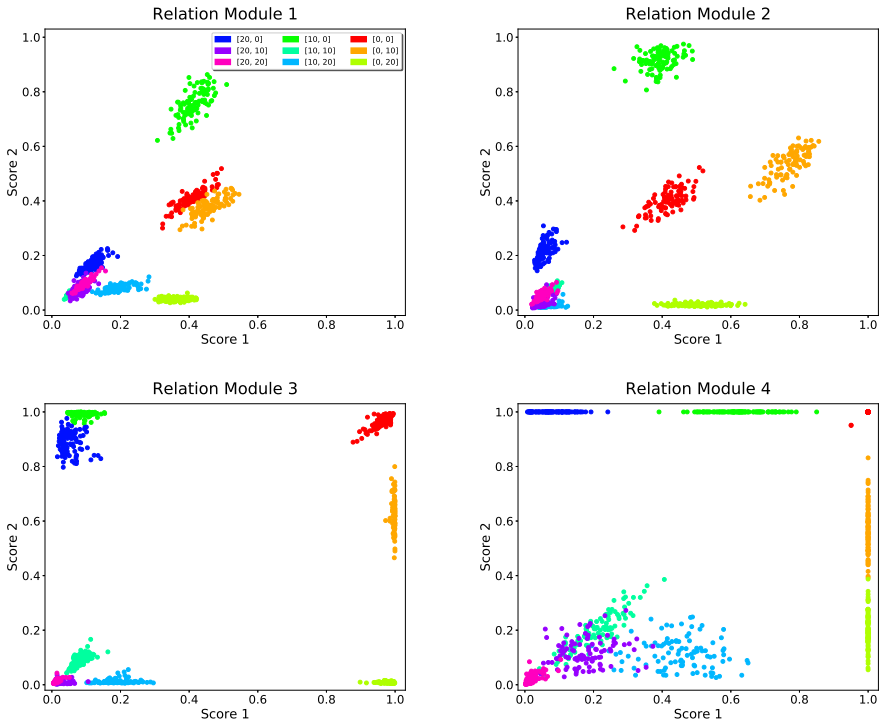


Figure 3.2 **Illustration of query-support score distribution and the link to ImageNet hierarchy.** Colors indicate query images of a $(query, support1, support2)$ class triple matching the specified ImageNet distance relationship $[D(q,s1), D(q,s2)]$.

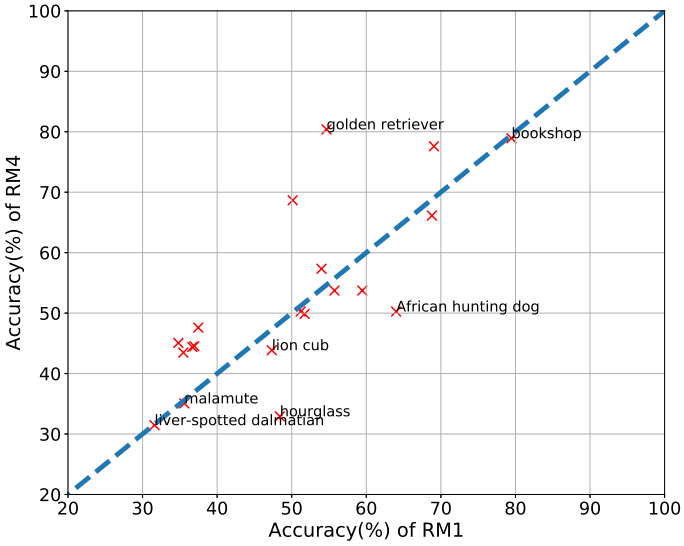


Figure 3.3 **Category-wise accuracy of RM1 vs RM4.** Different relation modules are better at detecting different categories.

3.6 Summary

In this chapter, we proposed RelationNet2, a.k.a. DCN, a new general purpose matching framework for few-shot learning. This architecture performed effective few-shot learning via learning multiple non-linear comparisons simultaneously corresponding to multiple levels of feature extraction, while resisting overfitting. The resulting method achieved state-of-the-art results on *mini*ImageNet and the more ambitious *tiered*ImageNet, while retaining architectural simplicity, and fast training and testing processes.

However, Relation Network uses the matching score to evaluate the similarity between support and query data, of which the computing cost is exponentially increased by the size of dataset. Our multi-metric RelationNet2 is based on the same paradigm of similarity evaluation with poor compute scalability to a large support set and a high number of way.

Chapter 4

Shallow Bayesian MetaQDA

Many state-of-the-art few-shot learners focus on developing effective training procedures for feature representations, before using simple (e.g., nearest centroid) classifiers. In this chapter, we take an approach that is agnostic to the features used, and focus exclusively on meta-learning the final classifier layer. Specifically, we introduce MetaQDA, a Bayesian meta-learning generalization of the classic quadratic discriminant analysis. This approach has several benefits of interest to practitioners: meta-learning is fast and memory efficient, without the need to fine-tune features. It is agnostic to the off-the-shelf features chosen, and thus will continue to benefit from future advances in feature representations. Empirically, it leads to excellent performance in cross-domain few-shot learning, class-incremental few-shot learning, and crucially for real-world applications. The Bayesian formulation leads to state-of-the-art uncertainty calibration in predictions.

4.1 Introduction

Few-shot recognition methods aim to solve classification problems with limited labelled training data. The practical importance of this capability across diverse sparse data applications has motivated a large body of work [177]. Contemporary approaches to few-shot recognition are characterized by a focus on deep meta-learning [62] methods that provide data efficient learning of new categories by using auxiliary data to train a model designed for rapid adaptation to new categories [34, 198], or for synthesizing a classifier for new categories in a feed-forward manner [106, 127]. Many of these meta-learning methods are intimately interwoven with the training algorithm and/or architecture of the deep network that they build upon. For example, many have relied on episodic training schemes [150, 170], where few-shot learning problems are simulated at each iteration of training; differentiable

optimizers [14, 84]; or new neural network modules [159, 41, 133] to facilitate data efficient learning and recognition.

Against this backdrop, a handful of recent studies [176, 45, 20, 102, 189, 172] have pushed back against deep meta-learning. They have observed, for example, that a well tuned convolutional network pre-trained for multi-class recognition and combined with a simple linear or nearest centroid classifier can match or outperform state-of-the-art meta-learners. Even self-supervised pre-training [102] has led to feature extractors that outperform many meta-learners. These analyses raise the question: *is meta-learning indeed beneficial, or is focusing on improving conventional pre-training sufficient?*

We take a position in defense of meta-learning for few-shot recognition. To disentangle the influences of meta-learning per-se and feature learning discussed above, we restrict ourselves to *fixed pre-trained features* and conduct no feature learning in this study. This result shows that both camps are correct: good vanilla pre-training strategies do provide strong downstream few-shot learning; but also meta-learning, even in its shallowest form, can boost few-shot learning above and beyond whatever is provided by the pre-trained features alone.

We take an amortized Bayesian inference approach [47, 59] to shallow meta-learning. During meta-testing, we infer a distribution over classifier parameters given the support set; and during meta-training we learn a feed-forward inference procedure for these parameters. While the limited recent work in Bayesian meta-learning is underpinned by amortized Variational Inference [47], our approach relies instead on conjugacy [82]. Specifically, we build upon the classic Quadratic Discriminant Analysis (QDA) [38] classifier and extended it with a Bayesian prior, an inference pipeline for the QDA parameter posterior given the support set, and gradient-based meta-training. We term the overall framework MetaQDA.

MetaQDA has several important practical benefits for real-world deployments. Firstly, many real-world applications lack computing infrastructure for end-to-end training [71]. MetaQDA allows few-shot meta-learning to be conducted in such resource constrained scenarios, while providing superior performance to recent fixed-feature approaches [176, 20, 102]. Furthermore by decomposing representation learning from classifier meta-learning, MetaQDA is expected to benefit from continued progress in CNN architectures and training strategies. Indeed our empirical results show that MetaQDA's feature-agnostic meta-learning strategy benefits a diverse range of classic and recent feature representations.

As computer vision systems begin to be deployed in high-consequence applications where safety [83] or fair societal outcomes [104] are at stake, their *calibration* becomes as equally, or more, important as their actual accuracy. E.g., models must reliably report low-certainty in those cases where they do make mistakes, thus allowing their decisions in those cases to

be reviewed. Indeed, proper calibration is a hard requirement for deployment in many high importance applications [49, 115]. Crucially, we show that our Bayesian MetaQDA leads to significantly better calibrated models than the standard classifiers in the literature.

Finally, we show that MetaQDA has particularly good performance in cross-domain scenarios where existing methods are weak [20], but which are ubiquitious in practical applications, where there is invariably insufficient domain-specific data to conduct in-domian meta-learning [51].

To summarize our contributions: (i) We present MetaQDA, a novel and efficient Bayesian approach to classifier meta-learning based on conjugacy. (ii) We empirically demonstrate that MetaQDA's efficient fixed feature learning provides excellent performance across a variety of settings and metrics including conventional, cross-domain, class-incremental, and probability calibrated few-shot learning. (iii) We shed light on the meta-learning vs vanilla pre-training debate by disentangling the two and showing a clear benefit from meta-learning, across a variety of fixed feature representations.

4.2 Related Work

4.2.1 Few-Shot and Meta-Learning Overview

Few-shot and meta-learning are now a widely studied area that is too broad to review here. We refer the reader to comprehensive recent surveys for an introduction and review [177, 62]. In general they proceed in two stages: meta-training the strategy for few-shot learning based on one or more auxiliary datasets; and meta-testing (learning new categories) on a target dataset, which should be done data-efficiently given the knowledge from meta-training. A high level categorization of common approaches groups them into methods that (1) meta-learn how to perform rapid gradient-based adaptation during meta-test [34, 198]; and (2) meta-learn a feed-forward procedure to synthesize a classifier for novel categories given an embedding of the support set [47, 127], where metric-based learners are included in the latter category [62].

4.2.2 Is Meta-Learning Necessary?

Many recent papers have questioned whether elaborate meta-learning procedures are necessary. SimpleShot [176] observes vanilla CNN features pre-trained for recognition achieve near SotA performance when appropriately normalized and used in a trivial nearest centroid classifier (NCC). Chen et al. [20] present the simple but high-performance Baseline++, based on fixing a pre-trained feature extractor and then building a linear classifier during meta-test. Goldblum et al. [45] observe that although SotA meta-learned deep features do exhibit

strong performance in few-shot learning, this feature quality can be replicated by adding simple compactness regularisers to vanilla classifier pre-training. S2M2 [102] demonstrates that after pre-training a network with self-supervised learning and/or manifold-regularised vanilla classification, excellent few-shot recognition is achieved by simply training a linear classifier on the resulting representation. Chen et al. [189] analyze whether the famous MAML algorithm is truly meta-learning, or simply pre-training a strong feature.

We show that for fixed features pre-trained by several of the aforementioned "off-the-shelf" non-meta techniques [176, 102], meta-learning *solely* in classifier-space further improves performance. This allows us to conclude that meta-learning *does* add value, since alternative vanilla (i.e., non-meta) pre-training approaches do not influence the final classifier. We leave conclusive analysis of the relative merits of meta-learning vs vanilla pre-training of feature representation space to future work. In terms of empirical performance, we surpass all existing strategies based on fixed pre-trained features, and most alternatives based on deep feature meta-learning.

4.2.3 Fixed Feature Meta-Learning

A minority of meta-learning studies such as [138, 94] have also built on fixed features. LEO [138] synthesizes a classifier layer for a fixed feature extractor using a hybrid gradient- and feedforward-strategy. The concurrent URT [94] addresses multi-domain few-shot learning by meta-training a module that fuses an array of fixed features and dynamically produces a new feature encoding for a new domain. Ultimately, URT uses a ProtoNet [150] classifier, and thus our contribution is orthogonal to URT's, as MetaQDA aims to replace the classifier (ie, ProtoNet), not produce a new feature. Indeed we show empirically that MetaQDA can use URT's feature and improve their performance, further demonstrating the flexibility of our feature-agnostic approach.

4.2.4 Bayesian Few-Shot Meta-Learning

Relatively few methods in the literature take Bayesian approaches to few-shot learning. A few studies [48, 190] focus on understanding MAML [34] as a hierarchical Bayesian model. Versa [47] treats the weights of the final linear classifier layer as the quantity to infer given the support set during meta-test. It takes an amortized variational inference (VI) approach, training an inference neural network to predict the classifier parameters given the support set. However, unlike us, it then performs end-to-end representation learning, and is not fully Bayesian as it does not ultimately integrate the classifier parameters, as we achieve here. Neural Processes [41] takes a Gaussian Process (GP) inspired approach to neural network

design, but ultimately does not provide a clear Bayesian model. The recent DKT [122] achieves true Bayesian meta-learning via GPs with end-to-end feature learning. However, despite performing feature learning, these Bayesian approaches have generally not provided SotA benchmark performance compared to the broader landscape of competitors at the time of their publication. A classic study [59] explored shallow learning-to-learn of linear regression by conjugacy. We also exploit conjugacy but for classifier learning, and demonstrate SotA results on heavily benchmarked tasks for the first time with Bayesian meta-learning.

4.2.5 Classifier Layer Design

The vast majority of few-shot studies use either linear [102, 47, 29], cosine similarity [127], or nearest centroid classifiers [176, 150] under some distance metric. We differ in: (i) using a quadratic classifier, and (ii) taking a "generative" approach to fitting the model [55]. While a quadratic classifier potentially provides a stronger fit than a linear classifier, its larger number of parameters will overfit catastrophically in a few-shot/high-dimension regime. This is why few studies have applied them, with the exception of [6] who had to carefully hand-craft regularisers for them. Our key insight is to use conjugacy to enable the quadratic classifier prior to be efficiently meta-learned, thus gaining improved fitting strength, while avoiding overfitting.

4.3 Probabilistic Meta-Learning

One can formalise a conventional classification problem as consisting of an input space $\mathcal{X}$, an output space $\mathcal{Y}$, and a distribution p over $\mathcal{X} \times \mathcal{Y}$ that defines the task to be solved. Few-shot recognition is the problem of training a classifier to distinguish between C different classes in a sparse data regime, where only K labelled training instances are available for each class. Meta-learning aims to distill relevant knowledge from multiple related few-shot learning problems into a set of shared parameters that boost the learning of subsequent novel few-shot tasks. The simplest way to extend the standard formalisation of classification problems to a meta-learning context is to instead consider the set, $\mathcal{P}$ of all distributions over $\mathcal{X} \times \mathcal{Y}$, each of which represents a possible classification task. One can then assume the existence of a distribution, Q over $\mathcal{P}$ [8].

From a probabilistic perspective, the parameters inferred by the meta-learner that are shared across tasks, which we denote by ϕ, can be seen as specifying or inducing a prior distribution over the task-specific parameters for each few-shot problem. As such, meta-learning can be thought of as learning a procedure to induce a prior over models for future

tasks by meta-training on a collection of related tasks. Representing task-specific parameters for task t by θ_t, the few-shot training (aka support) and testing (aka query) sets as D_S^t and D_Q^t, a Bayesian few-shot learner should use the learned prior to determine the posterior distribution over model parameters,

$$p(\theta_t|D_S^t,\phi) = \frac{p(D_S^t|\theta_t)p(\theta_t|\phi)}{\int p(D_S^t|\theta_t)p(\theta_t|\phi)\mathrm{d}\theta_t}. \tag{4.1}$$

Once this distribution is obtained, one can model novel query instances, $(\vec{x}_i^t, y_i^t) \in D_Q^t$, using the posterior predictive distribution,

$$p(D_Q^t|D_S^t,\phi) = \prod_{i=1}^{|D_Q^t|} \int p(\vec{x}_i^t, y_i^t|\theta_t)p(\theta_t|D_S^t,\phi)\mathrm{d}\theta_t. \tag{4.2}$$

A natural measure for the goodness of fit for ϕ is the expected log likelihood of the few-shot models that make use of the shared prior,

$$\mathop{\mathbb{E}}_{D_S,D_Q\sim q,q\sim Q}[L(\phi|D_S,D_Q)], \tag{4.3}$$

where

$$L(\phi|D_S,D_Q) = \sum_{i=1}^{|D_Q|} \log p(\vec{x}_i, y_i|D_S,\phi). \tag{4.4}$$

The process of meta-learning the prior parameters can then be formalised as a risk minimisation problem,

$$\phi^* = \arg\min_{\phi} \mathop{\mathbb{E}}_{D_S,D_Q\sim q,q\sim Q}[-L(\phi|D_S,D_Q)]. \tag{4.5}$$

Discussion A prior probabilistic meta-learner [47] focused on the term $p(\theta_t|D_S^t,\phi)$, taking an amortized variational inference perspective that treats ϕ as the parameters of a neural network that predicts a distribution over the parameters θ_t of a linear classifier given support set D_S^t. In contrast, our framework will use a QDA rather than linear classifier, and then exploit conjugacy to efficiently compute a distribution over the QDA mean and covariance parameters θ_t given the support set. This is both efficient and probabilistically cleaner, as our model contains a proper prior, while [47] does not.

The integrals in Equation 4.1 and 4.2 are key to Bayesian meta-learning, but can be computationally intractable and [47] relies on sampling. Our conjugate setup allows the integrals to be computed exactly in closed form, without relying on sampling.

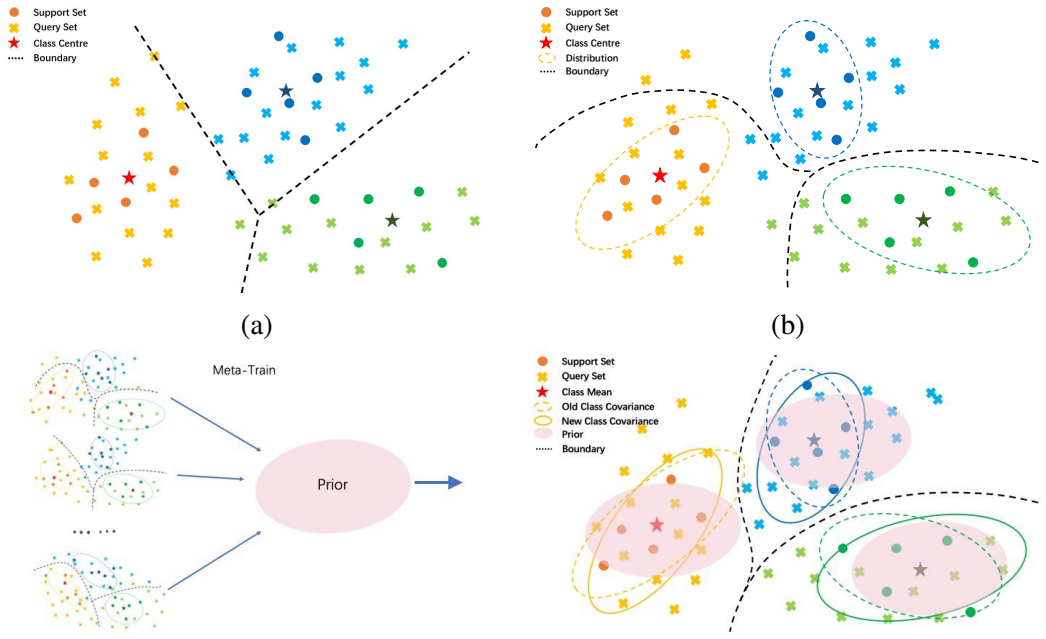


Figure 4.1 **Illustrative schematic of MetaQDA.** (a) NCC classifier uses the class mean to induce linear decision boundaries. (b) QDA uses both the support class mean and covariance to induce a curved decision boundary, but easily overfits in a few-shot regime due. (c) MetaQDA meta-learns the QDA parameter prior to provide stable estimation of a non-linear decision boundary without overfitting.

An Illustrative Example To illustrate the mechanism of MetaQDA, we compare it schematically to conventional linear classifier used in many studies [20, 150, 94], and vanilla QDA in Figure 4.1. In the figure, the colored circles indicate 3-way-5-shot support datasets, and the "x" data points are the query set of the corresponding color. The dashed line is the decision boundary of different classifiers. Figure 4.1(a) shows Nearest Centre Classifier (NCC) [150, 94], where the stars represent the mean of the support set class distributions, and these induce linear decision boundaries. Figure 4.1(b) depicts the Quadratic Discriminant Analysis (QDA) classifier, where the dashed ellipses represent the class covariance models, estimated from the support set. These induce a non-linear decision boundary. Figure 4.1(c) illustrates our MetaQDA, where the meta-training process learns a shared NIW prior (the shadow ellipse) from many few-shot training tasks. Then MetaQDA uses conjugacy to update the class covariances (solid line) using the support set and prior, and so induces a better non-linear decision boundary.

This illustrates how the MetaQDA setup allows us to exploit the benefit of a non-linear classifier, without the associated overfitting risk that would normally undermine such an attempt (as illustrated by the poor results of vanilla MetaQDA in the following experiments.

4.4 Meta-Quadratic Discriminant Analysis

Our MetaQDA provides a meta-learning generalization of the classic QDA classifier [55]. QDA works by constructing a multivariate Gaussian distribution θ corresponding to each class by maximum likelihood. At test time, predictions are made by computing the likelihood of the query instance under each of these distributions, and using Bayes theorem to obtain the posterior $p(y|x,\theta)$. Rather than using maximum likelihood fitting for meta-testing, we introduce a Bayesian version of QDA that will enable us to exploit a meta-learned prior over the parameters of the multivariate Gaussian distributions. Two Bayesian strategies for inference using such a prior are explored: 1) using the maximum a posterior (MAP) estimate of the Gaussian parameters; and 2) the fully Bayesian approach that propagates the parameter uncertainty through to the class predictions. The first of these is conceptually simpler, while the second allows for better handling of uncertainty due to the fully Bayesian nature of the parameter inference. For both cases, we make use of Normal-Inverse-Wishart priors [82], as their conjugacy with multivariate Gaussians leads to an efficient implementation strategy.

4.4.1 MAP-Based QDA

We begin by describing a MAP variant of QDA. In conventional QDA the likelihood of an instance, $\vec{x} \in \mathbb{R}^d$, belonging to class $j \in \mathbb{N}_C$ is given by $\mathcal{N}(\vec{x}|\vec{\mu}_j, \Sigma_j)$ and the parameters are found via maximum likelihood estimation (MLE) on the subset of the support set associated with class j,

$$\vec{\mu}_j, \Sigma_j = \arg\max_{\vec{\mu},\Sigma} \prod_{i=1}^{K} \mathcal{N}(\vec{x}_{j,i}|\vec{\mu}, \Sigma). \tag{4.6}$$

This optimisation problem has a convenient closed form solution: the sample mean and covariance of the relevant subset of the support set. In order to incorporate prior knowledge learned from related few-shot learning tasks, we define a Normal-inverse-Wishart (NIW) prior [110] over the parameters and therefore obtain a posterior for the parameters,

$$\begin{aligned} &p(\vec{\mu}_j, \Sigma_j|\vec{x}, \vec{m}, \kappa, S, \nu) \\ &= \frac{\prod_{i=1}^{K} \mathcal{N}(\vec{x}_{j,i}|\vec{\mu}_j, \Sigma_j)\mathcal{NIW}(\vec{\mu}_j, \Sigma_j|\vec{m}, \kappa, S, \nu)}{\int\int \prod_{i=1}^{K} \mathcal{N}(\vec{x}_{j,i}|\vec{\mu}, \Sigma)\mathcal{NIW}(\vec{\mu}', \Sigma'|\vec{m}, \kappa, S, \nu) d\vec{\mu}' d\Sigma'}. \end{aligned} \tag{4.7}$$

Training This enables us to take advantage of prior knowledge learned from related tasks when inferring the model parameters by MAP inference,

$$\vec{\mu}_j, \Sigma_j = \arg\max_{\vec{\mu}, \Sigma} \prod_{i=1}^{K} p(\vec{\mu}_j, \Sigma_j | \vec{x}_{j,i}, \vec{m}, \kappa, S, \nu). \tag{4.8}$$

Because NIW is the conjugate prior of multivariate Gaussians, we know that the posterior distribution over the parameters takes the form of

$$p(\vec{\mu}_j, \Sigma_j | \vec{x}, \vec{m}, \kappa, S, \nu) = \mathcal{NIW}(\vec{\mu}_j, \Sigma_j | \vec{m}_j, \kappa_j, S_j, \nu_j), \tag{4.9}$$

where

$$\begin{aligned} \vec{m}_j &= \frac{\vec{m} + K\hat{\vec{\mu}}_j}{\kappa + K}, \quad \kappa_j = \kappa + K, \quad \nu_j = \nu + K, \\ S_j &= S + \sum_{i=1}^{K} (\vec{x}_{j,i} - \hat{\vec{\mu}}_j)(\vec{x}_{j,i} - \hat{\vec{\mu}}_j)^T + \\ &\quad \frac{\kappa K}{\kappa + K} (\hat{\vec{\mu}}_j - \vec{m})(\hat{\vec{\mu}}_j - \vec{m})^T, \end{aligned} \tag{4.10}$$

and we have used $\hat{\vec{\mu}}_j = \frac{1}{k} \sum_{i=1}^{K} \vec{x}_{j,i}$. The posterior is maximised at the mode, which occurs at

$$\vec{\mu}_j = \vec{m}_j, \quad \Sigma_j = \frac{1}{\nu_j + d + 1} S_j. \tag{4.11}$$

Testing After computing point estimates of the parameters, one can make predictions on instances from the query set according to the usual QDA model,

$$p(y = j | \vec{x}, \vec{m}, \kappa, S, \nu) = \frac{\mathcal{N}(\vec{x} | \vec{\mu}_j, \Sigma_j) p(y = j)}{\sum_{i=1}^{C} \mathcal{N}(\vec{x} | \vec{\mu}_i, \Sigma_i) p(y = i)}. \tag{4.12}$$

Note the prior over the classes $p(y)$ can be dropped in the standard few-shot benchmarks that assume a uniform distribution over classes.

4.4.2 Fully Bayesian QDA

Computing point estimates of the parameters throws away potentially useful uncertainty information that can help to better calibrate the predictions of the model. Instead, we can

marginalise the parameters out when making a prediction,

$$p(y=j|\vec{x}) = \frac{\int\int \mathcal{N}(\vec{x}|\mu_j,\Sigma_j)\mathcal{NIW}(\vec{\mu}_j,\Sigma_j|\vec{m}_j,\kappa_j,S_j,\nu_j)d\vec{\mu}_j d\Sigma_j}{\sum_{i=1}^{C}\int\int \mathcal{N}(\vec{x}|\mu_i,\Sigma_i)\mathcal{NIW}(\vec{\mu}_i,\Sigma_i|\vec{m}_j,\kappa_j,S_j,\nu_j)d\vec{\mu}_i d\Sigma_i}. \tag{4.13}$$

Each of the double integrals has the form of a multivariate t-distribution [110], yielding

$$p(y=j|\vec{x},\vec{m},\kappa,S,\nu) = \frac{\mathcal{T}\left(\vec{x}|\vec{m}_j,\frac{\kappa_j+1}{\kappa_j(\nu_j-d+1)}S_j,\nu_j-d+1\right)}{\sum_{i=1}^{C}\mathcal{T}\left(\vec{x}|\vec{m}_i,\frac{\kappa_i+1}{\kappa_i(\nu_i-d+1)}S_i,\nu_i-d+1\right)}. \tag{4.14}$$

4.4.3 Meta-Learning the Prior

Letting $\phi=(\vec{m},\kappa,S,\nu)$, our objective is to minimise the negative expected log likelihood of models constructed with the shared prior on the parameters, as given in Equation 4.5. For MAP-based QDA, the log likelihood function is given by

$$L(\phi|D_S,D_Q)=\sum_{j=1}^{C}\sum_{i=1}^{K}\log\mathcal{N}(\vec{x}_{j,i}|\vec{\mu}_j,\Sigma_j), \tag{4.15}$$

where $\vec{\mu}_j$ and Σ_j are the point estimates computed via the closed-form solution to the MAP inference problem given in Equation 4.11. When using the fully Bayesian variant of QDA, we have the following log likelihood function:

$$L(\phi|D_S,D_Q) = \sum_{j=1}^{C}\sum_{i=1}^{K}\log\mathcal{T}\left(\vec{x}_{j,i}|\vec{m}_j,\frac{\kappa_j+1}{\kappa_j(\nu_j-d+1)}S_j,\nu_j-d+1\right). \tag{4.16}$$

Meta-Training We approximate the optimization in Equation 5.6 by performing empirical risk minimisation on a training dataset using episodic training. In particular, we choose $\mathcal{P}$ to be the set of uniform distributions over all possible C-way classification problems, Q as the uniform distribution over $\mathcal{P}$, and the process of sampling from each $q\in\mathcal{P}$ results in balanced datasets containing K instances from each of the C classes. Episodic training then consists of sampling a few-shot learning problem, building a Bayesian QDA classifier using the support set, computing the negative log likelihood on the query set, and finally updating ϕ using stochastic gradient descent. Crucially, the use of conjugate priors means that no iterative

Algorithm 1: Pseudocode for epsiodic meta-learning of hyper-parameters in MetaQDA.

Require: Distribution over tasks Q, number of iterations T, learning rate α
Result: prior parameters ϕ_T
Init: $\phi_0 = \{\vec{m} = \vec{0}, S = \mathbf{I}, \kappa = 1, \nu = d\}$
for $t = 1$ to T **do**
    Sample task, $q_t \sim Q$;
    Sample support and query set, $D_S^t, D_Q^t \sim q_t$;
    Build Bayesian QDA Model ;
    If MAP: $\theta_t \leftarrow \{(\vec{\mu}_j, \Sigma_j)\}_{j=1}^C$; `// Eq 4.11`
    If Fully Bayes: $\theta_t \leftarrow \{(\vec{m}_j, \kappa_j, S_j, \nu_j)\}_{j=1}^C$; `// Eq 4.10`
    Update Prior
    $\phi_t \leftarrow \phi_{t-1} - \alpha \nabla_\phi L(\phi_{t-1} | D_S^t, D_Q^t)$; `// Eq 4.15 or 4.16`
end

optimisation procedure must be carried out when constructing the classifier in each episode. Instead, we are able to backpropagate through the conjugacy update rules and directly modify the prior parameters with stochastic gradient descent. The overall learning procedure is given in Algorithm 1.

Some of the prior parameters must be constrained in order to learn a valid NIW distribution. In particular, S must be positive definite, κ must be positive, and ν must be strictly greater than $d-1$. The constraints can be enforced for κ and ν by clipping any values that are outside the valid range back to the minimum allowable value. We parameterise the scale matrix in terms of its Cholesky factors,

$$S = LL^T, \tag{4.17}$$

where L is a lower triangular matrix. During optimisation we ensure L remains lower triangular by setting all elements above the diagonal to zero after each weight update.

4.5 Experiments

We measure the efficacy of our model in standard, cross-domain and multi-domain few-shot learning problem settings. We also evaluate the uncertainty calibration error, which is also a significant advantage of our approach. MetaQDA is a shallow classifier-layer meta-learner that is agnostic to the choice of fixed extracted features. Unless otherwise stated, we report results for the FB-based variant of MetaQDA. During meta-training, we learn the priors $\phi = (\vec{m}, \kappa, S, \nu)$ over episodes drawn from the training set, keeping the feature extractor fixed.

We use the meta-validation datasets for model selection and hyperparameter tuning. During meta-testing, the support set is used to obtain the parameter posterior, and then a QDA classifier is established according to either Equation 4.12 or Equation 4.14. All algorithms are evaluated on C-way k-shot learning [150], with a batch of 15 query images per class in a testing episode. All accuracies are calculated by averaging over 600 randomly generated testing tasks with 95% confidence interval.

4.5.1 *mini*ImageNet

Dataset ***mini*ImageNet** [129] is split into 64/16/20 for meta-train/val/test, respectively, containing 100 classes and 600 examples per class, drawn from ILSVRC-12 [137]. Images are resized to 84×84 [56].

Settings As for **Conv-4 extractor** and **ResNet-18 extractor**, following [176], we use stochastic gradient descent (SGD) with a multi-step learning rate schedule, momentum of 0.9, and the initial learning rate is set to 0.01. At epochs 70 and 100 we reduce the learning rate by a factor of 0.1. Weight decay is set as 0.0001 through out training. Batch size is 256 images. In terms of **WRN-28-10 extractor**, following [102], as for 1-shot classification on *mini*ImageNet, we use stochastic gradient descent (SGD) with a multi-step learning rate schedule, momentum of 0.9, and the initial learning rate is set to 0.001. But as for 5-shot classification on *mini*ImageNet, we use ADAM optimiser.

Competitors We group competitors into two categories: (1) direct competitors that also make use of 'off-the-shelf' fixed pre-trained networks and only update the classifier to learn novel classes; and (2) non-direct competitors that specifically meta-learn a feature optimised for few-shot learning and/or update features during meta-testing. We do not attempt to be comprehensive in SotA comparison with latter *learnable* feature alternatives since our academic and practical motivation is fixed-feature meta-learning as explained earlier. *Baseline++* [20] fixes the feature encoder and only tunes the (cosine similarity) classifier during the meta-test stage. *SimpleShot* [176] uses an NCC classifier with different feature encoders and studies different feature normalizations. We use their best reported variant, CL2N. *S2M2* [102] uses a linear classifier after self-supervised and/or regularized classifier pre-training. *SUR* [29] also uses pre-trained feature extractors, but focuses on weighting multiple features extracted from different backbones or multiple layers of the same backbone. We compare their reported results of a single ResNet backbone trained for multi-class classification as per ours, but they have the advantage of fusing features extracted from multiple layers. Unravelling [45] proposes some new regularizers for vanilla backbone training that improve feature quality for few-shot learning without meta-learning. *PT-MAT*

Model	Backbone	1-shot	5-shot
MATCHINGNETS [170]	Conv-4	43.56 ± 0.84%	55.31 ± 0.73%
METALSTM [129]	Conv-4	43.44 ± 0.77%	60.60 ± 0.71%
MAML^O [34]	Conv-4	48.70 ± 1.84%	63.11 ± 0.92%
PROTONET [150]	Conv-4	49.42 ± 0.78%	68.20 ± 0.66%
GNN [40]	Conv-4	50.33 ± 0.36%	66.41 ± 0.63%
METASSL[132]	Conv-4	50.41 ± 0.31%	64.39 ± 0.24%
RELATIONNET [159]	Conv-4	50.44 ± 0.82%	65.32 ± 0.70%
METASGD^O [92]	Conv-4	50.47 ± 1.87%	64.03 ± 0.94%
CAVIA [198]	Conv-4	51.82 ± 0.65%	65.85 ± 0.55%
TPN [96]	Conv-4	52.78 ± 0.27%	66.59 ± 0.28%
R2D2 [14]	Conv-4*	51.90 ± 0.20%	68.70 ± 0.20%
RELATIONNET2[196]	Conv-4	53.48 ± 0.78%	67.63 ± 0.59%
GCR [88]	Conv-4	53.21 ± 0.40%	72.34 ± 0.32%
VERSA [47]	Conv-4	53.40 ± 1.82%	67.37 ± 0.86%
DYNAMICFSL† [44]	Conv-4	56.20 ± 0.86%	72.81 ± 0.62%
BASELINE++ [20]	Conv-4	48.24 ± 0.75%	66.43 ± 0.63%
SIMPLESHOT[176]	Conv-4	49.69 ± 0.19%	66.92 ± 0.17%
METAQDA	Conv-4	**56.41 ± 0.80%**	**72.64 ± 0.62%**
SNAIL [140]	ResNet-12	55.71 ± 0.99%	68.88 ± 0.92%
DYNAMIC FSL [44]	ResNet-12	55.45 ± 0.89%	70.13 ± 0.68%
ADARESNET [109]	ResNet-12	57.10 ± 0.70%	70.04 ± 0.63%
TADAM [116]	ResNet-12	58.50 ± 0.30%	76.70 ± 0.30%
CAML [72]	ResNet-12	59.23 ± 0.99%	72.35 ± 0.18%
AM3 [182]	ResNet-12	65.21 ± 0.49%	75.20 ± 0.36%
MTL [156]	ResNet-12*	61.20 ± 1.80%	75.50 ± 0.80%
TAP NET [191]	ResNet-12	61.65 ± 0.15%	76.36 ± 0.10%
RELATIONNET2[196]	ResNet-12	63.92 ± 0.98%	77.15 ± 0.59%
R2D2[14]	ResNet-12	59.38 ± 0.31%	78.15 ± 0.24%
METAOPT^O [84]	ResNet-12*	64.09 ± 0.62%	80.00 ± 0.45%
RELATIONNET [20]	ResNet-18	52.48 ± 0.86%	69.83 ± 0.68%
PROTONET [20]	ResNet-18	54.16 ± 0.82%	73.68 ± 0.65%
DCEM [28]	ResNet-18	58.71 ± 0.62%	77.28 ± 0.46%
AFHN [90]	ResNet-18	62.38 ± 0.72%	78.16 ± 0.56%
SUR[29]	ResNet-12	60.79 ± 0.62%	79.25 ± 0.41%
UNRAVELLING[45]	ResNet-12*	59.37 ± 0.32%	77.05 ± 0.25%
BASELINE++ [20]	ResNet-18	51.87 ± 0.77%	75.68 ± 0.63%
SIMPLESHOT[176]	ResNet-18	62.85 ± 0.20%	80.02 ± 0.14%
S2M2 [102]	ResNet-18	64.06 ± 0.18%	80.58 ± 0.12%
METAQDA	ResNet-18	**65.12 ± 0.66%**	**80.98 ± 0.75%**
LEO^O [138]	WRN	61.78 ± 0.05%	77.59 ± 0.12%
PPA [127]	WRN	59.60 ± 0.41%	73.74 ± 0.19%
SIMPLESHOT[176]	WRN	63.50 ± 0.20%	80.33 ± 0.14%
S2M2 [102]	WRN	64.93 ± 0.18%	83.18 ± 0.22%
METAQDA	WRN	**67.83 ± 0.64%**	**84.28 ± 0.69%**

Table 4.1 **Few-shot classification results on *mini*ImageNet.** †: two-step optimization with attention. ^{O}: requires gradient-based optimisation at meta-test time. *: uses a wider CNN than standard and higher dimensional embedding. Grey: fixed feature methods.

[68] is a transfer-based method building on preprocessing the feature vectors close to Gaussian distributions, and then leveraging this preprocessing features to an optimal-transport inspired algorithm. Actually, in *mini*ImageNetimplementation process, this method used extracted features pre-trained on Imagenet, which is much larger than *mini*ImageNetto make it an unfair comparison with other baselines.

Results Table 4.1 summarizes the results on *mini*ImageNet. MetaQDA performs better than all the previous methods that rely on off-the-shelf feature extractors, and also the majority of methods that meta-learn representations specialised for few-shot problems. We do not make efforts to carefully fine-tune the hyperparameters, but focus on showing that our model has robust advantages in different few-shot learning benchmarks with various backbones. A key benefit of fixed feature approaches (grey) is small *compute cost*, e.g., under 1-hour training. In contrast, the other state-of-the-art end-to-end competitors (white) such as [84, 44, 196] require over 10 hours.

4.5.2 *tiered*ImageNet

***tiered*ImageNet** is a more challenging benchmark [132] consisting of 608 classes (779,165 images) and is divided into 39197/160 classes for meta-train fold, 97 classes for meta-val fold, and 160 classes for meta-test fold, respectively. Images are also resized to 84×84.

Settings For both *Conv-4 extractor* and *ResNet-18 extractor*, following [176], we use stochastic gradient descent (SGD) with a multi-step learning rate schedule, momentum of 0.9, and the initial learning rate is set to 0.001. At epochs 70 and 100 we reduce the learning rate by a factor of 0.1. Weight decay is set as 0.0001 throughout training. Batch size is 256. In terms of *WRN-28-10 extractor*, following [102], as for 1-shot classification on *tiered*ImageNet, we use ADAM optimiser.

Results Table 4.2 shows that MetaQDA performs state-of-the-art on *tiered*ImageNet. Similar to *mini*ImageNet, we split the competitors to direct and indirect competitors, and use off-the-shelf feature extractors without fine-tuning. Because *tiered*ImageNetis bigger than *mini*ImageNet, it is much more important to save the computation time with holding the accuracy performance, which is the prominent advantage and contribution of our methodology. Specifically, MetaQDA achieves the best performance regarding different backbones, and way beyond more than 6% advantages with the shallow Conv-4 feature extractor.

Model	Backbone	1-shot	5-shot
REPTILE [96]	Conv-4	48.97%	66.47%
MAML [96]	Conv-4	51.67 ± 1.81%	70.30 ± 1.75%
METASSL† [132]	Conv-4	52.39 ± 0.44%	70.25 ± 0.31%
RELATIONNET [96]	Conv-4	54.48 ± 0.48%	71.31 ± 0.78%
TPN† [96]	Conv-4	59.91 ± 0.94%	73.30 ± 0.75%
RELATIONNET2[196]	Conv-4	60.58 ± 0.72%	72.42 ± 0.69%
PROTONET [96]	Conv-4	53.31 ± 0.89%	72.69 ± 0.74%
SIMPLESHOT[176]	Conv-4	51.02 ± 0.20%	68.98 ± 0.18%
METAQDA	Conv-4	**58.11 ± 0.48%**	**74.28 ± 0.73%**
TAPNET [191]	ResNet-12	63.08 ± 0.15%	80.26 ± 0.12%
RELATIONNET2 [196]	ResNet-12	68.58 ± 0.63%	80.65 ± 0.91%
METAOPTNET^{O} [84]	ResNet-12*	65.81 ± 0.74%	81.75 ± 0.53%
SIMPLESHOT [176]	ResNet-18	69.09 ± 0.22%	84.58 ± 0.16%
METAQDA	ResNet-18	**69.97 ± 0.52%**	**85.51 ± 0.58%**
LEO [138]	WRN	66.33 ± 0.05%	81.44 ± 0.09%
SIMPLESHOT [176]	WRN	69.75 ± 0.20%	85.31 ± 0.15%
S2M2 [102]	WRN	73.71 ± 0.22%	88.59 ± 0.14%
METAQDA	WRN	**74.33 ± 0.65%**	**89.56 ± 0.79%**

Table 4.2 **Few-shot classification results on *tiered*ImageNet.** All best-performing results are bold. †: makes use of additional unlabelled data for semi-supervised learning or transductive inference. ^{O}: requires gradient-based optimisation at meta-test time. *: uses a wider ResNet than standard size and higher dimensional embedding. Gray: uses fixed pre-trained backbones.

4.5.3 CIFAR-FS

Dataset **CIFAR-FS** [14] was created by randomly sampling from CIFAR-100 [80] by using the same criteria as *mini*ImageNet (100 classes with 600 images per class, split into folds of 64/16/20 for meta-train/val/test). Images are resized to 32×32.

Settings As for **Conv-4 and ResNet-18 extractor**, following [176], we use stochastic gradient descent (SGD) with a multi-step learning rate schedule, momentum of 0.9, and the initial learning rate is set to 0.01. At epochs 70 and 100 we reduce the learning rate by a factor of 0.1. Weight decay is set as 0.0001 through out training. In terms of **WRN-28-10 extractor**, we use the pre-trained WRN backbone of S2M2 [102].

Results Following former experiment settings, as shown in Table 4.3, we also use 15 query images here and split the competitors. Similarly, as an additional experiment, we also achieve state-of-the-art performance on CIFAR-FS, further verifying the effectiveness and generalization of MetaQDA.

Model	Backbone	1-shot	5-shot
MAML [102]	Conv-4	58.90 ± 1.90%	71.50 ± 1.00%
RELATIONNET [102]	Conv-4	55.50 ± 1.00%	69.30 ± 0.80%
PROTONET [102]	Conv-4	55.50 ± 0.70%	72.02 ± 0.60%
R2D2 [14]	Conv-4	62.30 ± 0.20%	77.40 ± 0.10%
SIMPLESHOT^{+} [176]	Conv-4	59.35 ± 0.89%	74.76 ± 0.72%
METAQDA	Conv-4	**60.52 ± 0.88%**	**77.33 ± 0.73%**
PROTONET [102]	ResNet-12	72.20 ± 0.70%	83.50 ± 0.50%
METAOPT [84]	ResNet-12*	72.00 ± 0.70%	84.20 ± 0.50%
UNRAVELLING [45]	ResNet-12*	72.30 ± 0.40%	86.30 ± 0.20%
BASELINE++ [20, 102]	ResNet-18	59.67 ± 0.90%	71.40 ± 0.69%
S2M2 [102]	ResNet-18	63.66 ± 0.17%	76.07 ± 0.19%
METAQDA	ResNet-18	**72.57 ± 0.48%**	**86.48 ± 0.66%**
METAOPTNET [84]	WRN	72.00 ± 0.70%	84.20 ± 0.50%
BASELINE++ [102, 20]	WRN	67.50 ± 0.64%	80.08 ± 0.32%
S2M2 [102]	WRN	74.81 ± 0.19%	87.47 ± 0.13%
METAQDA	WRN	**75.83 ± 0.88%**	**88.79 ± 0.75%**

Table 4.3 **Few-shot classification results on CIFAR-FS.** $^{+}$: Our implementation. Gray: uses fixed pre-trained backbones.

4.5.4 Cross-Domain Few-Shot Learning

Current metric-based few-shot learning methodologies often fail to generalize to novel domain due to the big difference of new feature distribution across domains. Several prior methods are proposed to tackle the challenge of cross-domain few-shot learning (CD-FSL). Tseng et.al [165] propose an approach meta-learns the hyper-parameters of feature-wise transformation layers to simulate different feature distributions by augmenting the instance features. Similarly, FEAT [188] adapts the instance embeddings to the target classification task with a set-to-set function such as Transformer. Adversarial training is used to battle the domain shift, and the embedding is generalized to a novel task by metric-based learning approach [99, 74]. Guo et.al [51] propose a new benchmark which captures a broader spectrum of image types such as industrial, aerial, and medical images. However, our metric-based meta-learning model (MetaQDA) is composed of an off-the-shelf feature encoder and a Bayesian classifier, providing intuitive generalization ability under domain shift. Then we conduct experiments trained on *mini*ImageNet but tested on CUB and Cars datasets, demonstrating that our MetaQDA is applicable to cross-domain condition directly.

Dataset **CUB** [60] is *Caltech-UCSD Birds 200* dataset with fine-grained classes (200 bird species) dividing into 100 for training, 50 for validation, and 50 for testing. Each image is resized to 84x84 pixels. **Cars** [79, 165] contains 196 classes (16,185 images) randomly split into folds of 98, 49, and 49 classes for meta-train/val/test, respectively.

	Model	Backbone	1-shot	5-shot
*mini*ImageNet→CUB	**MAML** [122]	Conv-4	34.01 ± 1.25%	-
	RELATIONNET [122]	Conv-4	37.13 ± 0.20%	-
	DKT [122]	Conv-4	40.22 ± 0.54%	-
	PROTONET [122]	Conv-4	33.27 ± 1.09%	-
	BASELINE++ [122, 20]	Conv-4	39.19 ± 0.12%	-
	SIMPLESHOT$^+$ [176]	Conv-4	45.36 ± 0.75%	61.44 ± 0.71%
	METAQDA	Conv-4	**47.25 ± 0.58%**	**64.40 ± 0.65%**
	MAML [20]	ResNet-18	-	51.34 ± 0.72%
	RELATIONNET [20]	ResNet-18	-	57.71 ± 0.73%
	LRP (CAN) [155]	ResNet-12	46.23 ± 0.42%	66.58 ± 0.39%
	LRP (GNN) [155]	ResNet-10	48.29 ± 0.51%	64.44 ± 0.48%
	LFWT [165]	ResNet-10	47.47 ± 0.75%	66.98 ± 0.68%
	PROTONET [20]	ResNet-18	-	62.02 ± 0.70%
	BASELINE++ [20]	ResNet-18	42.85 ± 0.69%	62.04 ± 0.76%
	SIMPLESHOT$^+$ [176]	ResNet-18	46.68 ± 0.49%	65.56 ± 0.70%
	METAQDA	ResNet-18	**48.88 ± 0.64%**	**68.59 ± 0.59%**
	S2M2 [102]	WRN	48.24 ± 0.84%	70.44 ± 0.75%
	SIMPLESHOT$^+$ [176]	WRN	49.65 ± 0.24%	66.77 ± 0.19%
	METAQDA	WRN	**53.75 ± 0.72%**	**71.84 ± 0.66%**
*mini*ImageNet→Cars	**SIMPLESHOT**$^+$ [176]	Conv-4	29.52 ± 0.56%	39.52 ± 0.66%
	METAQDA	Conv-4	**30.98 ± 0.66%**	**42.85 ± 0.68%**
	LRP (CAN) [155]	ResNet-12	32.66 ± 0.46%	43.86 ± 0.38%
	LRP (GNN) [155]	ResNet-10	32.78 ± 0.39%	46.20 ± 0.46%
	LFWT [165]	ResNet-10	30.77 ± 0.47%	44.90 ± 0.64%
	SIMPLESHOT$^+$ [176]	ResNet-18	34.72 ± 0.67%	47.26 ± 0.71%
	METAQDA	ResNet-18	**37.05 ± 0.65%**	**51.58 ± 0.52%**
	S2M2 [102]	WRN	31.52 ± 0.59%	47.48 ± 0.68%
	SIMPLESHOT$^+$ [176]	WRN	33.68 ± 0.63%	46.67 ± 0.68%
	METAQDA	WRN	**36.21 ± 0.62%**	**50.83 ± 0.64%**

Table 4.4 **Cross-domain few-shot classification results from *mini*ImageNet to CUB and Cars datasets.** All best-performing results are bold. $^+$: Our implementation. Gray: fixed pre-trained backbones.

Problem Setup Source domain is assumed as a collection of few-shot classification tasks, and the target domain is denoted to evaluate the generalization ability. E.g., meta-learning model can be trained on *mini*ImageNet and test on CUB or Cars. Note that the access to testing domains during meta-training is not allowed in this Problem Setup.

Competitors Better few-shot learning methods should degrade less when transferring to new domains [20, 165]. We are specifically interested in comparing MetaQDA with other methods using off-the-shelf features. In particular, we consider *Baseline++* [20] and *S2M2* [102] that use linear classifiers, and the nearest centroid method of SimpleShot [176].

Results Table 4.4 demonstrates that MetaQDA exhibits good robustness to domain shift. Specifically, our method outperforms other approaches by at least $2\% - 4\%$ across all dataset, support set size, and feature combinations.

4.5.5 Multi-Domain Few-Shot Learning

Multi-domain few-shot learning is obviously a bigger challenge than cross-domain few-shot learning, where both the number of domains and the episodic training procedure change to a more realistic way. Meta-Dataset is proposed to evaluate FSL models with diverse domains of datasets [164]. They leverage the same diverse source training datasets to improve the model's generalization in multi-domain testing. Comparing to standard few-shot learning on *mini*ImageNetand , Meta-Dataset changes the formulation of datasets and tasks (how to generate the training episode), and the details are shown in Section 4.5.5.

Dataset Meta-Dataset [164] is a large-scale benchmark spanning 10 image datasets. Specifically speaking, except for Traffic Signs and MSCOCO reserved for evaluation, the remaining 8 datasets are split roughly with 70/15/15% proportions for training/validation/testing sets. Following [133, 6], we report results using the first 8 datasets for meta training (some classes are reserved for "in-domain" testing performance evaluation), and hold out entirely the remaining 2 (*Traffic Signs* and *MSCOCO*) plus an additional 3 datasets (*MNIST* [186], *CIFAR10*, *CIFAR100* [80]) for an unseen "out-of-domain" performance evaluation. Note that the Meta-Dataset protocol is random way and shot.

Problem Setup Meta-Dataset [164] uses episode sampling mechanism yielding realistically imbalanced episodes of *random-way-random-shot* as shown in Figure 4.2. Each training episode is generated with the classes from the same single domain. During meta-training procedure, to mimic the meta-testing condition, the training tasks are episodically sampled as described above and the choice of the dataset (step 0) is uniformly random. We use both the classification accuracy and the rank computation to compare our models with other state-of-the-art main meta-learning algorithms.

Competitors CNAP [133] and SCNAP [6] meta-learn an adaptive feature extractor whose parameters are modulated by an adaptation network that takes the current task's dataset as input. SUR [29] performs feature selection among a suite of meta-train domain-specific features. The concurrent URT [94] meta-learns a transformer to dynamically meta-train dataset features before nearest-centroid classification with ProtoNet. We apply MetaQDA upon the fixed fused features learned by URT, replacing ProtoNet. **Implementation Details** We use the same backbone as SUR [29] and URT [94], and take the trained fused features by URT [94]. We use ADAM optimizer and cosine learning rate scheduler, and the initial

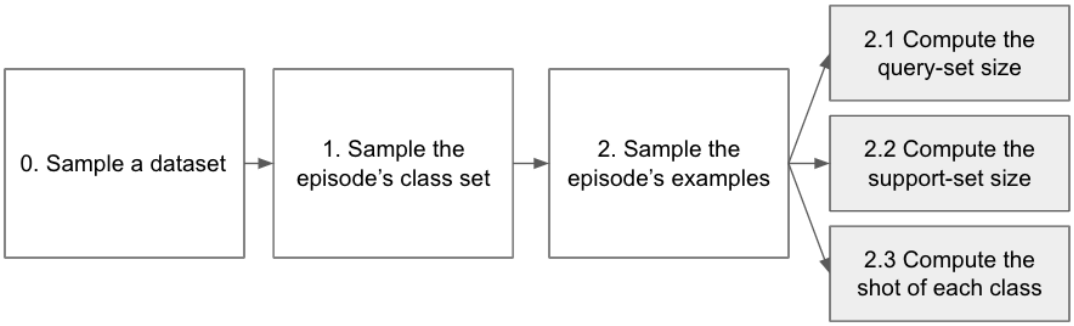


Figure 4.2 **Episode sampling of Meta-Dataset.** Firstly sample a dataset from the big collection of meta-dataset, then sample the classes of one episode, and formulate random-way-random-shot few shot learning tasks.

learning rate is set to 0.0003, beta is set as 0.9 and 0.999. Weight decay is set as 0.0001 throughout training. The number of training episodes is 10000.

Results Following [164], few-shot tasks are sampled with varying number of classes N, varying number of shots K and class imbalance. Table 4.5 reports the average rank and accuracy of each model across all 13 datasets. We also break accuracy down among the 'in-domain' and 'out-of-domain' datasets (i.e., seen/unseen during meta-training). MetaQDA has the best average rank and overall accuracy. In particular, it achieves strong out-of-domain performance, which is in line with our good cross-domain results above. Furthermore, Table 4.6 reports more detailed performance in accuracy over 600 sampled meta-test tasks. Because most of the results have very similar confidence interval, we omit this part to make the table more readable. The results of other SotA algorithms are taken from URT [94] and SCNAP[6]. From the results, we can see that MetaQDA performs well in both seen domains (left) and out-of-distribution unseen (right) domains. It achieves the highest performance in 8 of 13 domains within the Meta-Dataset benchmark.

Model	**Avg. Rank**	**Avg. Accuracy**		
	overall	**overall**	**in-domain**	**out-of-domain**
CNAP [133]	4.5	$65.9 \pm 0.8\%$	$69.6 \pm 0.8\%$	$59.8 \pm 0.8\%$
SCNAP [6]	2.9	$72.2 \pm 0.8\%$	$73.8 \pm 0.8\%$	$\mathbf{69.7 \pm 0.8\%}$
SUR [29]	3.2	$72.7 \pm 0.9\%$	$75.6 \pm 0.8\%$	$68.1 \pm 0.8\%$
URT+PN [94]	2.4	$73.7 \pm 0.8\%$	$77.2 \pm 0.9\%$	$68.1 \pm 0.9\%$
URT+MQDA	1.8	$\mathbf{74.3 \pm 0.8\%}$	$\mathbf{77.7 \pm 0.9\%}$	$68.8 \pm 0.9\%$

Table 4.5 **Few-shot classification results on Meta-Dataset.** The performance is evaluated by the classification accuracies and the rank across episodes and datasets. Gray: fixed pre-trained backbones.

Model	ImageNet	Omniglot	Aircraft	Birds	DTD	Quickdraw	Fungi	Flower	Signs	Mscoco	MNIST	CIFAR10	CIFAR100
MAML [34]	32.4	71.9	52.8	47.2	56.7	50.5	21.0	70.9	34.2	24.1	NA	NA	NA
RELATIONNET [159]	30.9	86.6	69.7	54.1	56.6	61.8	32.6	76.1	37.5	27.4	NA	NA	NA
MATCHINGNET [170]	36.1	78.3	69.2	56.4	61.8	60.8	33.7	81.9	55.6	28.8	NA	NA	NA
FINETUNE [193]	43.1	71.1	72.0	59.8	69.1	47.1	38.2	85.3	66.7	35.2	NA	NA	NA
PROTONET [150]	44.5	79.6	71.1	67.0	65.2	64.9	40.3	86.9	46.5	39.9	74.3	66.4	54.7
CNAP [133]	51.3	88.0	76.8	71.4	62.5	71.9	46.0	89.2	60.1	42.3	88.6	60.0	48.1
SCNAP [6]	**58.6**	91.7	82.4	74.9	67.8	77.7	46.9	**90.7**	73.5	46.2	93.9	74.3	60.5
SUR [29]	56.3	93.1	85.4	71.4	71.5	81.3	63.1	82.8	70.4	**52.4**	94.3	66.8	56.6
URT [94]	55.7	94.9	85.8	**76.3**	71.8	82.5	63.5	88.2	69.4	52.2	**94.8**	67.3	56.9
METAQDA	56.5	**96.3**	**86.5**	75.1	**73.4**	**82.6**	**63.7**	87.4	**73.8**	49.8	94.3	**68.2**	**57.8**

Table 4.6 **Full details of testing performance on the extended Meta-Dataset benchmark.** Left is the in-domain (seen) dataset performance, where MetaQDA ranks first 5 times in 8 domains. Right is the out-of-domain (unseen) dataset performance, where MetaQDA ranks first 3 times in 5 domains. Overall, MetaQDA outperforms other state-of-the-art models.

4.6 Further Analysis

4.6.1 Model Calibration

In real world scenarios, where high-importance decisions are being made, neural networks should indicate the probability of correctness beyond only reporting the accuracy performance [49]. Many real systems are required of high reliability, where any errors they make should be accompanied with associated low-confidence scores, e.g., so they can be checked by another process. For example, when a self-driving car algorithm cannot confidently predict the presence or absence of immediate obstructions, it should rely more on the other sensors.

Metrics Following [115, 49], we compute Expected Calibration Error (ECE) with and without temperature scaling (TS). ECE assigns each prediction to a bin that indicates how confident the prediction is, which should reflect its probability of correctness. IE: $ECE = \sum_{b=1}^{B} \frac{n_b}{N} |\text{acc}(b) - \text{conf}(b)|$, where n_b is the number of predictions in bin b, N is the number of instances, and $\text{acc}(b)$ and $\text{conf}(b)$ are the accuracy and confidence of bin b. We use $B = 20$. Temperature scaling uses validation episodes to calibrate a softmax temperature for best ECE. Please see [115, 49] for full details.

Results Table 4.7 shows MetaQDA has superior uncertainty quantification compared to existing competitors. Vanilla QDA and SimpleShot are poorly calibrated, demonstrating the importance of our learned prior. The deeper WRN is also worse calibrated despite being more accurate, but MetaQDA ultimately compensates for this. Finally, we see that our fully-Bayesian (MetaQDA-FB, Section 4.4.2) variant outperforms our MAP (MetaQDA-MAP, Section 4.4.1) variant.

Model	Backbone	ECE+TS		ECE	
		1-shot	5-shot	1-shot	5-shot
LIN.CLASSIF.	Conv-4	3.56	2.88	8.54	7.48
SIMPLESHOT	Conv-4	3.82	3.35	33.45	45.81
QDA	Conv-4	8.25	4.37	43.54	26.78
MQDA-MAP	Conv-4	2.75	0.89	8.03	5.27
MQDA-FB	Conv-4	**2.33**	**0.45**	**4.32**	**2.92**
S2M2+LIN.CLASSIF	WRN	4.93	2.31	33.23	36.84
SIMPLESHOT	WRN	4.05	1.80	39.56	55.68
QDA	WRN	4.52	1.78	35.95	18.53
MQDA-MAP	WRN	3.94	0.94	31.17	17.37
MQDA-FB	WRN	**2.71**	**0.74**	**30.68**	**15.86**

Table 4.7 **Expected calibration error (ECE) comparison on *mini*ImageNet.** Lower is better. TS indicates temperature scaling.

Model	Backbone	1-shot	5-shot
LDA	Conv-4	-	$64.24 \pm 1.42\%$
QDA	Conv-4	-	$34.45 \pm 0.67\%$
LDA (PRIOR)	Conv-4	$54.84 \pm 0.80\%$	$71.48 \pm 0.64\%$
QDA (PRIOR)	Conv-4	$54.84 \pm 0.80\%$	$71.40 \pm 0.64\%$
METALDA	Conv-4	$56.24 \pm 0.80\%$	$72.39 \pm 0.64\%$
METAQDA	Conv-4	$\mathbf{56.41 \pm 0.80\%}$	$\mathbf{72.64 \pm 0.62\%}$
LDA	WRN	-	$51.83 \pm 1.29\%$
QDA	WRN	-	$27.14 \pm 0.59\%$
LDA (PRIOR)	WRN	$63.79 \pm 0.83\%$	$81.05 \pm 0.56\%$
QDA (PRIOR)	WRN	$63.79 \pm 0.83\%$	$81.18 \pm 0.56\%$
METALDA	WRN	$64.92 \pm 0.85\%$	$83.18 \pm 0.83\%$
METAQDA	WRN	$\mathbf{67.83 \pm 0.64\%}$	$\mathbf{84.28 \pm 0.69\%}$

Table 4.8 **Comparison of different classifiers and hand-crafted vs. meta-learned prior measured on *mini*ImageNet.** We compare LDA and QDA classifiers with/without priors based on different embeddings of various backbones from shallow to deep.

4.6.2 Discussion

Why QDA but not other classifiers? In principle, one could attempt an analogous Bayesian meta-learning approach to other classifiers, but we build on discriminant analysis. This is because most classifiers do not admit a tractable Bayesian treatment, besides logistic regression (LR) and discriminant analysis. While LR has a Bayesian generalization [101], it requires approximate inference and is significantly more complicated to implement, making it difficult to extend to meta-learning. In contrast, our generative discriminant analysis approach admits an exact closed-form solution, and is easy to extend to meta-learning.

Why not other Discriminant Analysis methods? We compare how moving from LDA to QDA changes performance; and study the impact of changing from (i) no prior, (ii) hand-crafted NIW prior, and (iii) meta-learned prior. We set the hard-crafted NIW prior to $\vec{m} = 0$, $\kappa = 1$, $S = I$, and $\nu = d$ which worked well in practice. Table 4.8 demonstrates that classic unregularized discriminant analysis methods (LDA and QDA without priors) have very poor performance in the few-shot setting, due to extreme overfitting. This can be seen because: 1) the higher capacity QDA exhibits worse performance than the lower capacity LDA; and 2) incorporating a prior into LDA and QDA, thereby reducing model capacity and overfitting, results in an improvement in performance. Finally, by meta-learning the prior, we are able to optimize inductive bias for few-shot learning performance. Both LDA and QDA benefit from meta-learning, but QDA performs better overall.

Why not non-Bayesian Meta-Learning? To disentangle the impact of Bayesian modeling from our classifier architecture and episodic meta-learning procedure, we evaluate a non-Bayesian MetaQDA as implemented by performing MAML learning on the initialization of the QDA covariance factor L (Eq 4.17). From Table 4.9, we can see that MAML is worse than MetaQDA in both accuracy and calibration.

Meta Alg.	Backbone	1-shot Acc.	ECE	5-shot Acc.	ECE
MAML	Conv-4	$54.33 \pm 0.78\%$	52.75	$69.17 \pm 0.77\%$	38.84
Bayesian	Conv-4	$56.41 \pm 0.80\%$	8.03	$72.64 \pm 0.62\%$	5.27
MAML	ResNet-18	$63.66 \pm 0.80\%$	58.11	$77.82 \pm 0.62\%$	44.62
Bayesian	ResNet-18	$65.12 \pm 0.66\%$	33.56	$80.98 \pm 0.75\%$	13.86

Table 4.9 **Comparison of Bayesian vs. non-Bayesian realization of MetaQDA on *mini*ImageNet.** We compare our Bayesian implementation with MAML paradigm, and find that our model holds an obvious advantage no matter with shallow or deep backbones.

4.7 Summary

We propose an efficient shallow meta-learner for few-shot learning. MetaQDA provides a fast exact inference strategy for amortized Bayesian meta-learning through conjugacy, and highlights a distinct avenue of meta-learning research in contrast to meta representation learning. The empirical performance of our model exceeds that of others that rely on off-the-shelf feature extractors, and often outperforms those that train extractors specialised for few-shot learning. In particular, it excels in a number of challenging but highly practically benchmarks and providing accurate probability calibration – a vital property for many applications where safety or reliability is of paramount concern.

MetaQDA remains some limitations of computing efficiency, firstly the memory use of covariance is bigger than simple linear classifiers but the additional memory is still small compared to feature extractor memory. Secondly, the fully Bayesian version of MetaQDA costs much more training time than MAP version, but providing better calibration performance.

Chapter 5

Extensions of MetaQDA

MetaQDA provides a novel research direction to demonstrate the usefulness of meta-learning for few-shot recognition, which is an amortized Bayesian inference approach relying on a shallow QDA classifier with *conjugacy*. Furthermore, MetaQDA is inherently more suitable to the highly practical, but otherwise harder to achieve few-shot learning settings. The previous chapter showed that MetaQDA has achieved particularly good performance in cross-domain and multi-domain few-shot learning problems 'out of the box'. This chapter will further explore different extensions of the current MetaQDA approach to more realistic problem settings, where existing deep meta-learning algorithms cannot be generalized easily. In terms of generalized few-shot learning and few-shot class-incremental learning [161, 130], MetaQDA easily concatenates likelihood models produced by both many-shot and few-shot data via conjugacy during meta-testing, and produces a single Bayesian classifier in the joint label space. Because class conditional models are fitted independently for each class and built on fixed features, there are no forgetting problems. Furthermore, applying this procedure repeatedly also enables MetaQDA to easily overcome the class-incremental learning problem without forgetting. As for few-shot open-set recognition, we calculate the probability belonging to the novel class by marginalisation. Compared to other approaches that rely on training with auxiliary pseudo-unseen data, our approach does not require extra data, and thus saves time and computation while also providing state-of-the-art performance.

5.1 Introduction

Modern meta learning approaches can be viewed as learning the high-level concept and shared knowledge or paradigm among previously seen tasks. However, in current standard few-shot learning benchmarks, the model assumes a fixed label space defined by the support set which is not always suitable for real-world applications. From this perspective, previous

meta learning approaches may not be applied to new scenarios, e.g. predicting both seen instances and novel ones at the same time. This forms our motivation to further investigate this topic. MetaQDA can learn a prior such that, when only combined with a small amount of new data, the prediction can still maintain highly accuracy because of the introduced Bayesian inference. Its superiority to recognize novel instances can be extended to many cutting-edge scenarios, such as generalized few-shot learning, few-shot class-incremental learning, and few-shot open-set recognition.

5.1.1 Generalized Few-Shot Learning (GFSL)

Comparing to standard few-shot learning, generalized few-shot learning (GFSL) focuses on the ability to perform recognition in the joint label space of many-shot (base) and few-shot (novel) categories. However, generalized few-shot learning in the context of meta-learning is not yet a very well-studied research area with unified definition and benchmark settings.

Some previous approaches use transductive learning algorithms by applying the exemplar-based classification paradigms on both *base* and *novel* categories, so it requires recomputing the centroids during the query phase [39, 175, 53]. Some others [145, 174, 97] learn *base* and *novel* classifiers separately, ignoring the explicit relationship. Assuming that we cannot have access to *novel* classes during meta-training the model, thus the model is end-to-end learnable framework required to inductively transfer meta-knowledge from *base* class to novel classes during the meta-testing phase. *DFSLwoF* [44] is proposed in the meta-learning setup to extend the classifier to joint label space, by utilizing an attention-based weight generator for novel classes and learning with recurrent back-propagation. *CADA-VAE* [145] meta-learns a global feature encoder with the latent embedding of both image features and class embedding via aligned variational autoencoder (VAE), and trains a classifier in the joint label space. *CASTLE* [187] proposes a learning framework to synthesize calibrated few-shot classifiers in addition to the head base classifier with a shared neural dictionary. *GcGPN* [147] learns the weighted graphs embedding previously seen and novel classes into a joint prototype space, leveraging side information of inter-class relationships. *FEAT* [188] instantiates a set-to-set function with transformer, customizing task-specific embedding spaces via a self-attention architecture, to achieve fast adaptation on novel classes.

In this thesis, we shed light on meta-learning based models to solve inductive generalized few-shot learning problems. Thus, models trained on *base* categories should be capable of incorporating the limited *novel* class examples, and make predictions on both *base* many-shot categories and *novel* few-shot categories.

5.1.2 Few-Shot Class-Incremental Learning

Standard few-shot learning benchmarks provide a new task as a batch of a fixed number of support classes. However, in real world applications, we often face a growing number of novel categories, and hope to maintain prediction performance on all categories seen thus far.

Actually, it is often desirable to have the flexibility to incrementally enrol novel categories received in as a stream without forgetting the old ones. To address this problem in the context of few-shot learning, we follow the paradigm of few-shot class-incremental learning (FSCIL) [161]. The training procedure of which requires to learn novel classes from few labelled samples presented at the same time, and new classes could be added progressively. This FSCIL setting is similar to GFSL, in terms of both requiring the model to perform recognition in the joint space of all categories seen so far. However, while GFSL receives the novel categories in a single batch, FSCIL receives them incrementally over multiple training sessions.

Most existing class-incremental learning approaches focus on the general many-shot problem settings to distinguish new classes being added progressively in the growing joint label space. *iCaRL* [130] learns stronger classifiers and data representation simultaneously, *NCM* [63] and *BIC* [181] propose a bias-correction mechanism for the output to alleviate the bias between progressively added new classes and old classes. *AAM* [131] trains a set of new weights to recognize novel classes by the technique of recurrent back-propagating through the optimization process and facilitate parameter learning. Tao et al. [161] propose a *TOPIC* framework by a neural gas network to represent the knowledge preserving the topology of the feature manifold formed by different classes. We claim that our MetaQDA framework is naturally suited to this setting, and it can keep learning the prior hyperparameters along with the progressive training sessions, and therefore update the classifier with the plugged-in novel statistical data. In this condition, the fixed feature assumption reduces the risk of forgetting and saves the computing time and resources, holding more advantages than disadvantages.

5.1.3 Few-Shot Open-set Recognition

Open-set recognition (OSR) is a real-world challenge to reject when a given testing sample does not belong to any known classes, as well as maintaining the high accuracy of the known classes. Since OSR requires a large amount of datasets and FSL considers only closed-set classification and fails to recognize open samples directly, few-shot open-set recognition (FSOSR) problem in fact collects the challenges from both, and existing OSR and FSL methods may not perform well via a direct use. The main difference of FSOSR from previous

GFSL and FSCIL problem setting is the requirement to reject the open-set instances, also known as *anomaly detection*.

There are two main existing strategies to reject open-set instances. The '$N+1$'-way approach reduces open-set recognition to a traditional classification problem by viewing all novel classes/anomalies as members of a single *unknown* class [192]. This benefits from using standard supervised learning tools, but suffers from the difficulty of finding representative and diverse enough training data to train the unknown-category classifier. The difficulty of data collection limits its performance to generalize well to true anomalies. Another branch of approaches is based on standard N-way classification using confidence score, as a threshold, to reject open-set examples. This looks like the same benefit of 'N+1'-way method aforementioned, but it relies on the assumption that anomalous examples can be projected to low-confidence areas feature space, and the method may not perform well if the assumption cannot be fully satisfied [46, 12, 192, 117].

However, regardless of rejection classifier, most previous FSOSR methods [93, 113, 143] follow the pseudo-open class sample-based manner, which collects pseudo-open samples from other datasets or synthesize samples to model open-set representations. However, this approach is heavily dependent on the composition of the pseudo samples. In contrast, based on Bayesian architecture, MetaQDA provides two key benefits for outlier detection. Both are contributed by the fact that a distribution over the unknown classifier parameter can be maintained. Firstly, the Bayesian architecture leads to better calibration, as discussed in Chapter 4. This means that thresholding-based outlier detection strategies can be improved. Secondly, the Bayesian architecture leads to the ability to define a posterior probability for an instance being drawn from a known class (similarly to the N+1 classifier approach but without requiring auxiliary data) by marginalizing out the unknown classifier parameters. We also show that MetaQDA is complementary with the strategy and losses based on pseudo-unseen data proposed in the PEELER method [93], leading to state-of-the-art performance overall.

Recognition Paradigm	NO. of Samples per training class	Support UNSEEN classes in testing?
CLOSED-SET	Large	No
FEW-SHOT	Few	No
OPEN-SET	Large	Yes
FEW-SHOT OPEN-SET	Few	Yes

Table 5.1 **Comparison of different image recognition tasks.** Conventional closed-set recognition has a large dataset, but standard few-shot learning only has access to few labelled data. Conventional open-set recognition needs to support unseen detection during testing, while few-shot open-set recognition should realize anomaly detection with only a few labelled seen data.

5.2 Method

In this section, we formulate the problem of each specific real-world challenge, from generalized few-shot learning (GFSL) to few-shot class-incremental learning (FSCIL) and few-shot open-set recognition (FSOSR). We then demonstrate the framework, algorithm and pseudocode of each methodology. All the problems share the same challenge to overcome the overfitting of few-shot tasks and prevent catastrophic forgetting old tasks, but they also have different detailed settings regarding to the formulation of novel tasks. For example, (1) there is only one novel task session in each GFSL training episode, but FSCIL requires much more training sessions for the class-incremental scenario in one training episode; (2) FSOSR should reject all unseen open-set instances, whereas GFSL and FSCIL will not encounter unseen instances. In addition, the latter two should recognize both many-shot and few-shot instances in the joint label space.

5.2.1 Generalized Few-Shot Learning

The vanilla MetaQDA framework as introduced in Chapter 4 can already apply to generalized few shot learning off-the-shelf by independently processing base and novel categories. In this chapter, we also introduce *MetaQDA+* which customises the training algorithm to provide improved GFSL performance.

As for standard few-shot learning, a meta-training task is represented as N-way K-shot classification problem with N novel classes sampled from a set of support classes. However, in the setup of generalized few-shot learning, the instances seen during meta-training cannot be available during testing time. In contrast, the model is required to incorporate the novel few-shot classes into the existing space of base classes while maintaining global classification in the joint label space ($N^+ := N + N_{base}$ instead of only N). As for N^+-way K-shot generalized few-shot classification, the model is required to discriminate the query instance in the joint label space $\mathcal{Y}_{joint} = \mathcal{Y}_{base} \cup \mathcal{Y}_{novel}$, where $\mathcal{Y}_{base}$ is the previously seen training classes and $\mathcal{Y}_{novel}$ depicts the novel few-shot classes. Assume that the training dataset has many-shot base categories as the only input of the learning system. To avoid disambiguation, we formalize the task of generalized few-shot learning with both base data and novel data, denoting the data from base classes as $\mathcal{D}_{base}$, and the data from novel classes as $\mathcal{D}_{novel}$, following the definition in a standard few-shot learning. Then both base and novel data are shown as

$$\mathcal{D}_{base} = \bigcup_{n=1}^{N_{base}} \{(x_{nk}, y_n)\}_{k=1}^{K_{base}}, \mathcal{D}_{novel} = \bigcup_{n=1}^{N} \{(x_{nk}, y_n)\}_{k=1}^{K},$$

where N_{base} is the number of classes and K_{base} is the number of labelled samples available for each class. In most cases, $N_{base} \gg N$ and $K_{base} \gg K$.

Meta-training We also apply episodic meta-training for GFSL. The support set of GFSL is the same as the standard FSL with N-way few-shot novel classes, but the query set of GFSL contains a joint label space of both base and novel classes, motivating the model to maintain a generalized classification capability instead of only focusing on novel classes. During meta-training, the real novel dataset is unavailable, and the feature encoder is pre-trained and frozen on $\mathcal{D}_{base}$. Thus, in each training episode, we re-sample N *pseudo-novel* classes from the N_{base} base classes, and the remaining $N_{base} - N$ classes are the label space of the *pseudo-base* classes. We denote ϕ similar to Chapter 4.4, specifying the prior distribution over the task-specific parameters for each GFSL task. We denote the parameters of QDA classifier as $\psi = \{\mu_j, \sigma_j\}$, where in terms of standard FSL, $j = 1, 2, ..., N$, but in the case of GFSL, $j = 1, 2, ..., N, N+1, ...N+N_{base}$. Specifically, $\psi = [\psi_{pseudo-novel}, \psi_{pseudo-base}]$, where $\psi_{pseudo-base}$ is initialized as the mean and covariance of $\mathcal{D}_{base}$, and both pseudo-base classes and pseudo-novel classes share the same prior ϕ. We compute the loss gradient to update the prior parameter ϕ and classifier parameter ψ of MetaQDA+ in the GFSL setup. Intuitively, the fixed feature encoder can avoid catastrophic forgetting by nature, and Bayesian classifier can learn a joint classifier prior ϕ with generatively modelled class conditionals. After meta-training, we get the joint-space classifier prior ϕ, and set the learned $\psi_{pseudo-base}$ as the ψ_{base} for meta-testing.

Meta-testing During meta-testing, in a realistic problem setting, the model cannot get access to the original $\mathcal{D}_{base}$, but only the few-shot novel classes in $\mathcal{D}_{novel}$ are accessible. The support set of meta-testing episode is sampled from the $\mathcal{D}_{novel}$, and the query set is sampled from non-overlap instances from both $\mathcal{D}_{novel}$ and $\mathcal{D}_{base}$ in the joint label space $\mathcal{Y}_{joint}$. Facing with a query x, a GFSL model performs

$$\hat{y} = \underset{y \in \mathcal{Y}_{joint}}{\operatorname{argmax}} p_{\psi}(y|x, \mathcal{D}_{novel}), \tag{5.1}$$

where $\psi = [\psi_{novel}, \psi_{base}]$ has fixed ψ_{base} from the output of meta-training process. Base classes are completely fixed and unable to access during meta-testing in our setting. We can evaluate the performance by calculating the classification accuracy in different label spaces. The pseudocode of MetaQDA+ for GFSL is shown as Alg.2.

Discussion Prototypical Network [150, 147] could be easily adapted to GFSL problem setting in principle, where the prototypes of base classes are available and we can use the joint sets of prototypes for generalized few-shot learning. However, most existing meta-learning approaches may fall into a dilemma that either the base classes are recognized much

Algorithm 2: Pseudocode of MetaQDA+ for GFSL.

Require: Distribution over tasks Q, number of iterations T, learning rate α
`/* Meta-train */`
Input: Base Dataset $\mathcal{D}_{base}$
Init: $\phi_0 = \{\vec{m} = \vec{0}, S = \mathbf{I}, \kappa = 1, \nu = d\}$
for $t = 1$ to T **do**
  Sample N-way-K-shot pseudo-novel instances as support set from D_{base}, and the remaining is pseudo-base data, then sample query set from the joint space;
  Build QDA model on pseudo-base instances to get ψ_{p-base};
  Build Bayesian QDA model on pseudo-novel support set to get $\psi_{p-novel}$ by Eq. 4.10/ 4.11;
  Concatenate $\psi = \text{concat}[\psi_{p-novel}, \psi_{p-base}]$;
  Use query set to predict the label $\hat{y}$ in the joint label space and calculate the loss by Eq. 4.15/ 4.16;
  Back propagate to update the joint prior ϕ and ψ.
end
Output: $\phi = \{\vec{m}, S, \kappa, \nu\}$, base classifier ψ_{base}
`/* Meta-test */`
Input: Testing Base Dataset $\mathcal{D}_{testbase}$, Novel Dataset $\mathcal{D}_{novel}$, ϕ, base classifier ψ_{base}
Build the novel classifier ψ_{novel} on support set from D_{novel} by Eq. 4.10/ 4.11;
Concatenate with the fixed base classifier to get the joint classifier: $\psi_{joint} = \text{concat}[\psi_{novel}, \psi_{base}]$;
Predict the label $\hat{y}$ of query set with the joint classifier ψ_{joint} in $\mathcal{Y}_{joint}$;
Output: the predicted label $\hat{y}$ of query set

worse than a simple supervised learning, or joint space recognition is highly biased towards base classes because novel-class are rarely selected. For example, MAML [34] struggles to implement GFSL without a linear classifier layer which is the union of base and novel classes. Ideally, we want to avoid catastrophic forgetting of base classes and maintain the performance as close as possible to standard few-shot learning. *MetaQDA+* avoids these problems because the use of a fixed feature extractor prevents catastrophic forgetting; while the independent training of generative class-conditional models for base and novel classes prevents base and novel classes from destructively interfering with each other. Moreover, as a generative model *MetaQDA+* also enables tuning the trade-off between base and novel class bias via class priors.

5.2.2 Few-Shot Class-Incremental Learning

Incremental learning addresses the scenario of continually arriving data in stream instead of in batch, while the prior knowledge should be transferred without forgetting. Various previous approaches are memory-based which store the trained examples explicitly or regularize the parameter updates [11, 185]. However, since the model cannot get access to base class data within the support set of each few-shot training episode, it is challenging to learn a classifier capable to jointly classify both base and novel categories. In our setting, the model starts from a pre-trained network on a set of base classes, and during each training episode, we directly augment the classifier with learning new classifier parameters for the incremental augmented classes batch of data. Similar to generalized few-shot learning, the class-incremental learning scenario also requires the model to maintain the base and novel classification accuracy. Specifically speaking, the label space covered by novel categories increases at each episode in few-shot class-incremental scenario.

Assuming that the novel instances are coming in a streaming sequence of mini-batch labelled samples as $\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^i$, where i is the index of the i-th training novel dataset, and none of the incremental dataset shares overlap label spaces. To make the definition consistent, $\mathcal{D}^0$ is the base class dataset, which is a many-shot (large-scale) recognition task, and the following incremental novel class datasets are few-shot recognition tasks. The meta-learning model is trained incrementally within the increasing joint label space, but only the current dataset $\mathcal{D}^i$ is available as the support set for each iteration (a.k.a the training session). However, the model is tested on all encountered classes with a joint label space, in order to prevent overfitting on few-shot novel classes and avoid catastrophic forgetting on many-shot base classes.

We illustrate the details of the experiment setting in section 5.3.2, and show the outline of our meta-learning procedure of *MetaQDA+* in the pseudocode (Alg.3): (1) Firstly learn a base classifier on a set of base classes; (2) During meta-training, we have incremental training sessions to optimize an augmented pseudo-novel class classifier; (3) After meta-training, we learned and fixed the prior hyperparameters ϕ to perform joint prediction on both base many-shot and novel class-incremental few-shot classification.

Pretraining Stage: We pretrain a feature encoder of the base dataset to achieve a good representation of instance features. We also get a set of QDA classifier parameters from the base dataset to fix the initialization of the pseudo-base classifier during meta-training.

Incremental Few-Shot Meta-training Stage: We re-sample pseudo-novel classes from the original base dataset to generate the few-shot learning tasks. One meta-training episode could have several incremental sessions, and for each training session, the few-shot label

Algorithm 3: Pseudocode of meta-training MetaQDA+ for FSCIL.

Require: Distribution over tasks Q, number of incremental iterations I, number of training iterations T, learning rate α
Input: Base Dataset $\mathcal{D}_{base}$
Init: $\phi_0 = \{\vec{m} = \vec{0}, S = \mathbf{I}, \kappa = 1, \nu = d\}$
for $t = 1$ to T **do**
    Dataset is randomly split into $\mathcal{D}_{p-base}$ and I incremental $\mathcal{D}^i_{p-novel}$ datasets;
    Build QDA model on $\mathcal{D}_{p-base}$ to get $\psi^0 = \{\mu_{p-base}, \sigma_{p-base}\}$;
    for $i = 1$ to I **do**
        Build Bayesian QDA model on pseudo-novel support set from $\mathcal{D}^i_{p-novel}$ to get $\psi^i_{p-novel}$ by Eq. 4.10/ 4.11;
        Concatenate $\psi^i = \text{concat}[\psi^{i-1}, \psi^i_{p-novel}]$;
        Use query set to predict the label $\hat{y}$ in the joint label space and calculate the loss by Eq. 4.15/ 4.16;
        Back propagate to update the joint prior ϕ and ψ_i;
    end
end
Output: the prior ϕ

space should be disjoint with the base classes. For each session, we can lean a classifier on the support set and evaluate the joint prediction on query set including both base and novel classes. We iteratively meta lean the hyperparameters of priors in order to minimize the joint prediction loss on the joint query set.

Meta-Testing Stage: We meta-test the model for joint prediction in each few-shot episode, and we directly concatenate the augmented classifier with the optimized hyperparameters. As for evaluation, we show the average test accuracy of each training session, indicating the change curve of the class-incremental scenario.

5.2.3 Few-Shot Open-Set Recognition

Given that it is impossible to model every concept, open-set recognition (OSR) is ubiquitously required for a real-world visual recognition system. Furthermore, open-set recognition in combination with few-shot learning is of particular interest to support a highly practical scenario which an agent continually explores an unconstrained open-world. Novel concepts can be flagged by the open-set capability of the vision system, and subsequently annotated by a human. Annotated concepts, can subsequently be recognised by the agent using the few-shot

learning capability of the vision system. Supporting both two capabilities simultaneously is the challenging goal of few-shot open-set recognition.

In this section, we discuss how to extend MetaQDA to deal with this few-shot open-set recognition problem. Many existing approaches to OSR use auxiliary pseudo-open data to explicitly train the outlier detection model. In addition to its computational burden, the issue with this is that OSR performance depends highly on whether the auxiliary data is representative of the true open-set/anomalies encountered at testing-time, which is impossible to be guaranteed in real open-world deployments. In contrast, we will see that a key feature of MetaQDA is that it can directly provide open-set recognition without requiring any auxiliary data. We denote *MetaQDA++* as the variant of our framework extended with the open-set few-shot recognition capability. Furthermore, if representative auxiliary data can actually be available, MetaQDA can also use this data for training to further boost its performance, which is denoted as *MetaQDA++PO*.

5.2.3.1 MetaQDA++ without Pseudo-Open Data

Assume that D_K is the *known* dataset (closed-set), D_U is the *unknown* dataset (open-set). D_K should be split into training subset D_{K_tr} and testing subset D_{K_te}, sharing the same label space $\mathcal{Y}$ but non-overlapping instances. During training, we can only have access to D_{K_tr}, but during testing we need to realize closed-set classification on D_{K_te} and open-set rejection on D_U. We model each known class $C_i, i \in \mathcal{Y}$ with a *distribution* parameterized by θ_i, upon which we place a *shared prior* parameterized by ϕ. For a novel instance $\vec{x}$, a Bayesian learner should use the learned prior ϕ to determine the posterior distribution over model parameters, and predict the label of the instance to a known class $y \in \mathcal{Y}$ as

$$p(y = i|\vec{x}, \phi, D_K) = \frac{\int_{\theta_i} p(\vec{x}|\theta_i)p(\theta_i|D_K, \phi)p(y = i)d\theta_i}{p(\vec{x}|\phi)}. \tag{5.2}$$

Obviously, we can compute the likelihood of a novel instance belonging to any possible class for the standard few-shot learning, as per regular MetaQDA in Chapter 4. However, if a new sample comes from an unknown dataset D_U which we have never seen before, the model should be capable of computing the likelihood of the test instance belonging to unknown classes. The Bayesian formulation of *MetaQDA++* enables dealing with the possibility of an unknown class via the shared prior over classifier parameters $p(\theta|\phi)$. More specifically, the marginal likelihood for an instance x that may come from a known class in $i \in \mathcal{Y}$ or an unknown class $y \notin \mathcal{Y}$ is:

$$p(\vec{x}|\phi, \mathcal{D}_K) = \sum_{i \in \mathcal{Y}} \int_{\theta_i} p(\vec{x}|\theta_i) p(\theta_i|D_K, \phi) p(y = i) d(\theta_i) + p(y \notin \mathcal{Y}) \int_{\theta} p(\vec{x}|\theta) p(\theta|\phi) \cdot d\theta, \tag{5.3}$$

where $p(\theta|\phi)$ is a Normal Inverse-Wishart distribution and $p(\vec{x}|\theta)$ is a multivariate Gaussian distribution. Due to the conjugacy shown in Section 4.5, the integral in the above equation is known to be a multivariate t distribution. Thus the denominator in Bayes' rule is straightforward to evaluate, and one can therefore compute the probability of an instance belonging to an unknown class as

$$p(y \notin \mathcal{Y}|\vec{x}, \phi) = 1 - \sum_{i \in \mathcal{Y}} p(y = i|\vec{x}, \phi). \tag{5.4}$$

MetaQDA++ enables predicting the probability of an instance being in the open-set, without requiring any auxiliary data to train an explicit open-set detector, which is essential for many other methods [157, 113, 117], verifying the advantage of Bayesian mechanism in meta-learning context. Instead of training the feature extractor to estimate the unseen sample distribution, *MetaQDA++* provides efficient few-shot open-set recognition performance, without highly relying on the quality of pseudo-open configuration which may increase the training cost.

5.2.3.2 MetaQDA++ with Pseudo-Open Data

Even though our *MetaQDA++* does not require pseudo-open samples dependency for meta training, the method could still get benefits from the pseudo-open paradigm. Existing methods such as PEELER [93] widely used the re-sampled pseudo-open samples from the known dataset (closed-set). Unlike sampling the auxiliary non-overlapping data in the problem setting of OSR, our method avoids to train the feature extractor in order to obtain a better generalization ability. Actually, it is hard to achieve representative auxiliary data additional to the existing closed-set dataset, which makes the pseudo-open based approaches highly dependent on the data quality. Instead, MetaQDA could utilize off-the-shelf feature extractor of the closed training dataset, and plug-in the learnable conjugate priors of Bayesian QDA classifier to solve the few-shot open-set recognition problems. Throughout this problem setting, we interchangeably use the terms D_K and D_U to denote the known closed-set and unknown open-set, respectively. More specifically, during each meta-training episode, we randomly sample M classes as *pseudo-closed*, and the remaining classes are used as *pseudo-open*. Then each support set includes pseudo-closed $D^S_{K\_tr}$, and query set consists of both pseudo-closed and pseudo-open $D^Q_{K\_tr}$. To further improve the model performance on anomaly

detection without compulsively mapping them into a single feature cluster, MetaQDA++ uses re-sampled pseudo-open data to optimize the training of the conjugate priors.

In conventional closed-set few-shot image classification, a natural measure for the goodness of fit for ϕ is the expected log likelihood of the model plus the use of the shared prior,

$$\mathbb{E}[L(\phi|D^S_{K\_tr}, D^Q_{K\_tr})]. \tag{5.5}$$

The process of learning the prior parameters can then be formalised as a risk minimisation problem,

$$\phi^* = \underset{\phi}{arg\min}\mathbb{E}[L(\phi|D^S_{K\_tr}, D^Q_{K\_tr})]. \tag{5.6}$$

When it comes to FSOSR, the optimal model requires a suitable loss function to take advantage of the pseudo-open samples. *MetaQDA++* is able to explicitly predict the probability of an unknown query sample belonging to open-set, shown as Equation 5.5. Therefore, the out-of-distribution (OOD) anomaly detection can be viewed as another binary classification problem [65]. Combining with the traditional N-way closed few-shot learning classification task, it becomes '$N+1$'-way classification for both closed-set and open-set.

Closed-Set Loss As for known instances in query set, the closed-set classification problem is the same as conventional few-shot recognition. We can use any suitable popular loss function (e.g., cross-entropy loss) to supervise the back-propagation optimization.

$$L_{cls}(\phi|\mathcal{D}^S_{K\_tr}, \mathcal{D}^Q_{K\_tr}) = -\log \sum_{y\in\mathcal{Y}} p(y|\vec{x}, \mathcal{D}^S_{K\_tr}, \mathcal{D}^Q_{K\_tr}). \tag{5.7}$$

Rejection Loss Anomaly detection requires to reject the unknown instances which can be formalised as a binary classification problem. As a result, we can also calculate the cross-entropy loss of the unknown instances rejection as follow,

$$L_{rej}(\phi|\mathcal{D}^S_{K\_tr}, \mathcal{D}^Q_{K\_tr}) = -\log \sum_{y\notin\mathcal{Y}} p(y|\vec{x}, \mathcal{D}^S_{K\_tr}, \mathcal{D}^Q_{K\_tr}). \tag{5.8}$$

It is worth noting that the probability of an unseen open sample in the query set is calculated as Equation 5.5, which is the uniqueness of our model. *MetaQDA++* provides an explicit probability from the forward pass, then backprop to optimize the model. This classifier extends the existing closed-set N-way classifier to an '$N+1$'-way classifier, and the rejection loss is assigned to the one more shot. Hence, the loss function of *MetaQDA++* can be addressed as the combination of the above two loss functions that applied to closed

and open samples in the query set separately with weight α,

$$L_{N+1} = L_{cls} + \alpha * L_{rej}. \tag{5.9}$$

5.2.3.3 MetaQDA with Pseudo-Open Data

Similar to most existing open set recognition models, standard MetaQDA is also an N-way classifier. Unlike the large-scale setting that seen classes can be well trained with sufficient examples, the few-shot condition makes it much harder to achieve open detection. In this case, facing an unseen open sample, the model cannot assign a large probability to any known class. It should be rejected if the maximum predicted probability among all closed classes is small.

Open-Set Loss An open sample in the query set does not belong to any known classes represented by the N-way classifier, so the predicted value of each way will tell no difference. To enable this smoothing function, the learning algorithm should minimize the predicted probabilities on known classes for open samples from unknown classes, which can be implemented by maximizing the entropy of closed-set class probabilities. Following [93], we can use the negative entropy for pseudo-open data in query set,

$$L_{op}(\phi|\mathcal{D}^{S}_{K\_tr}, \mathcal{D}^{Q}_{K\_tr}) = \sum_{y\in\mathcal{Y}} p(y|\vec{x}, \mathcal{D}^{S}_{K\_tr}, \mathcal{D}^{Q}_{K\_tr}) \log \sum_{y\in\mathcal{Y}} p(y|\vec{x}, \mathcal{D}^{S}_{K\_tr}, \mathcal{D}^{Q}_{K\_tr}). \tag{5.10}$$

Hence, the loss function of MetaQDA with pseudo-open data can be addressed as the combination of closed-set loss shown as Equation 5.8 and the open-set smoothing loss with weight β,

$$L_N = L_{cls} + \beta * L_{op}. \tag{5.11}$$

5.3 Experiments

In this section, we show various experiments of extended MetaQDA models in real-world scenarios of generalized few-shot learning and open-set few-shot learning.

5.3.1 Generalized Few-Shot Learning

Dataset Split To formalize the problem setup, we take *mini*ImageNet dataset as an example. As for standard few shot learning problem, the dataset is split into train/validation/test subsets as 64/16/20 categories (600 instances of each category). We pre-train the model with *train*

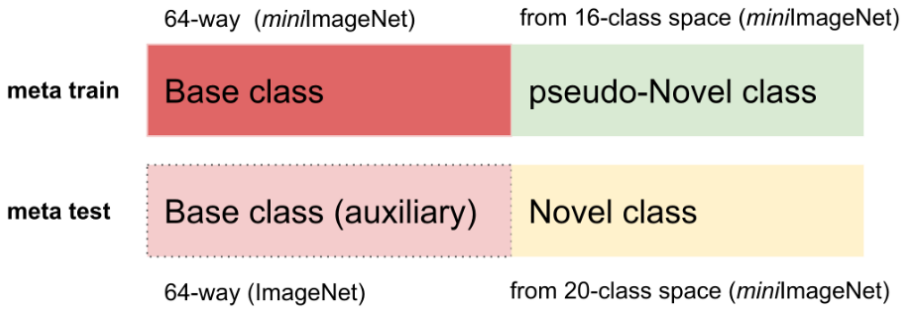


Figure 5.1 **The dataset split of *mini*ImageNet in the generalized few-shot learning scenario (GFSL).** The training *base* data are the same as standard few shot learning (red part), and the non-overlap auxiliary *base* instances are sampled from original ImageNet with the same label space (pink part). The *pseudo-novel* data in meta-training are the same as the validation dataset in standard few-shot learning (green part), and the *novel* instances of meta-testing are the same as the test dataset in standard few-shot learning (yellow part).

dataset (or both *train* and *validation* dataset, then use *train* dataset to meta-train the model, and choose the best model by *validation* dataset. During meta-test, we evaluate the model on *test* dataset, and compare our performance with other state-of-the-art approaches. However, when it comes to generalized few-shot learning as shown in Figure 5.1, we use standard train dataset for *base* classes and validation dataset for *pseudo-novel* classes during meta-training, and use auxiliary train data for *base* classes and test dataset for *novel* classes during meta-testing. The auxiliary 300 *base* instances for mate-testing are sampled from ImageNet [187] with the same label space of the base classes in meta-training.

Episodic Meta Training MetaQDA method has two phases to predict the label of query instances. The first step is getting the prior of the QDA classifier, and the second step is to use both the prior and the statistical parameters (μ, σ) to infer the classification result by Bayes' rule. Thus during meta-training, we can firstly initialize the prior with the *base* categories (64-way), and then sample the novel instances (N-way-K-shot) from the *pseudo-novel* dataset to calculate the class-conditional distribution (μ_n, σ_n) of the novel data in support set (N-way), and concatenate with the statistical parameters (μ_b, σ_b) of the base data (64-way) to get the GFSL parameters ('N^+64'-way). Specifically speaking, if standard 5-way-1-shot few-shot learning has 5-way-15-shot query instances each episode, then generalized few-shot learning has 64-way-15-shot *base* instances and 5-way-15-shot *novel* instances to compose 69-way-15-shot query instances in each episode for GFSL.

Meta Testing During meta-testing, we sample novel instances (e.g. 5-way-1-shot) from the *novel* dataset (yellow part) as support set, and sample *novel* instances (non-overlap 15-shot of the same 5-way) and *base* instances (e.g. 64-way-15-shot) from the auxiliary base dataset

(pink part) as query set. After meta-training, we acquire the trained prior of the MetaQDA model, then we use the prior and the support set to get the class conditional distribution of novel classes by conjugacy, and concatenate it with the pre-trained class conditional distribution of base classes to produce a GFSL classifier by Bayes' rule (69-way).

Evaluation Metric Following previous meta-learning research in GFSL [147], we also evaluate the model and calculate the average accuracy with 95% confidence interval over 600 random test episodes, while each episode is composed of all base classes plus N-way novel categories.We evaluate the GFSL performance by 5 metrics following [147], such as *Base-Base, Novel-Novel, Base-Joint, Novel-Joint, Joint-Joint*, where the item before the hyphen is the query data used, and the item after the hyphen is the label space considered for classifying the specified data. To penalize unsatisfactory performance in either metric, we also report the harmonic mean to balance the unequal sizes of base and novel data.

Results We compare our methodology with some other baselines: *PN+* [150], *DFSLwoF* [44], and *GcGPN* [147]. Specifically, Prototypical Network could be straightly extended to *PN+* to satisfy the memorization of base classes, the details are shown in [147]. Then we evaluate our model on *mini*ImageNet both for 1-shot and 5-shot of '5^+base'-way classification. As shown in 5.2, *MetaQDA+* is capable of both achieving excellent few-shot recognition accuracy on novel categories and maintaining high performance on base categories, surpassing prior state-of-the-art approaches.

Model		**FSL**		**GFSL**		
1-shot	**Base-Base**	**Novel-Novel**	**Base-Joint**	**Novel-Joint**	**Joint-joint**	**H-mean**
PN+ [150]	54.02 ± 0.46%	53.88 ± 0.78%	54.02 ± 0.46%	0.02 ± 0.01%	27.02 ± 0.23%	0.04 ± 0.03%
DFSLwoF [44]	69.93 ± 0.41%	55.80 ± 0.78%	58.54 ± 0.43%	40.30 ± 0.74%	49.42 ± 0.41%	46.95 ± 0.55%
GcGPN-as [147]	68.13 ± 0.43%	60.40 ± 0.71%	54.68 ± 0.46%	48.59 ± 0.72%	51.63 ± 0.41%	50.83 ± 0.45%
MetaQDA	64.25 ± 0.18%	56.41 ± 0.80%	59.47 ± 0.18%	5.08 ± 0.82%	55.52 ± 0.16%	9.25 ± 0.58%
MetaQDA+	68.66 ± 0.15%	56.35 ± 0.80%	61.58 ± 0.16%	45.25 ± 0.79%	**58.22 ± 0.13%**	**52.65 ± 0.55%**
5-shot	**Base-Base**	**Novel-Novel**	**Base-Joint**	**Novel-Joint**	**Joint-joint**	**H-mean**
PN+ [150]	60.42 ± 0.45%	70.84 ± 0.66%	60.41 ± 0.45%	2.99 ± 0.20%	31.70 ± 0.25%	5.54 ± 0.34%
DFSLwoF [44]	70.24 ± 0.43%	72.59 ± 0.62%	59.89 ± 0.47%	58.26 ± 0.68%	59.08 ± 0.40%	58.58 ± 0.41%
GcGPN-as [147]	68.30 ± 0.45%	73.31 ± 0.62%	57.93 ± 0.48%	59.32 ± 0.68%	58.63 ± 0.40%	56.69 ± 0.41%
MetaQDA	65.98 ± 0.12%	72.64 ± 0.62%	61.72 ± 0.12%	10.97 ± 0.58%	58.04 ± 0.15%	16.39 ± 0.45%
MetaQDA+	68.67 ± 0.12%	72.08 ± 0.65%	62.74 ± 0.13%	55.19 ± 0.67%	**62.19 ± 0.13%**	**59.73 ± 0.42%**

Table 5.2 **Generalized few-shot learning results on *mini*ImageNet.** This table demonstrates the average accuracy and harmonic mean to show the joint-joint performance, which is the main objective of GFSL. Harmonic mean is calculated by the Base-Joint and Novel-Joint accuracies. The results are evaluated on testing set both for 1-shot and 5-shot of 5^+base-way classification. Note that GcGPN uses Conv4-128 backbone, but DFSLwoF and ours use Conv4-64 backbone. *MetaQDA* means that using vanilla MetaQDA trained by standard FSL benchmark directly, which is easily overfitted to base categories. *MetaQDA+* means the extended model shown in GFSL methodology.

5.3.2 Few-Shot Class-Incremental Learning

Dataset Split Following [161], *mini*ImageNet is split into 60 *Base* classes and 40 *Novel* classes, each with 500 training instances and 100 testing instances, respectively. Each episode starts from a base classifier and proceeds in 8 learning sessions by adding a 5-way-5-shot support set per session. After each session, models are evaluated on the full set of classes seen so far, leading to a 100-way generalized few-shot problem in the 9^{th} session (session 8), as shown in Figure 5.2. During the meta training process, only the 60 *Base* class training dataset is available, then we re-sample them to pseudo-base and pseudo-novel classes to get the trained prior of MetaQDA, and directly deploy it on the incremental meta-testing process.

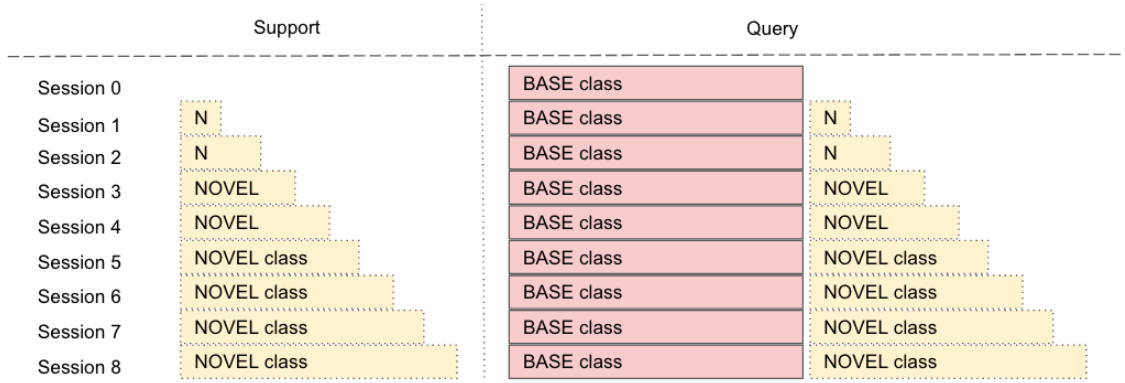


Figure 5.2 **One meta-test task episode of *mini*ImageNet in the few-shot class-incremental learning (FSCIL) scenario.** Base class (60-way) only appears in the query set (in pink) and the category number of novel class increases 5-way per training session (in yellow). Note that base instances are from the testing dataset disjoint with the meta-training dataset.

Implementation Detail As per [161], we pre-train a ResNet18 backbone and then meta-train *MetaQDA*+ model on 60 base classes before performing incremental meta testing. We use stochastic gradient descent (SGD) with the initial learning rate of 0.1, decreasing the learning rate to 0.01/0.001 after 30/40 epochs, respectively, and mini-batch size is 128. The *MetaQDA*+ prior is not updated during meta-testing. *Meta-Training*: We adopt the 5-way-5-shot few-shot paradigm, and we have 9 training sessions in total, while each incremental session is constructed by randomly picking 5 training samples per class from the original dataset. The *MetaQDA*+ prior is then trained by generating sequential (multi-session) episodes from the 60 base class set, using the feature extractor trained as above. *Meta-Testing*: Due to our Bayesian class-conditional modeling, meta-testing decomposes over classes. Class-incremental learning is thus trivially realized by running the update step of *MetaQDA*+ for each new category, and adding the final mean and covariance to the set used by the final QDA

Model	AL_MML [161]	NCC	METAQDA+	Margin
session 0 (60)	**61.31**	46.62	59.57	-
session 1 (65)	50.09	43.26	**54.98**	(+4.89)
session 2 (70)	45.17	40.87	**51.06**	(+5.89)
session 3 (75)	41.16	39.04	**47.69**	(+6.53)
session 4 (80)	37.48	37.50	**44.71**	(+7.23)
session 5 (85)	35.52	35.96	**42.08**	(+6.56)
session 6 (90)	32.19	34.13	**39.74**	(+7.55)
session 7 (95)	29.46	33.19	**37.66**	(+8.20)
session 8 (100)	24.42	32.26	**35.78**	(+11.36)

Table 5.3 **Class-incremental few-shot learning results on *mini*ImageNet.** Start with 60-way base classifier and add 5-way-5-shot per session. At each session, the models are evaluated on the test sets of the full set of classes encountered so far. All models in the table use ResNet-18 backbone. (#): classifier-way at each session.

classifier. We apply *MetaQDA+* both for the many-shot base classes, and 5-shot incrementally increasing novel classes.

Results We depict the experiment results in terms of accuracy and session number shown in Table 5.3, noting that the number of categories increases in the label space over iterations from session 0 (60-way) to session 8 (100-way). Specifically, the accuracy is the average number generated by independently repeating both meta-train and meta-test (8 incremental sessions each) phases 10 times with random 5-shot episodes. It shows that *PN* and *NCC* could outperform the other State of the art baselines with big session number, which requires to overcome the challenge of forgetting. Clearly *MetaQDA+* also mitigates the forgetting of old classes and improves the few-shot learning of new classes in the class-incremental scenario due to the Bayesian classifier, significantly show the advantage of our model over other baselines [161].

5.3.3 Few-Shot Open-Set Recognition

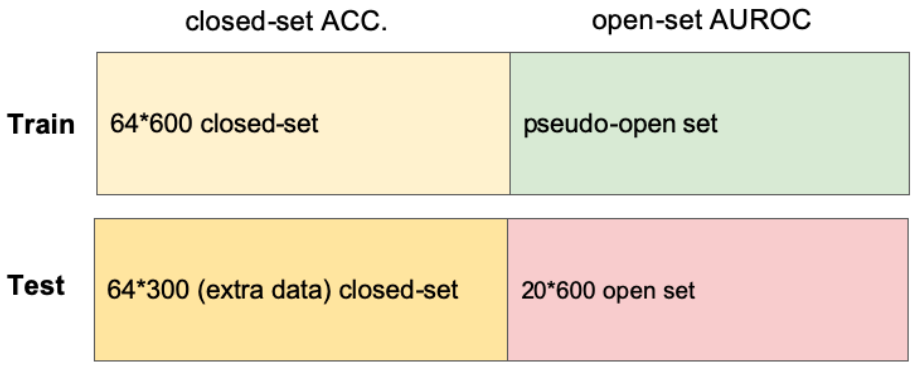
Dataset Split Following [93], we use *mini*ImageNet to evaluate the algorithm performance on both closed-set classification and open-set detection performance. Similar to standard few-shot benchmark, *mini*ImageNet dataset is split into 64 classes for training, 16 classes for validation and 20 classes for testing. Because the support set could not have access to open-set, so our support set only includes N-way-K-shot closes set samples, but the query set contains both closed classes and also open-set samples. *MetaQDA++* can be implemented without meta-training by pseudo-open data, and both *MetaQDA++* and MetaQDA can be

implemented with re-sampled pseudo-open data to further train the anomaly detector. Here we show the dataset split with pseudo-open data following [93]. To clarify the difference of pseudo-open data between large-scale and few-shot open set recognition, we have an illustrative visualisation of OSR and FSOSR shown as Figure 5.3. It is noting that large scale open set recognition does not use meta-learning paradigm, and to guarantee the testing samples non-overlap with the training samples, we need to sample extra closed-set samples from original ImageNet. However, when it comes to FSOSR, we follow the 5-way few-shot open-set recognition paradigm as [93] to guarantee the fair comparison. To make the balance of closed-set and open-set for evaluation, we only re-sample N-way pseudo-open classes for the query set. Specifically speaking, in each meta training episode, 5 classes are randomly selected as pseudo-closed set and 5 other classes are randomly selected as pseudo-open set. For both meta training and meta testing tasks, all support samples are selected from the closed-set, but the query set samples are selected from both closed and open sets.

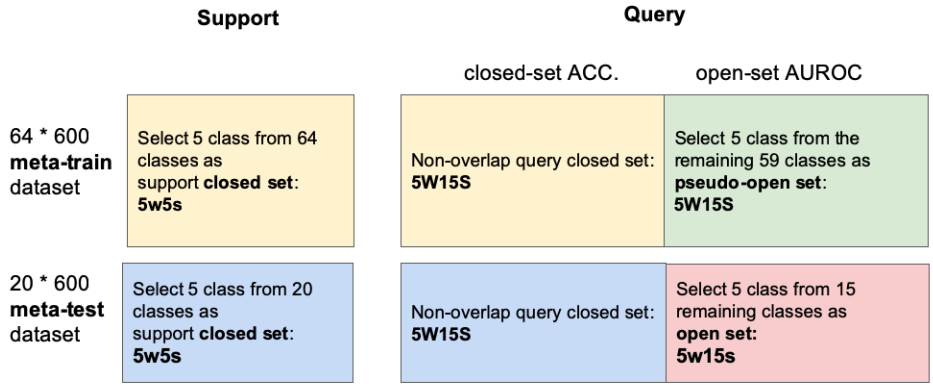
Implementation Detail Similar to previous methods, we pre-train a ResNet18 backbone on the known base classes. We meta-train *MetaQDA++* by stochastic gradient descent (SGD) with the initial learning rate of 0.01, decreasing the learning rate to 0.001 after 40 epochs, and mini-batch size is 128. As for the scenario with pseudo-open set, α is set to 0.3 for *MetaQDA++PO* and β is set to 0.5 for MetaQDA.

Evaluation Metric Following Neal et al. [113], we use the conventional closed-set classification accuracy and open-set detection AUROC (Area Under ROC Curve) to evaluate open-set recognition performance. For test images drawn from known categories, we evaluate the conventional N-way multi-class accuracy as per standard closed-set few-shot learning. Considering all test images including both open- and closed-set examples, *MetaQDA++* performs outlier detection as a binary classification problem between known and unknown categories, which is explicitly assigned by the '$N+1$'-way classifier. MetaQDA with open-loss performs outlier detection by the entropy of an N-way classifier prediction, similar to [93]. We report the outlier detection performance in terms of AUROC. A good open-set classifier should not sacrifice the standard known-class recognition performance, so we calculate the *closed-set accuracy* applying to only the known classes with open-set detection disabled.

Results As for few shot open set recognition, we compare different experiment settings to show the effectiveness of different components of our methodology. Specifically, we evaluate *MetaQDA++* with and without pseudo-open data meta-training, and compare standard MetaQDA with pseudo-open data. Our baselines are including some previous large-scale open-set recognition approaches which can be applied to few-shot scenario, shown in [93]. To disambiguate the confusion of various components adding to the original algorithm, we only report the best results obtained by previous methods. Thus the baselines in our experiments



(a)



(b)

Figure 5.3 **Illustrative visualization of the paradigm of open-set recognition (OSR) and few-shot open-set recognition (FSOSR) on *mini*ImageNet.** (a) Large-scale open set recognition on *mini*ImageNet with pseudo-open data. (b) Few-shot open-set recognition on *mini*ImageNet without requiring extra data, but only re-sample to get the pseudo-open data.

are: *ProtoNet* [150], *FEAT* [188], *OpenMax* [143], *Counterfactual* [113], and *PEELER* [93]. Our Bayesian paradigm could obviously hold the advantage to balance the closed-set and open-set out-of-distribution detection at the same time. The first line in grey shade is our *MetaQDA++* with a vanilla Bayesian outlier detector without re-sampling pseudo-open data during meta-training process, which has already outperformed the previous state-of-the-art result [93] with a large margin. Our standard MetaQDA is designed for standard few-shot learning, but it can also get improved by being re-trained with pseudo-open data. In complementary, we use the pseudo-open data to train the *MetaQDA++*, whereupon the margin over competitors increases.

Model	Pseudo.	OpLoss	5-way-1-shot		5-way-5-shot	
			Acc (%)	AUROC (%)	Acc (%)	AUROC (%)
PROTONET [150]	✓		64.01 ± 0.88%	51.81 ± 0.93%	80.09 ± 0.86%	60.39 ± 0.77%
FEAT [188]	✓		67.02 ± 0.85%	57.01 ± 0.84%	82.02 ± 0.77%	63.18 ± 0.83%
OPENMAX [143]	✓		57.89 ± 0.59%	58.92 ± 0.59%	75.31 ± 0.76%	67.54 ± 0.67%
COUNTER. [113]	✓		57.89 ± 0.59%	52.20 ± 0.61%	75.31 ± 0.76%	63.25 ± 0.59%
PEELER [93]	✓		56.31 ± 0.57%	58.94 ± 0.60%	74.19 ± 0.75%	66.00 ± 0.67%
PEELER [93]	✓	✓	58.31 ± 0.58%	61.66 ± 0.62%	75.08 ± 0.72%	69.85 ± 0.70%
METAQDA	✓	✓	60.45 ± 0.68%	63.55 ± 0.65%	76.22 ± 0.86%	70.18 ± 0.78%
METAQDA++			62.75 ± 0.62%	69.08 ± 0.85%	79.97 ± 0.66%	75.49 ± 0.71%
METAQDA++	✓		**64.25 ± 0.58%**	**70.08 ± 0.83%**	**80.47 ± 0.58%**	**81.33 ± 0.68%**

Table 5.4 **Few-shot open-set recognition results on *mini*ImageNet.** Average closed-set accuracy and open-set AUROC are shown on both 1-shot and 5-shot of 5^+-way few-shot open-set recognition experiments. We use ResNet-18 backbone for fair comparison.

5.4 Summary

In this chapter, we explored extending and applying MetaQDA to various more realistic application scenarios, and empirically demonstrated that MetaQDA provided excellent performance across various settings and evaluation metrics including generalized, class-incremental, and open-set few-shot recognition. We shed light on the superiority of our method due to a combination of fixed feature representation and a Bayesian meta-learned classifier. It is obvious that the fixed features avoid catastrophic forgetting by nature, and the Bayesian classifier with generatively modelled class conditionals further alleviates the forgetting problem in contrast to discriminately learned decision boundaries. Consequently, GFSL with class imbalance is easier to solve, and class-incremental scenario is implemented by adding new class conditionals without modifying the existing a class-conditional models. Finally, our Bayesian classifier also makes outlier detection easier by providing a direct estimate of unknown class probability through marginalising out the unknown classifier parameters.

Even though using fixed features is an advantage in many ways as discussed above, it provides a limitation in other ways. If (meta)-tested on images with very different statistics from the meta-training data (e.g. from *ImageNet* to medical images), the performance of MetaQDA is likely to be worse because the feature extractor does not update in response to meta-test images. While this can be ameliorated with strong multi-domain features as shown in Chapter 4.6.5, it remains doubtful to provide a sufficient solution in general, or if new methods with end-to-end updates during meta-test will ultimately be required.

Chapter 6

Conclusions and Future Work

This thesis studied few-shot meta-learning in computer vision. We started by focusing on deep learning architectures to improve metric-based meta-learning in the form of DCN/ RelationNet2. Subsequently we moved on to more general model-based meta-learning in the form of MetaQDA, where we focused on Bayesian meta-learning approaches to a final classifier layer, while being agnostic to the deep architecture used. Over the course of the thesis we also moved from addressing the most widely studied but least realistic C-way-K-shot style academic benchmarks, to addressing more practically realistic and valuable problem settings such as generalized, class-incremental, and open-set few-shot learning. The vast majority of existing academic research has focused on the narrowly defined C-way-K-shot problem for few-shot learning. While this has helped the core technology to advance, their restrictive assumptions limit them to hardly benefit some main potential use-cases of few shot learning technology such as autonomous agents that need to learn from each newly encountered object online [98, 158]. We hope that the broader contributions of our work in incremental generalized few-shot learning and novelty detection could help make such autonomous agent applications a reality.

6.1 Contributions

The main contributions of this thesis are the investigation of using meta-learning methodologies to entangle few-shot learning and open-ended image recognition challenges.

(1) We propose a parameterized metric-learning approach, RelationNet2, as a matching framework comprised of embedding and relation modules, learning multiple non-linear comparisons simultaneously corresponding to multiple levels of extracted features.

(2) We defend the meta-learning role in various few-shot learning scenarios by introducing a shallow meta-learner, which utilizes an amortized Bayesian quadratic discriminant analysis

through conjugacy. This method outperforms the others relying on off-the-shelf feature extractors, and even exceeds those training deep features specialized on few-shot learning with higher computation cost.

(3) Our methods achieve state-of-the-art performance on the standard few-shot learning benchmarks, retaining the simplicity and effectiveness of meta-learning pipelines. Furthermore, our RelationNet2 reduces the overfitting by adding Gaussian noise regularization, and MetaQDA avoids overfitting to a few-shot support set through use of a prior on the parameters which prevents them from overfitting to a few-shot support set.

(4) Faced with real-world applications, our MetaQDA is inherently suited to highly practical but more challenging tasks, with the probability calibration of the model being especially critical. The Bayesian paradigm with MetaQDA's efficient fixed-feature learning performs excellently across various settings from cross/multi-domain and class-incremental to open-set recognition.

(5) We make a major contribution to the recent debate in few-shot learning for computer vision: "Is meta-learning really helpful or not, given improving techniques for training basic features?" Our results show that the answer is Yes. Meta-learning can benefit few shot learning even for fixed pre-provided features, simply by performing suitably designed meta-learning at the classifier layer.

(6) We evaluate our models by both classification accuracy and calibration performance, and also calculate the AUROC of the open-set settings. Experiments are conducted on various benchmarks, such as the widely-used *mini*ImageNet and *tiered*Imagenet, and also the fine-grained CUB and Cars, showing our approaches can achieve stable state-of-the-art performances with less uncertainty.

6.2 Limitations

In this thesis, we have already explored the pipelines of meta-learning and contributed an impressive promotion to various computer vision applications with data scarcity. However, after many academic endeavors, we have still come across several limitations. This section offers a discussion about the disadvantages of current approaches and potential future directions.

Lack of Scalability Even though our RelationNet2/DCN achieves state-of-the-art performance on standard few-shot learning benchmarks, e.g., *mini*ImageNet and *tiered*ImageNet, it has scalability limitation of additional multi-metrics computing. When it comes to a large support set and a bigger number of classes, the non-linear growth of parameters will accelerate the computing cost based on deep backbones with enormous parameters. Also, all experiments in Chapter 3 are conducted on standard few shot learning benchmarks, but when

it comes to more realistic applicants, where the agent needs to classify both novel few-shot and seen many-shot classes, a.k.a, generalized few shot learning, the model could hardly achieve a competitive performance without forgetting the previous seen classes.

Memory and Computing Cost MetaQDA utilizes a QDA classifier, of which the covariance matrix increases the memory cost from linear level to square level. Wider network architectures are infeasible to enhance the performance of the current model, thus we need to limit the extracted feature dimension to a reasonable range. Meanwhile, quadratic classifier requires more computing cost than linear classifier to update the model, even though our conjugacy-based methods are much faster than other backdrop-based methods. We should design more effective dimensionality reduction algorithm to augment the capability of MQDA.

Generalization for Out-of-Distribution Datasets As discussed in Chapter 5, current few shot learning assumes sampling from a pre-defined distribution of tasks, which in practice requires laborious human engineering, but we hope to design models with high generalization ability to diverse realistic scenarios. Despite the advantage of using off-the-shelf features in MetaQDA, it seems that meta testing on images out of distribution (OOD) is challenging for current MetaQDA without back propagation. To enhance the generalization ability in the future, we may need stronger features, e.g., multi-domain features trained from meta-dataset, to provide a more general solution.

6.3 Future Work

In this section, we would like to discuss some immediate extensions of the current meta-learning research to overcome the previous discussed limitation, and some potential promising directions of future work in both the academy and the industry.

Deep Bayesian Meta-Learning Our current work only exploits Bayesian meta-learning at the classifier layer, performing prior-posterior updates on the classifier distribution given the support set, but treats the feature extractor as the black box. Future work could investigate whether performance could be improved by end-to-end Bayesian meta-learning to train the optimal Bayesian prior for every weight in the deep network. In this more general case, simple and efficient conjugacy-based updates would likely no longer be possible. Developing a tractable approach, e.g., by drawing on tools such as amortized variational inference [70, 128], would be the key challenge.

Multi-Domain Meta-Learning Multi-domain meta-learning is an outstanding challenge in this field. While MetaQDA performs well when leveraging a multi-domain pre-trained

feature, shown in Chapter 4, it misses an opportunity to exploit a statistical model of the situation, which would likely improve results further. Single domain MetaQDA corresponds to learning a uni-modal NIW prior on the QDA classifier. In the multi-domain case (e.g., given a mixture of everyday, medical and astronomical images), the relevant statistical model could be a mixture-model prior on the QDA classifier. Then the meta-test step would involve estimating the relevant mixture component and using that component to update the classifier.

Lifelong Meta Learning Due to the conjugacy, MetaQDA can perform efficient and non-forgetting class-incremental and instance-incremental few-shot learning out of the box, and thus could be described as 'meta-learning a lifelong learner'. However it is not truly a lifelong meta-learner in that the globally shared meta-knowledge (prior parameters) are not updated after the meta-train stage. In a lifelong learning setting, once the number of new meta-test tasks encountered becomes large compared to the initial number of meta-train tasks, MetaQDA misses an increasing opportunity for cross-task knowledge transfer. Developing a lifelong meta-learner remains an outstanding challenge.

Few-Shot Learning beyond Object Recognition Most existing work on few shot learning focuses on simple object recognition. This only scratches the surface of the applications within vision where few-shot learning is necessary, but ignoring other applications include for example relationship detection in scene graph recognition. We did some preliminary studies on this topic but not covered in the thesis, unfortunately lack enough time to finish it. Visual relationships in a scene graph are represented as subject-predicate-object triplet $< S, P, O >$, providing a more powerful query modality than simple image tags. However, the predicates in a scene graph meet long-tailed distribution and have a large label space with incomplete annotation, so few-shot predicate classification problem setting has more realistic meaning. Datasets that have a long tail of relationships with few-shot samples, e.g, Visual Genome (VG) only focuses on 50 frequent predicates, which were ignored in previous researches. Semantic segmentation models tend to provide specialised architectures for performing image-to-image prediction. However, there is still usually a conventional classification layer applied pixel-wise to get the unary potential of each pixel belonging to a given category. These pixel-wise classifiers could be upgraded from conventional linear classifiers to meta-learned MetaQDA classifiers. More generally, because MetaQDA does not focus on any particular neural network architecture, it is in no way specific to computer vision. The ideas in MetaQDA could be applied to any supervised classifier learning application where there are task families and new tasks required to be solved with few examples, such as ASR (Auto Speech Recognition) [100], TTS (Text To Speech) [195].

Bibliography

[1] Han Altae-Tran et al. “Low data drug discovery with one-shot learning”. In: *ACS central science* 3.4 (2017), pp. 283–293.

[2] Nassim Ammour et al. “Deep learning approach for car detection in UAV imagery”. In: *Remote Sensing* 9.4 (2017), p. 312.

[3] Marcin Andrychowicz et al. “Learning to learn by gradient descent by gradient descent”. In: *NeurIPS*. 2016.

[4] Antreas Antoniou, Harrison Edwards, and Amos Storkey. “How to train your MAML”. In: *ICLR*. 2018.

[5] Caglar Aytekin et al. “Clustering and unsupervised anomaly detection with l2 normalized deep auto-encoder representations”. In: *IJCNN*. 2018.

[6] Peyman Bateni et al. “Improved Few-Shot Visual Classification”. In: *CVPR*. 2020.

[7] Jonathan Baxter. “A Bayesian/information theoretic model of learning to learn via multiple task sampling”. In: *Machine learning* 28.1 (1997), pp. 7–39.

[8] Jonathan Baxter. “A model of inductive bias learning”. In: *Journal of Artificial Intelligence Research* 12 (2000), pp. 149–198.

[9] Atilim Gunes Baydin et al. “Online learning rate adaptation with hypergradient descent”. In: *CoRR* abs/1703.04782 (2017).

[10] Irwan Bello et al. “Neural optimizer search with reinforcement learning”. In: *ICML*. 2017.

[11] Eden Belouadah and Adrian Popescu. “Il2m: Class incremental learning with dual memory”. In: *CVPR*. 2019.

[12] Abhijit Bendale and Terrance E Boult. “Towards open set deep networks”. In: *CVPR*. 2016.

[13] Luca Bertinetto et al. “Learning feed-forward one-shot learners”. In: *NeurIPS*. 2016.

[14] Luca Bertinetto et al. “Meta-learning with differentiable closed-form solvers”. In: *ICLR*. 2019.

[15] Christopher M Bishop. *Mixture density networks*. Tech. rep. Aston University, 1994.

[16] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent dirichlet allocation”. In: *Journal of machine Learning research* 3 (2003), pp. 993–1022.

[17] Mariusz Bojarski et al. “End to end learning for self-driving cars”. In: *CoRR* abs/1604.07316 (2016).

[18] Francisco M Castro et al. “End-to-end incremental learning”. In: *ECCV*. 2018.

[19] Xiaobin Chang, Timothy M. Hospedales, and Tao Xiang. "Multi-level factorisation net for person re-identification". In: *CVPR*. 2018.

[20] Wei-Yu Chen et al. "A closer look at few-shot classification". In: *ICLR*. 2019.

[21] Zhiyuan Chen and Bing Liu. "Lifelong machine learning". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12.3 (2018), pp. 1–207.

[22] Paul Covington, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations". In: *ACM Recommender Systems*. 2016.

[23] Gabriela Csurka. *Domain adaptation in computer vision applications*. Springer International Publishing, 2017.

[24] Aminu Da'u and Naomie Salim. "Recommendation system based on deep learning methods: a systematic review and new directions". In: *Artificial Intelligence Review* 53.4 (2020), pp. 2709–2748.

[25] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *CVPR*. 2009.

[26] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *NAACL*. 2019.

[27] Yan Duan et al. "One-shot imitation learning". In: *NeurIPS*. 2017.

[28] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. "Diversity with cooperation: Ensemble methods for few-shot classification". In: *ICCV*. 2019.

[29] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. "Selecting relevant features from a multi-domain representation for few-shot classification". In: *ECCV*. 2020.

[30] Harrison Edwards and Amos Storkey. "Towards a neural statistician". In: *ICLR*. 2017.

[31] Desmond Elliott. "Adversarial evaluation of multimodal machine translation". In: *EMNLP*. 2018.

[32] Li Fe-Fei, Fergus, and Perona. "A Bayesian approach to unsupervised one-shot learning of object categories". In: *ICCV*. 2003.

[33] Li Fei-Fei, Rob Fergus, and Pietro Perona. "One-shot learning of object categories". In: *TPAMI* 28.4 (2006), pp. 594–611.

[34] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for fast adaptation of deep networks". In: *ICML*. 2017.

[35] Chelsea Finn, Kelvin Xu, and Sergey Levine. "Probabilistic Model-Agnostic Meta-Learning". In: *NeurIPS*. 2018.

[36] Luca Franceschi et al. "Forward and reverse gradient-based hyperparameter optimization". In: *ICML*. 2017.

[37] Robert M French. "Catastrophic forgetting in connectionist networks". In: *Trends in cognitive sciences* 3.4 (1999), pp. 128–135.

[38] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[39] Hang Gao et al. "Low-shot learning via covariance-preserving adversarial augmentation networks". In: *NeurIPS*. 2018.

[40] Victor Garcia and Joan Bruna. “Few-shot learning with graph neural networks”. In: *ICLR*. 2018.

[41] Marta Garnelo et al. “Conditional neural processes”. In: *ICML*. 2018.

[42] Weifeng Ge et al. “Deep metric learning with hierarchical triplet loss”. In: *ECCV*. 2018.

[43] ZongYuan Ge et al. “Generative Openmax for multi-class open set classification”. In: *BMVC*. 2017.

[44] Spyros Gidaris and Nikos Komodakis. “Dynamic few-shot visual learning without forgetting”. In: *CVPR*. 2018.

[45] Micah Goldblum et al. “Unraveling meta-learning: understanding feature representations for few-shot tasks”. In: *ICML*. 2020.

[46] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”. In: *ICLR*. 2015.

[47] Jonathan Gordon et al. “Meta-learning probabilistic inference for prediction”. In: *ICLR*. 2019.

[48] Erin Grant et al. “Recasting gradient-based meta-learning as hierarchical Bayes”. In: *ICLR*. 2018.

[49] Chuan Guo et al. “On calibration of modern neural networks”. In: *ICML*. 2017.

[50] Jianzhu Guo et al. “Learning meta face recognition in unseen domains”. In: *CVPR*. 2020.

[51] Yunhui Guo et al. “A broader study of cross-domain few-shot learning”. In: *ECCV*. 2020.

[52] Abid Haleem, Mohd Javaid, and Ibrahim Haleem Khan. “Current status and applications of Artificial Intelligence (AI) in medical field: An overview”. In: *Current Medicine Research and Practice* 9.6 (2019), pp. 231–237.

[53] Bharath Hariharan and Ross Girshick. “Low-shot visual recognition by shrinking and hallucinating features”. In: *ICCV*. 2017.

[54] Bharath Hariharan et al. “Hypercolumns for object segmentation and fine-grained localization”. In: *CVPR*. 2015.

[55] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[56] Kaiming He et al. “Deep residual learning for image recognition”. In: *CVPR*. 2016.

[57] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *ICCV*. 2015.

[58] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. “Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem”. In: *CVPR*. 2019.

[59] Tom Heskes. “Empirical Bayes for learning to learn”. In: *ICML*. 2000.

[60] Nathan Hilliard et al. “Few-shot learning with metric-agnostic conditional embeddings”. In: *CoRR* abs/1802.04376 (2018).

[61] Geoffrey E Hinton and David C Plaut. “Using fast weights to deblur old memories”. In: *CCSS*. 1987.

[62] Timothy M Hospedales et al. “Meta-learning in neural networks: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[63] Saihui Hou et al. “Learning a unified classifier incrementally via rebalancing”. In: *CVPR*. 2019.

[64] Rein Houthooft et al. “Evolved policy gradients”. In: *NeurIPS*. 2018.

[65] Yen-Chang Hsu et al. “Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data”. In: *CVPR*. 2020.

[66] Jie Hu, Li Shen, and Gang Sun. “Squeeze-and-Excitation Networks”. In: *CVPR*. 2018.

[67] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. “Discriminative deep metric learning for face verification in the wild”. In: *CVPR*. 2014.

[68] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. “Leveraging the feature distribution in transfer-based few-shot learning”. In: *CoRR* abs/2006.03806 (2020).

[69] Gao Huang et al. “Densely connected convolutional networks.” In: *CVPR*. 2017.

[70] Ekaterina Iakovleva, Jakob Verbeek, and Karteek Alahari. “Meta-learning with shared amortized variational inference”. In: *ICML*. 2020.

[71] Andrey Ignatov et al. “AI Benchmark: All about deep learning on smartphones in 2019”. In: *CoRR* abs/1910.06663 (2019).

[72] Xiang Jiang et al. “Learning to learn with conditional class dependencies”. In: *ICLR*. 2019.

[73] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* (2021), pp. 1–11.

[74] Bingyi Kang and Jiashi Feng. “Transferable meta learning across domains”. In: *UAI*. 2018.

[75] Alexandros Karargyris et al. “Creation and validation of a chest X-ray dataset with eye-tracking and report dictation for AI development”. In: *Scientific data* 8.1 (2021), pp. 1–18.

[76] Diederik P. Kingma and Max Welling. “Auto-encoding variational Bayes”. In: *ICLR*. 2014.

[77] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *ICLR*. 2017.

[78] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. “Siamese neural networks for one-shot image recognition”. In: *ICML Deep Learning Workshop*. 2015.

[79] Jonathan Krause et al. “3D object representations for fine-grained categorization”. In: *ICCV workshops*. 2013.

[80] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: *Technical Report* (2009).

[81] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *NeurIPS*. 2012.

[82] John K Kruschke. “Bayesian data analysis”. In: *Wiley Interdisciplinary Reviews: Cognitive Science* 1.5 (2010), pp. 658–676.

[83] Lindsey Kuper et al. “Toward scalable verification for safety-critical deep networks”. In: *SysML*. 2018.

[84] Lee Kwonjoon et al. “Meta-learning with differentiable convex optimization”. In: *CVPR*. 2019.

[85] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. “Human-level concept learning through probabilistic program induction”. In: *Science* 350.6266 (2015), pp. 1332–1338.

[86] S. Lazebnik, C. Schmid, and J. Ponce. “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories”. In: *CVPR*. 2006.

[87] Chen-Yu Lee et al. “Deeply-supervised nets”. In: *AISTATS*. 2015.

[88] Aoxue Li et al. “Few-shot learning with global class representations”. In: *ICCV*. 2019.

[89] Da Li et al. “Learning to generalize: Meta-learning for domain generalization”. In: *AAAI*. 2018.

[90] Kai Li et al. “Adversarial feature hallucination networks for few-shot learning”. In: *CVPR*. 2020.

[91] Yiying Li et al. “Feature-critic networks for heterogeneous domain generalization”. In: *ICML*. 2019.

[92] Zhenguo Li et al. “Meta-sgd: Learning to learn quickly for few-shot learning”. In: *CoRR* abs/1707.09835 (2017).

[93] Bo Liu et al. “Few-shot open-set recognition using meta-learning”. In: *CVPR*. 2020.

[94] Lu Liu et al. “A universal representation transformer layer for few-shot image classification”. In: *ICLR*. 2021.

[95] Shan Liu et al. “Visual driving assistance system based on few-shot learning”. In: *Multimedia Systems* (2021), pp. 1–11.

[96] Yanbin Liu et al. “Transductive propagation network for few-shot learning”. In: *ICLR*. 2019.

[97] Ziwei Liu et al. “Large-scale long-tailed recognition in an open world”. In: *CVPR*. 2019.

[98] Vincenzo Lomonaco and Davide Maltoni. “Core50: a new dataset and benchmark for continuous object recognition”. In: *PMLR*. 2017.

[99] Zelun Luo et al. “Label efficient learning of transferable representations across domains and tasks”. In: *NeurIPS*. 2017.

[100] Florian Lux and Ngoc Thang Vu. “Meta-learning for improving rare word recognition in end-to-end ASR”. In: *ICASSP*. 2021.

[101] D. J. C. MacKay. “The evidence framework applied to classification networks”. In: *Neural Computation* 4.5 (1992), pp. 720–736.

[102] Puneet Mangla et al. “Charting the right manifold: Manifold mixup for few-shot learning”. In: *WACV*. 2020.

[103] Gary Marcus. “Deep learning: A critical appraisal”. In: *arXiv preprint arXiv:1801.00631* (2018).

[104] Du Mengnan et al. “Fairness in Deep Learning: A Computational Perspective”. In: *CoRR*. 2019.

[105] Erik G Miller, Nicholas E Matsakis, and Paul A Viola. “Learning from one example through shared densities on transforms”. In: *CVPR*. 2000.

[106] Nikhil Mishra et al. “A simple neural attentive meta-learner”. In: *ICLR*. 2018.

[107] Eric Mitchell et al. “Offline meta-reinforcement learning with advantage weighting”. In: *ICML*. 2021.

[108] Tsendsuren Munkhdalai and Hong Yu. “Meta Networks”. In: *ICML*. 2017.

[109] Tsendsuren Munkhdalai et al. “Rapid adaptation with conditionally shifted neurons”. In: *ICML*. 2018.

[110] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[111] Anusha Nagabandi et al. “Learning to adapt in dynamic, real-world environments through meta-reinforcement learning”. In: 2019.

[112] Milad Nasr, Reza Shokri, and Amir Houmansadr. “Machine learning with membership privacy using adversarial regularization”. In: *ACM SIGSAC*. 2018.

[113] Lawrence Neal et al. “Open set learning with counterfactual images”. In: *ECCV*. 2018.

[114] Alex Nichol, Joshua Achiam, and John Schulman. “On first-order meta-learning algorithms”. In: *arXiv preprint arXiv:1803.02999* (2018).

[115] Jeremy Nixon et al. “Measuring calibration in deep learning.” In: *CVPR Workshops*. 2019.

[116] Boris Oreshkin, Pau Rodrıguez López, and Alexandre Lacoste. “Tadam: Task dependent adaptive metric for improved few-shot learning”. In: *NeurIPS*. 2018.

[117] Poojan Oza and Vishal M Patel. “C2ae: Class conditioned auto-encoder for open-set recognition”. In: *CVPR*. 2019.

[118] Poojan Oza and Vishal M Patel. “One-class convolutional neural network”. In: *IEEE Signal Processing Letters* 26 (2018), pp. 277–281.

[119] S. J. Pan and Q. Yang. “A survey on transfer learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359.

[120] Liang Pang et al. “Deeprank: A new deep architecture for relevance ranking in information retrieval”. In: *CIKM*. 2017.

[121] German I Parisi et al. “Continual lifelong learning with neural networks: A review”. In: *Neural Networks* 113 (2019), pp. 54–71.

[122] Massimiliano Patacchiola et al. “Bayesian meta-learning for the few-shot setting via deep kernels”. In: *NeurIPS*. 2020.

[123] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. “Ocgan: One-class novelty detection using gans with constrained latent representations”. In: *CVPR*. 2019.

[124] Pramuditha Perera and Vishal M Patel. “Learning deep features for one-class classification”. In: *IEEE TIP* (2019).

[125] Pramuditha Perera et al. “Generative-discriminative feature representations for open-set recognition”. In: *CVPR*. 2020.

[126] Juan-Manuel Perez-Rua et al. “Incremental few-shot object detection”. In: *CVPR*. 2020.

[127] Siyuan Qiao et al. “Few-shot image recognition by predicting parameters from activations”. In: *CVPR*. 2018.

[128] Sachin Ravi and Alex Beatson. “Amortized Bayesian meta-learning”. In: *ICLR*. 2019.

[129] Sachin Ravi and Hugo Larochelle. “Optimization as a model for few-shot learning”. In: *ICLR*. 2017.

[130] Sylvestre-Alvise Rebuffi et al. “iCaRL: Incremental Classifier and Representation Learning”. In: *CVPR*. 2017.

[131] Mengye Ren et al. “Incremental few-shot learning with attention attractor networks”. In: *NeurIPS*. 2019.

[132] Mengye Ren et al. “Meta-learning for semi-supervised few-shot classification”. In: *ICLR*. 2018.

[133] James Requeima et al. “Fast and flexible multi-task classification using conditional neural adaptive processes”. In: *NeurIPS*. 2019.

[134] Samuel Ritter et al. “Been there, done that: Meta-learning with episodic recall”. In: *ICML*. 2018.

[135] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. “Transfer learning in a transductive setting”. In: *NeurIPS*. 2013.

[136] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[137] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[138] Andrei A Rusu et al. “Meta-learning with latent embedding optimization”. In: *ICLR*. 2019.

[139] Tim Salimans et al. “Evolution strategies as a scalable alternative to reinforcement learning”. In: *arXiv preprint arXiv:1703.03864* (2017).

[140] Adam Santoro et al. “A simple neural network module for relational reasoning”. In: *NeurIPS*. 2017.

[141] Adam Santoro et al. “Meta-learning with memory-augmented neural networks”. In: *ICML*. 2016.

[142] Walter Scheirer et al. “Toward open set recognition”. In: *TPAMI* 35 (2013).

[143] Patrick Schlachter, Yiwen Liao, and Bin Yang. “Open-set recognition using intra-class splitting”. In: *EUSIPCO*. 2019.

[144] Jürgen Schmidhuber. “Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook”. PhD thesis. Technische Universität München, 1987.

[145] Edgar Schonfeld et al. “Generalized zero-and few-shot learning via aligned variational autoencoders”. In: *CVPR*. 2019.

[146] Maruan Al-Shedivat et al. “Continuous adaptation via meta-learning in nonstationary and competitive environments”. In: *ICLR*. 2018.

[147] Xiahan Shi et al. “Relational generalized few-shot learning”. In: *BMVC*. 2020.

[148] Hoo-Chang Shin et al. “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning”. In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1285–1298.

[149] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *nature* 529.7587 (2016), pp. 484–489.

[150] Jake Snell, Kevin Swersky, and Richard S Zemel. “Prototypical Networks for Few-shot Learning”. In: *NeurIPS*. 2017.

[151] C. Spearman. “The proof and measurement of association between two things”. In: *The American Journal of Psychology* 15.1 (1904), pp. 72–101.

[152] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *JMLR* 15.1 (2014), pp. 1929–1958.

[153] R. K. Srivastava, K. Greff, and J. Schmidhuber. “Highway Networks”. In: *ICML Deep Learning Workshop*. 2015.

[154] Kenneth O Stanley et al. “Designing neural networks through neuroevolution”. In: *Nature Machine Intelligence* 1.1 (2019), pp. 24–35.

[155] Jiamei Sun et al. “Explanation-guided training for cross-domain few-shot classification”. In: *arXiv preprint arXiv:2007.08790* (2020).

[156] Qianru Sun et al. “Meta-transfer learning for few-shot learning”. In: *CVPR*. 2019.

[157] Xin Sun et al. “Conditional gaussian distribution learning for open set recognition”. In: *CVPR*. 2020.

[158] Niko Sünderhauf et al. “The limits and potentials of deep learning for robotics”. In: *The International Journal of Robotics Research* 37.4-5 (2018), pp. 405–420.

[159] Flood Sung et al. “Learning to compare: Relation network for few-shot learning”. In: *CVPR*. 2018.

[160] Christian Szegedy et al. “Going deeper with convolutions”. In: *CVPR*. 2015.

[161] Xiaoyu Tao et al. “Few-shot class-incremental learning”. In: *CVPR*. 2020.

[162] Sebastian Thrun. “Is learning the n-th thing any easier than learning the first?” In: *NeurIPS*. 1996.

[163] Sebastian Thrun and Lorien Pratt. “Learning to learn: Introduction and overview”. In: *Learning to learn*. 1998, pp. 3–17.

[164] Eleni Triantafillou et al. “Meta-dataset: A dataset of datasets for learning to learn from few examples”. In: *ICLR*. 2019.

[165] Hung-Yu Tseng et al. “Cross-domain few-shot classification via learned feature-wise transformation”. In: *ICLR*. 2020.

[166] Gido M Van de Ven and Andreas S Tolias. “Three scenarios for continual learning”. In: *NeurIPS Workshop*. 2018.

[167] Ashish Vaswani et al. “Attention is all you need”. In: *NeurIPS*. 2017.

[168] Michael Veale and Reuben Binns. “Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data”. In: *Big Data & Society* (2017).

[169] A. Vedaldi et al. “Multiple kernels for object detection”. In: *ICCV*. 2009.

[170] Oriol Vinyals et al. “Matching networks for one shot learning”. In: *NeurIPS*. 2016.

[171] Risto Vuorio et al. “Multimodal model-agnostic meta-learning via task-aware modulation”. In: *NeurIPS*. 2019.

[172] Hongyu Wang et al. “A comparison of machine learning methods for cross-domain few-shot learning”. In: *AJCAI*. 2020.

[173] Jane X Wang et al. “Learning to reinforcement learn”. In: *arXiv preprint arXiv:1611.05763* (2016).

[174] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. “Learning to model the tail”. In: *NeurIPS*. 2017.

[175] Yu-Xiong Wang et al. “Low-shot learning from imaginary data”. In: *CVPR*. 2018.

[176] Yan Wang et al. “Simpleshot: Revisiting nearest-neighbor classification for few-shot learning”. In: *arXiv preprint arXiv:1911.04623* (2019).

[177] Yaqing Wang et al. “Generalizing from a few examples: A survey on few-shot learning”. In: *ACM Comput. Surv.* 53.3 (2020).

[178] Jian Wei et al. “Collaborative filtering and deep learning based recommendation system for cold start items”. In: *Expert Systems with Applications* 69 (2017), pp. 29–39.

[179] Olga Wichrowska et al. “Learned optimizers that scale and generalize”. In: *ICML*. 2017.

[180] Ken CL Wong, Tanveer Syeda-Mahmood, and Mehdi Moradi. “Building medical image classifiers with very limited data using segmentation networks”. In: *Medical image analysis* (2018).

[181] Yue Wu et al. “Large scale incremental learning”. In: *CVPR*. 2019.

[182] Chen Xing et al. “Adaptive cross-modal few-shot learning”. In: *NeurIPS* (2019).

[183] Zhongwen Xu, Hado van Hasselt, and David Silver. “Meta-gradient reinforcement learning”. In: 2018.

[184] Chao Yan, Xiaojia Xiang, and Chang Wang. “Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments”. In: *Journal of Intelligent & Robotic Systems* 98.2 (2020), pp. 297–309.

[185] Shengxiang Yang and Xin Yao. “Population-based incremental learning with associative memory for dynamic environments”. In: *IEEE Transactions on Evolutionary Computation* 12.5 (2008).

[186] Corinna Cortes Yann LeCun and Chris Burges. *MNIST handwritten digit database*. http://yann.lecun.com/exdb/mnist/. 2010.

[187] Han-Jia Ye, Hexiang Hu, and De-Chuan Zhan. “Learning adaptive classifiers synthesis for generalized few-shot learning”. In: *International Journal of Computer Vision* (2021), pp. 1–24.

[188] Han-Jia Ye et al. “Few-shot learning via embedding adaptation with set-to-set functions”. In: *CVPR*. 2020.

[189] Mingzhang Yin et al. “Meta-learning without memorization”. In: *ICLR*. 2020.

[190] Jaesik Yoon et al. “Bayesian model-agnostic meta-learning”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 7332–7342.

[191] Sung Whan Yoon, Seo Jun, and Moon Jaekyun. “Tapnet: Neural network augmented with task-adaptive projection for few-shot learning”. In: *ICML*. 2019.

[192] Ryota Yoshihashi et al. “Classification-reconstruction learning for open-set recognition”. In: *CVPR*. 2019.

[193] Jason Yosinski et al. “How transferable are features in deep neural networks?” In: *NeurIPS*. 2014.

[194] M. D. Zeiler and R. Fergus. “Visualizing and understanding convolutional networks”. In: *ECCV*. 2014.

[195] Zhang Zewang et al. “AdaDurIAN: Few-shot adaptation for neural Text-to-Speech with DurIAN”. In: *CoRR* abs/2005.05642 (2020).

[196] Xueting Zhang et al. “RelationNet2: Deep comparison columns for few-shot learning”. In: *IJCNN*. 2020.

[197] Wenyi Zhao et al. “Face recognition: A literature survey”. In: *ACM computing surveys (CSUR)* 35.4 (2003), pp. 399–458.

[198] Luisa M Zintgraf et al. “Fast context adaptation via meta-learning”. In: *ICML*. 2019.

[199] Bo Zong et al. “Deep autoencoding gaussian mixture model for unsupervised anomaly detection”. In: *ICLR*. 2018.