Deep Knowledge Transfer for Generalization across Tasks and Domains under Data Scarcity

On intersections of anomaly detection, few-shot learning, continual learning, domain generalization and data-free learning

Dissertation von Ahmed Frikha

München 2022

Deep Knowledge Transfer for Generalization across Tasks and Domains under Data Scarcity

On intersections of anomaly detection, few-shot learning, continual learning, domain generalization and data-free learning

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

eingereicht von
Ahmed Frikha

aus
Tunis, Tunesien

am 05.08.2022

Erstgutachter:	Prof. Dr. Volker Tresp
Zweitgutachter:	Prof. Dr. Amos Storkey
Drittgutachter:	Prof. Dr. Florian Büttner
Tag der mündlichen Prüfung:	07.11.2022

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. .5.)

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Frikha, Ahmed
Name, Vorname

Princeton (USA), 05.08.2022
Ort, Datum

Ahmed Frikha
Unterschrift Doktorand/in

Formular 3.2

Contents

Abstract

Over the last decade, deep learning approaches have achieved tremendous performance in a wide variety of fields, e.g., computer vision and natural language understanding, and across several sectors such as healthcare, industrial manufacturing, and driverless mobility. Most deep learning successes were accomplished in learning scenarios fulfilling the two following requirements. First, large amounts of data are available for training the deep learning model and there are no access restrictions to the data. Second, the data used for training and testing is independent and identically distributed (i.i.d.). However, many real-world applications infringe at least one of the aforementioned requirements, which results in challenging learning problems. The present thesis comprises four contributions to address four such learning problems. In each contribution, we propose a novel method and empirically demonstrate its effectiveness for the corresponding problem setting.

The first part addresses the underexplored intersection of the few-shot learning and the one-class classification problems. In this learning scenario, the model has to learn a new task using only a few examples from only the majority class, without overfitting to the few examples or to the majority class. This learning scenario is faced in real-world applications of anomaly detection where data is scarce. We propose an episode sampling technique to adapt meta-learning algorithms designed for class-balanced few-shot classification to the addressed few-shot one-class classification problem. This is done by optimizing for a model initialization tailored for the addressed scenario. In addition, we provide theoretical and empirical analyses to investigate the need for second-order derivatives to learn such parameter initializations. Our experiments on 8 image and time-series datasets, including a real-world dataset of industrial sensor readings, demonstrate the effectiveness of our method.

The second part tackles the intersection of the continual learning and the anomaly detection problems, which we are the first to explore, to the best of our knowledge. In this learning scenario, the model is exposed to a stream of anomaly detection tasks, i.e., only

examples from the normal class are available, that it has to learn sequentially. Such problem settings are encountered in anomaly detection applications where the data distribution continuously changes. We propose a meta-learning approach that learns parameter-specific initializations and learning rates suitable for continual anomaly detection. Our empirical evaluations show that a model trained with our algorithm is able to learn up 100 anomaly detection tasks sequentially with minimal catastrophic forgetting and overfitting to the majority class.

In the third part, we address the domain generalization problem, in which a model trained on several source domains is expected to generalize well to data from a previously unseen target domain, without any modification or exposure to its data. This challenging learning scenario is present in applications involving domain shift, e.g., different clinical centers using different MRI scanners or data acquisition protocols. We assume that learning to extract a richer set of features improves the transfer to a wider set of unknown domains. Motivated by this, we propose an algorithm that identifies the already learned features and corrupts them, hence enforcing new feature discovery. We leverage methods from the explainable machine learning literature to identify the features, and apply the targeted corruption on multiple representation levels, including input data and high-level embeddings. Our extensive empirical evaluation shows that our approach outperforms 18 domain generalization algorithms on multiple benchmark datasets.

The last part of the thesis addresses the intersection of domain generalization and data-free learning methods, which we are the first to explore, to the best of our knowledge. Hereby, we address the learning scenario where a model robust to domain shift is needed and only models trained on the same task but different domains are available instead of the original datasets. This learning scenario is relevant for any domain generalization application where the access to the data of the source domains is restricted, e.g., due to concerns about data privacy concerns or intellectual property infringement. We develop an approach that extracts and fuses domain-specific knowledge from the available teacher models into a student model robust to domain shift, by generating synthetic cross-domain data. Our empirical evaluation demonstrates the effectiveness of our method which outperforms ensemble and data-free knowledge distillation baselines. Most importantly, the proposed approach substantially reduces the gap between the best data-free baseline and the upper-bound baseline that uses the original private data.

Zusammenfassung

Im vergangenen Jahrzent haben Deep-Learning-Ansätze in einer Vielzahl von Bereichen, z.B. im Computer-Vision und beim Verstehen natürlicher Sprache, sowie in verschiedenen Sektoren wie dem Gesundheitswesen, der industriellen Fertigung und der fahrerlosen Mobilität enorme Leistungen erzielt. Die meisten Deep-Learning-Erfolge wurden in Lernszenarien erzielt, die die folgenden zwei Anforderungen erfüllen. Erstens, es stehen große Datenmengen für das Training des Deep-Learning-Modells zur Verfügung und es gibt keine Zugangsbeschränkungen zu den Daten. Zweitens, die für Training und Test verwendeten Daten sind unabhängig und identisch verteilt (i.i.d.). Viele reale Anwendungen verstoßen jedoch gegen mindestens eine der vorgenannten Anforderungen, was zu herausfordernden Lernproblemen führt. Die vorliegende Dissertation umfasst vier Beiträge, die sich mit vier solchen Lernproblemen befassen. In jedem Beitrag schlagen wir eine neue Methode vor und demonstrieren empirisch ihre Effektivität für die entsprechende Problemstellung.

Der erste Teil befasst sich mit der noch wenig erforschten Überschneidung der Probleme der *few-shot learning* und *one-class classification*. In diesem Lernszenario muss das Modell eine neue Aufgabe mit nur wenigen Beispielen aus der Mehrheitsklasse erlernen, ohne dass eine Überanpassung an die wenigen Beispiele oder an die Mehrheitsklasse erfolgt. Dieses Lernszenario kommt in der Praxis bei Anomalieerkennung-Anwendungen vor, in denen die Daten knapp sind. Wir schlagen ein Episoden-Sampling-Verfahren vor, um Meta-Learning-Algorithmen, die für eine klassenbalancierte *few-shot learning* entwickelt wurden, an das adressierte *few-shot one-class classification* Problem anzupassen. Dies geschieht durch die Optimierung einer Parameterinitialisierung, die auf das adressierte Lernszenario zugeschnitten ist. Darüber hinaus bieten wir theoretische und empirische Analysen, um die Notwendigkeit von Ableitungen zweiter Ordnung zum Erlernen solcher Parameterinitialisierungen zu untersuchen. Unsere Experimente mit 8 Bild- und Zeitseriendatensätzen, einschließlich eines realen Datensatzes von industriellen Sensormesswerten, demonstrieren die Effektivität unserer Methode.

Der zweite Teil befasst sich mit der Überschneidung der Probleme des kontinuierlichen Lernens und der Anomalieerkennung, die wir unseres Wissens nach als erste untersuchen. In diesem Lernszenario wird das Modell einer Sequenz von unterschiedlichen Anomalieerkennungsaufgaben ausgesetzt, d.h. es stehen nur Beispiele aus der normalen Klasse zur Verfügung, die es sequentiell lernen muss. Solche Problemstellungen treten bei Anomalieerkennung-Anwendungen auf, bei denen sich die Datenverteilung ständig ändert. Wir schlagen einen Meta-Learning-Ansatz vor, der parameterspezifische Initialisierungen und Lernraten erlernt, die für die kontinuierliche Anomalieerkennung geeignet sind. Unsere empirischen Auswertungen zeigen, dass ein mit unserem Algorithmus trainiertes Modell in der Lage ist, bis zu 100 Anomalieerkennungsaufgaben sequentiell zu erlernen, und zwar mit minimalem katastrophalen Vergessen und minimaler Überanpassung an die Mehrheitsklasse.

Im dritten Teil befassen wir uns mit dem Problem der Domänengeneralisierung, bei dem von einem Modell, das auf mehreren Quelldomänen trainiert wurde, erwartet wird, dass es sich gut auf Daten aus einer zuvor unbekannten Zieldomäne verallgemeinern lässt, ohne dass es modifiziert wird oder seinen Daten ausgesetzt wird. Dieses schwierige Lernszenario tritt bei Anwendungen auf, die Daten aus mehreren Domänen, d.h. mehrere Datenverteilungen, involvieren, z.B. verschiedene klinische Zentren, die unterschiedliche MRI-Scanner oder Datenerfassungsprotokolle verwenden. Wir gehen davon aus, dass das Erlernen eines reichhaltigeren Satzes von Merkmalen die Übertragung auf einen größeren Satz unbekannter Domänen verbessert. Aus diesem Grund schlagen wir einen Algorithmus vor, der die bereits gelernten Merkmale identifiziert und sie verfälscht, um so die Entdeckung neuer Merkmale zu erzwingen. Wir nutzen Methoden aus der Literatur des erklärbaren maschinellen Lernens, um diese Merkmale zu identifizieren, und wenden die gezielte Verfälschung auf mehreren Repräsentationsebenen an, einschließlich Eingabedaten und High-Level-Repräsentationen. Unsere umfangreiche empirische Evaluierung zeigt, dass unser Ansatz 18 Algorithmen der Domänengeneralisierung in mehreren Benchmark-Datensätzen übertrifft.

Der letzte Teil der Arbeit befasst sich mit der Überschneidung von Domänengeneralisierung und datenfreien Lernmethoden, die wir unseres Wissens nach als erste untersuchen. Wir befassen uns mit dem Lernszenario, in dem ein Modell benötigt wird, das gegenüber Domänenverschiebungen bzw. -änderungen robust ist, und in dem anstelle der ursprünglichen Datensätze nur Modelle zur Verfügung stehen, die auf der gleichen Aufgabe, aber in unterschiedlichen Domänen trainiert wurden. Dieses Lernszenario ist für jede

Domänengeneralisierungsanwendung relevant, bei der der Zugang zu den Daten der Quelldomänen eingeschränkt ist, z.B. aufgrund von Bedenken hinsichtlich des Datenschutzes oder der Verletzung geistigen Eigentums. Wir entwickeln einen Ansatz, der domänenspezifisches Wissen aus den verfügbaren Lehrermodellen extrahiert und in einem Schülermodell verschmilzt, das gegenüber Domänenverschiebungen robust ist, indem es synthetische domänenübergreifende Daten erzeugt. Unsere empirische Evaluierung zeigt die Effektivität unserer Methode, die Ensemble- und datenfreie Wissensdestillationsmethoden übertrifft. Besonders wichtig ist, dass der vorgeschlagene Ansatz die Lücke zwischen dem besten datenfreien Ansatz und der Methode, die die privaten Originaldaten verwendet, erheblich verringert.

Acknowledgements

During the last years, many people have contributed to the successful completion of my PhD.

First, I would like to express my deepest gratitude to my advisors and mentors Dr. Denis Krompaß and Prof. Dr. Volker Tresp. Denis and Volker gave me the excellent opportunity to undertake my PhD research at the Ludwig-Maximilian University of Munich and Siemens AG. They provided me with the freedom to explore multiple research fields and were always supportive and helpful. Denis has always taken the extra time for long and fruitful discussions about methodological details in research and tooling. Besides, he encouraged me to investigate unconventional and innovative research topics, and taught me to cope with failures lightheartedly and learn from them. I am extremely thankful for that. Volker has always provided prompt feedback and taken the time to discuss all our publications, which enabled me to learn how a world-class researcher thinks and assesses research work. Volker has also organized a platform for regular exchange with many other PhD students, which was helpful to receive early feedback and discuss the current state of our research works. I am very honored that Prof. Dr. Amos Storkey and Prof. Dr. Florian Büttner agreed to be external examiners of my thesis.

Furthermore, I would like to thank all the colleagues at Siemens who supported me during my PhD. First, I would like to thank Dr. Mark Buckley for introducing me to the Machine Intelligence research group and to my future master thesis and PhD advisor, Denis, when I was a working student at Siemens back in 2018. I am very thankful to Dr. Ulli Waltinger, the head of the former research group I worked in, for being an enabler and inspiring leader. He encouraged me to present early results of my PhD work at the German-French Summer School on Transfer Learning and gave me the opportunity of leading an AI Lab acceleration project. I would also like to thank the head of my current research group, Dr. Sebastian Mittelstaedt, for his support and for facilitating the attendance of conferences and the ELLIS Doctoral Symposium to present my work. Moreover, I am

grateful to my advisor Denis, the head of the research group in Princeton, Olympia Brikis, and my internship advisor Dr. Biswadip Dey, for the great research internship at Siemens in Princeton, USA. I would like to extend my thanks to the MIC colleagues and the participants of the AI Journal Club and Team Board for sharing their knowledge about recent research publications and their work on ongoing customer projects. These weekly meetings have helped me stay up-to-date with the most recent advances in the field and learn about the relevant challenges faced in industrial real-world applications. Special thanks to Dr. Sigurd Spieckermann for the long and various technical discussions, and to Dr. Fabian Rhein for organizing the Siemens PhD network and organizing social and knowledge exchange events for us. I would like to express my gratitude to the AI Lab team for welcoming me into their team and offices, and for organizing three awesome and intensive Hackathons. Many thanks also to the master's students I had the pleasure of collaborating with, including Markus Kittel, Sebastian Gruber, and Alexander Schober. Special thanks also to my former master thesis student, current fellow PhD student, and co-author Haokun Chen for the long and fruitful discussions, his dedication, and the great collaborations. I am also thankful to my fellow PhD students at LMU and Siemens for their support, including Julia Gottfriedsen, Usama Yaseen, and Zhiliang Wu.

Finally, I would like to thank all my friends who supported me throughout this journey. I would to especially express my gratitude to Amine, Aymen, Memix, Nawes, Sana, and Yassine, for always being there and for all the weekend plans. Special thanks to Amine Kechaou for proofreading all my publications and for his valuable feedback. I am also extremely grateful to Sana for her continual support, encouragement, and patience. Last but not least, I would like to thank my family for their unconditional love and support, their constant encouragement, and their sacrifices, throughout my life. This would not have been possible without them.

List of Publications and Declaration of Authorship

- Ahmed Frikha, Denis Krompaß, Hans-Georg Köpken, and Volker Tresp. Few-shot one-class classification via meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7448–7456, 2021b

  Denis Krompaß and I conceived the original research contributions. I performed all implementations and evaluations. I wrote the initial draft of the manuscript and did most of the subsequent corrections. I regularly discussed this work with my co-author Denis Krompaß. All co-authors contributed to improving the manuscript.

  This published work serves as Chapter 2 in this thesis.

- Ahmed Frikha, Denis Krompaß, and Volker Tresp. Arcade: A rapid continual anomaly detector. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10449–10456, 2021d. doi: 10.1109/ICPR48806.2021.9412627

  I conceived the original research contributions and performed all implementations and evaluations. I wrote the initial draft of the manuscript and did most of the subsequent corrections. I regularly discussed this work with my co-author Denis Krompaß. All co-authors contributed to improving the manuscript.

  This published work serves as Chapter 3 in this thesis.

- Ahmed Frikha, Denis Krompaß, and Volker Tresp. Columbus: Automated discovery of new multi-level features for domain generalization via knowledge corruption. *arXiv:2109.04320*, 2021c. *To appear in 26th International Conference on Pattern Recognition (ICPR), 2022*

I conceived the original research contributions and performed all implementations and evaluations. I wrote the initial draft of the manuscript and did most of the subsequent corrections. I regularly discussed this work with my co-author Denis Krompaß. All co-authors contributed to improving the manuscript.

This published work serves as Chapter 4 in this thesis.

- Ahmed Frikha, Haokun Chen, Denis Krompaß, Thomas Runkler, and Volker Tresp. Towards data-free domain generalization. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021a

  I conceived the original research contributions and designed the experimental evaluations. My co-author Haokun Chen performed the implementations and evaluations. I wrote the initial draft of the manuscript and did most of the subsequent corrections. Haokun Chen and I regularly discussed this work with the co-authors, Denis Krompaß and Thomas Runkler. All co-authors contributed to improving the manuscript.

  This published work serves as Chapter 5 in this thesis.

Other publications

- Haokun Chen, Ahmed Frikha, Denis Krompass, and Volker Tresp. Fraug: Tackling federated learning with non-iid features via representation augmentation. *arXiv preprint arXiv:2205.14900*, 2022

Chapter 1

Introduction

This chapter provides an overview of the technical background useful for understanding the following chapters. In section 1, we discuss challenging learning scenarios relevant to real-world applications and highlight the importance of knowledge transfer. Section 2 introduces research areas related to our contributions in chapters 2 and 3, including anomaly detection, meta-learning, few-shot learning, and continual learning. In section 3, we present the domain generalization and data-free learning topics, which are relevant for our contributions in chapters 4 and 5. In particular, we discuss the motivation behind addressing each of these topics, their related works, as well as their connection to our contributions.

1.1 Knowledge transfer

This section presents challenging learning scenarios encountered in real-world applications and underlines the importance of knowledge transfer methods in addressing them. Subsequently, we discuss two types of data distribution shift, domain and task shift.

1.1.1 Motivation

Over the last decade, deep learning methods have achieved impressive results on different data types, such as images Krizhevsky et al. (2012), natural language Devlin et al. (2018), tabular data Arık and Pfister (2021), time-series Oreshkin et al. (2019), and combinations thereof Radford et al. (2021); Ren et al. (2019), as well as across multiple sectors including industrial manufacturing Nasir and Sassani (2021), healthcare Bakator and Radosav (2018)

and finance Ozbayoglu et al. (2020) to name a few. Most deep learning successes were achieved in offline learning scenarios involving a single task and under the assumption that the data used for training and evaluation is independent and identically distributed (i.i.d.). Hereby, the availability and unrestricted access to a large dataset are usually assumed.

Real-world applications exhibit characteristics, which give rise to challenging learning problems. Firstly, there exist applications where data is scarce, e.g., due to high annotation or collection costs, leading to the overfitting of deep learning models to the small datasets. Such applications motivated the research efforts in the field of few-shot learning Wang and Yao (2019) aiming to train models to learn new tasks using few datapoints. Secondly, in multiple applications, data collection happens gradually and with frequent changes in the data distribution, e.g., as a result of modifications to a manufactured product. In such learning scenarios, deep learning models suffer from the catastrophic forgetting phenomenon French (1999), i.e., they are not able to retain knowledge acquired from earlier tasks. Hence, developing models that are able to continuously learn new tasks without forgetting has gained a surge of interest in the continual learning research Parisi et al. (2019) during the last years. Thirdly, distribution shifts are often observed between the data used for training in the model development phase and the data encountered in the production phase after the model deployment. Such distribution shifts result in a deterioration of the model performance on the target distribution faced at evaluation time. A plethora of domain adaptation and generalization works Wang and Deng (2018); Wilson and Cook (2020); Zhou et al. (2021a) was proposed to tackle this problem. Fourthly, in the light of increasing data privacy awareness and concerns, several data-owning entities have become reluctant to share their data, raising the need for privacy-preserving data-driven methods. Consequently, deep learning approaches leveraging federated learning Kairouz et al. (2021) and, more recently, data-free knowledge transfer Liu et al. (2021b) have been proposed. Finally, some applications demonstrate an extreme class-imbalance in the data, e.g., the detection of banking fraud, rare diseases, or manufacturing deficiencies. Training neural networks on severely class-imbalanced data, or in the extreme case on data from only one class, leads to overfitting to the majority class and performance degradation. Therefore, one-class classification and anomaly detection were extensively studied and several approaches able to cope with class-imbalance were proposed Khan and Madden (2014); Aggarwal (2015).

In many real-world applications, the aforementioned learning problems are usually not encountered separately. In the present thesis, we investigate three underexplored inter-

sections of pairs of these problems. In particular, we address the intersection of few-shot learning and one-class classification (Chapter 2), the intersection of continual learning and one-class classification (Chapter 3), and the intersection of domain generalization and privacy-preserving data-free learning (Chapter 5). Moreover, we propose a novel approach to the domain generalization problem (Chapter 4). We present an overview of the problem settings we address in Table 1.1.

Table 1.1: Overview of the problem settings addressed in the thesis contributions.

Learning scenario addressed	Related research problems	Chapter
Few-Shot One-Class Classification	Few-Shot Learning One-Class Classification	Chapter 2
Continual Anomaly Detection	Continual Learning Anomaly Detection	Chapter 3
Domain Generalization	-	Chapter 4
Data-Free Domain Generalization	Data-Free Knowledge Transfer Domain Generalization	Chapter 5

When faced with different source and target distributions, as is the case in some of the aforementioned problems and their intersections, knowledge transfer from the (usually label-rich) source distributions to the (usually label-scarce) target distributions is crucial. Hence, to tackle these challenging learning problems, we develop approaches that leverage knowledge transfer.

Transfer learning Torrey and Shavlik (2010) refers to the problem setting and the family of techniques that improve learning a task, the target task, by leveraging data from one or many other task(s), the source task(s). The most common transfer learning method consists in further training some or all the layers of a model, which is initially trained on the source task(s), using data from the target task. A wide variety of transfer learning approaches were developed over the years. We refer to Zhuang et al. (2020) for an extensive overview.

Transfer learning can improve the learning of the target task in three major ways, which we visualize in Figure 1.1. First, a higher start performance can be achieved which is beneficial in cold start situations for instance. In addition, transfer learning can enable faster learning of the target task, i.e., the same performance is reached with fewer training itera-

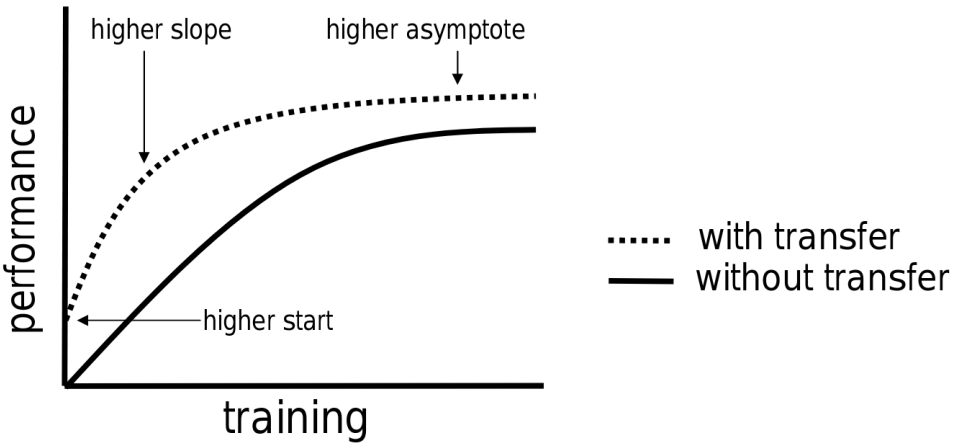


Figure 1.1: The learning improvements yielded by transfer learning Torrey and Shavlik (2010).

tions. Finally, the final model performance can be increased by leveraging the knowledge acquired from the source task(s).

1.1.2 Tasks and Domains

In the works that we propose in chapters 2-5 of the present thesis, the source data distribution used for training and the target data distributions used for evaluation are different. In the following, we distinguish two types of distribution differences that are relevant for our works: domain shift and task change.

We define domain and tasks, following Zhuang et al. (2020). A domain $\mathcal{D}$ by a feature space $\mathcal{X}$ and a marginal distribution $P(X)$, i.e., $\mathcal{D} = \{\mathcal{X}, P(X)\}$, where X denotes a set of instances, i.e., $X = \{\mathbf{x}|\mathbf{x}_i \in \mathcal{X}, i = 1, ..., n\}$. On the other hand, a task $\mathcal{T}$ id defined by a label space $\mathcal{Y}$ and an implicit decision function f that needs to be learned from the data. f can be described by the conditional distribution $P(y|\mathbf{x})$. Our contributions in the present thesis consider scenarios that involve multiple datasets with different data distributions $P_i(\mathbf{x}, y)$. $P_i(\mathbf{x}, y)$ can be rewritten as $P_i(y|\mathbf{x})P_i(\mathbf{x})$ and $P_i(\mathbf{x}|y)P_i(y)$.

Following Kairouz et al. (2021) and Li et al. (2021a), we define domain shift as the setting covering *covariate shift* and *concept shift*. In covariate shift, the datasets have different marginal feature distributions, i.e., $P_i(\mathbf{x}) \neq P_j(\mathbf{x})$, while the conditional label distribution are the same, i.e., $P_i(y|\mathbf{x}) = P_j(y|\mathbf{x})$. In concept shift, the datasets have different conditional feature distributions, i.e., $P_i(\mathbf{x}|y) \neq P_j(\mathbf{x}|y)$, while having the same marginal

label distribution, i.e., $P_i(y) = P_j(y)$. Domain shift is also referred to as *feature shift* in Li et al. (2021a).

Figure 1.2: Exemplary sets of data exhibiting domain shift

Figure 1.2 displays a concrete example of domain shift. Hereby, the different datasets have the same classes, e.g., horse, house, guitar, and dog, but different input feature distributions, i.e., pixel distributions, including hand-drawn sketches and pictures taken by a camera. The images used in Figure 1.2 are from the PACS dataset Li et al. (2017a) commonly used for domain generalization benchmarks.

Different learning settings involving domain shift are investigated in different research problems, including supervised Wang and Deng (2018) and unsupervised domain adaptation Wilson and Cook (2020), as well as domain generalization Zhou et al. (2021a) (Section 1.3). Our works presented in Chapters 4 and 5 consider problem settings involving domain shift.

We consider that two datasets have different tasks in the following cases: 1) the datasets have different label spaces, i.e., $\mathcal{Y}_i \neq \mathcal{Y}_j$, or 2) the datasets have different conditional label

distributions $P_i(y|\mathbf{x}) \neq P_j(y|\mathbf{x})$, even if the marginal feature distribution is the same, i.e., $P_i(\mathbf{x}) = P_j(\mathbf{x})$. Our works presented in Chapters 2 and 3 consider problem settings involving different tasks. A concrete example of datasets with different tasks can be seen in Figure 1.3. Here, the different tasks have different classes, while having the same number of classes, i.e., the same label space $\mathcal{Y}$. Note that even though some classes are shared across the tasks, they have different labels, i.e., $P_i(y|\mathbf{x}) \neq P_j(y|\mathbf{x})$. The images used are from the MiniImageNet dataset Ravi and Larochelle (2016) commonly used for few-shot learning benchmarks.

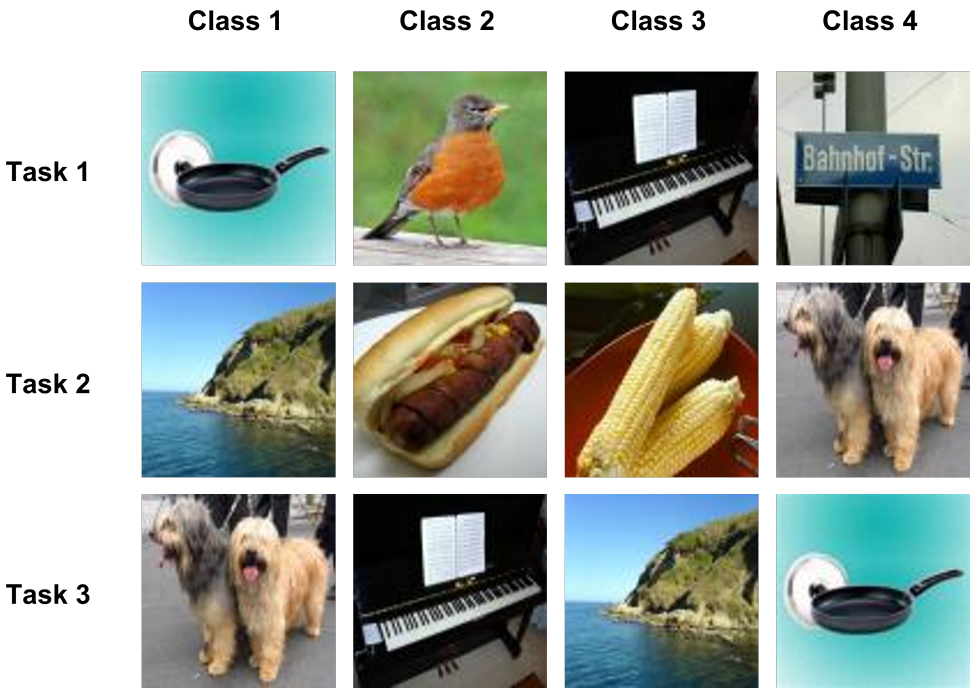


Figure 1.3: Examples of different 4-class classification tasks.

Different learning settings involving different tasks are investigated in different research problems, including few-shot learning Wang and Yao (2019) (Section 1.2.2), meta-learning Hospedales et al. (2020) (Section 1.2.1) and continual learning Delange et al. (2021) (Section 1.2.3). Recently, learning scenarios involving both domain shift and different tasks have enjoyed a surge of interest. These include cross-domain few-shot learning Guo et al. (2020) and continual domain generalization Li et al. (2020a).

1.2 Anomaly Detection via Meta-Learning

This section presents an overview of research areas that are related to our contributions in chapters 2 and 3. Section 1.2.1 provides information about anomaly detection and meta-learning, which are relevant for both contributions. Thereafter, section 1.2.2 introduces few-shot learning, the Model-Agnostic Meta-Learning (MAML) algorithm, and how they relate to our contribution from chapter 2. Finally, in section 1.2.3, we present the continual learning problem and its relationship to our contribution in chapter 3.

1.2.1 Introduction

In this section, we introduce and discuss the anomaly detection problem and the meta-learning paradigm, which are relevant for our contributions in chapters 2 and 3.

Anomaly Detection

Anomaly Detection (AD) is the task of differentiating between normal and abnormal examples, also called anomalies or outliers Chandola et al. (2009); Aggarwal (2015). Anomaly detection is present in several real-world applications in different sectors, including healthcare Prastawa et al. (2004), banking Raj and Portia (2011), cybersecurity Garcia-Teodoro et al. (2009) and industrial manufacturing Scime and Beuth (2018). Since anomalies occur very rarely compared to normal behavior, the data collected for anomaly detection applications exhibits a high class-imbalance in favor of the normal class. Hence, AD problems are usually formulated as One-Class Classification (OCC) problems Moya et al. (1993), i.e., few or no data examples from the anomalous class are available for model training Khan and Madden (2014). In other words, the task is to determine a binary classification decision boundary, while having access to only examples from one class, i.e., the normal class.

Several works were developed to address the AD and OCC problems. In the following, we discuss some of them and refer to Chandola et al. (2009); Khan and Madden (2014) for an extensive literature review. Classical OCC such as One-Class SVM (OC-SVM) Schölkopf et al. (2000) and Support Vector Data Description (SVDD) Tax and Duin (2004) methods leverage SVMs to learn a decision boundary that separates the normal class from outliers. To cope with high-dimensional data, feature extraction models are used to yield lower-dimensional representations of the data before feeding them to the SVM-based classifier Xu et al. (2015); Andrews et al. (2016); Erfani et al. (2016). Fully end-to-end deep

learning methods that incorporate both steps, i.e., embedding and classification were also developed Ruff et al. (2018). Another line of work trains auto-encoder models Hinton and Salakhutdinov (2006) to learn to reconstruct normal class examples and detects anomalies by their higher reconstruction loss values Hawkins et al. (2002); An and Cho (2015); Chen et al. (2017). More recent approaches leverage generative models, e.g., Generative Adversarial Networks (GAN) Goodfellow et al. (2014a) and Normalizing Flows Rezende and Mohamed (2015), to detect anomalies Schlegl et al. (2017); Ravanbakhsh et al. (2017); Sabokrou et al. (2018); Rudolph et al. (2021).

Meta-Learning

Traditional machine learning involves the optimization of model predictions over a dataset D including several data examples, as shown in Equation 1.1. Hereby the parameters θ of the prediction model are optimized to minimize a defined loss function $\mathcal{L}$. Here, ω denotes the meta-learner. The meta-learner defines the learning process of new tasks by the learner model parameterized by θ. It can be viewed as the learning algorithm used and its assumptions, e.g., the choice of a random initialization for θ or the usage of a specific rule for updating the learner's parameters.

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\ \mathcal{L}(D; \theta, \omega) \tag{1.1}$$

In contrast, meta-learning, also referred to as learning to learn Schmidhuber (1987); Thrun and Pratt (2012), is the process of optimizing a meta-learner ω over several learning problems, also called tasks $\mathcal{T}$ Hospedales et al. (2020) (Equation 1.2).

$$\omega^* = \underset{\omega}{\operatorname{argmin}}\ \mathbb{E}_{\mathcal{T} \sim P(\mathcal{T})}\ \mathcal{L}(D; \omega) \tag{1.2}$$

We present a schematic representation describing the meta-learning problem setting, inspired by De Lange et al. (2021), in Figure 1.4.

The tasks used for meta-learning usually share common characteristics and are sampled from a task-distribution $P(\mathcal{T})$. Hereby, we implicitly define a task $\mathcal{T}_i$ by its corresponding dataset D_i (Section 1.1.2). Examples of task-distributions include a distribution over 5-class classification tasks of natural images of animals, or a distribution over regression tasks on sine functions with different amplitudes and periods. In these examples, a task instance would be an image classification task with 5 specific classes of animals, and a regression

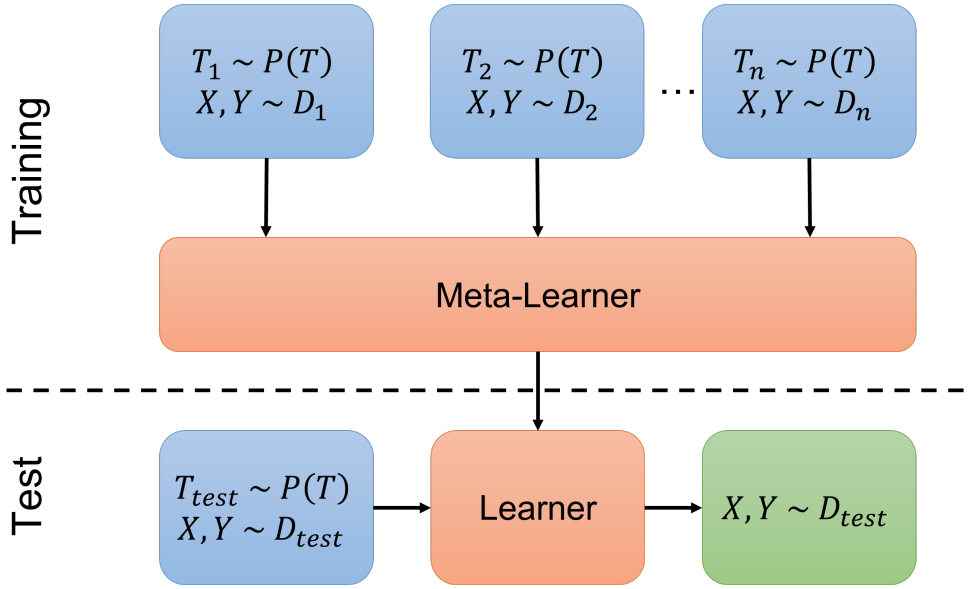


Figure 1.4: Schematic representation of the meta-learning problem setting.

task of a sine function with a defined amplitude and period. The tasks used for training the meta-learner are called the meta-training tasks.

At test time, the resulting meta-learner ω^* is applied to the learner model to facilitate learning a new task T_{test} which was not observed during training. The tasks used for the evaluation of the meta-learner are referred to as meta-testing tasks. A subset of the dataset D_{test} of the meta-testing task is used for training the learner using the meta-learner ω. The meta-learner ω can have a high impact on learning-related performance metrics such as convergence speed and data efficiency Finn et al. (2017), as well as the performance and characteristics of the predictive model θ trained by it, e.g., robustness to domains shift Li et al. (2018a); Dou et al. (2019) or adversarial attacks Yin et al. (2018). Examples of meta-learners include an initialization of the parameters of the predictive model θ Finn and Levine (2017), the parameters of another model that updates Hochreiter et al. (2001); Andrychowicz et al. (2016); Ravi and Larochelle (2016) or generates Rusu et al. (2018a); Zhmoginov et al. (2022) the parameters of the predictive model, a loss function Bechtle et al. (2021), or an optimization algorithm Li and Malik (2017).

The task distribution $P(\mathcal{T})$ plays an important role in defining the meta-learning objective and influences the optimization of the learning strategy. The task-distribution is usually designed to include tasks that share characteristics with the target tasks, i.e., the tasks expected to be encountered at test time. For instance, if the target application ex-

hibits data scarcity, the task-distribution should include few-shot learning tasks, i.e., tasks with a small training set Vinyals et al. (2016). In this case, the resulting meta-learner is optimized to learn new tasks with few datapoints. In practice, the task distribution $P(\mathcal{T})$ is usually discrete, i.e., it represents a collection of n datasets $P(\mathcal{T}) = \{D_1, D_2, ..., D_n\}$.

Several works developed meta-learning-based techniques to address the few-shot learning problem Finn et al. (2017); Li et al. (2017b); Nichol and Schulman (2018); Rusu et al. (2018b); Lee et al. (2019). Further applications include continual learning Javed and White (2019); Spigler (2019); Beaulieu et al. (2020); Requeima et al. (2019), robustness to domain shift Li et al. (2018a); Balaji et al. (2018); Dou et al. (2019), robustness to adversarial attacks Yin et al. (2018); Zhang et al. (2020), unsupervised learning Hsu et al. (2018); Khodadadeh et al. (2019), active learning Contardo et al. (2017); Pang et al. (2018), reinforcement learning Gupta et al. (2018); Nagabandi et al. (2018); Rakelly et al. (2019), hyperparamneter optimization Franceschi et al. (2018) and Neural Architecture Search (NAS) Liu et al. (2018). For an extensive review of meta-learning methods we refer to Hospedales et al. (2020); Huisman et al. (2021).

1.2.2 Few-Shot Anomaly Detection

This section introduces the few-shot learning problem and its related works, with a focus on the Model-Agnostic Meta-Learning (MAML) algorithm. Thereafter, we motivate and summarize our contribution from chapter 2 Frikha et al. (2021b).

Few-Shot Learning

Several real-world applications exhibit a high data scarcity, hence disallowing the usage of the conventional and data-hungry deep learning methods. There exist various reasons for data scarcity. On the one hand, the data collection process itself might be expensive, e.g., in the healthcare sector, or gradual, e.g., in cold start situations. On the other hand, the data annotation process might be expensive due to the scarcity of domain experts required for it, e.g., sensor readings collected in an industrial manufacturing plant can only be labeled by highly trained experts. To enable learning from scarce datasets, the few-shot learning (FSL) problem was introduced and studied.

A few-shot learning task includes a small training set, also called the support set, and a test set, commonly called the query set. A few-shot classification task is defined by the number of its classes n and the number of per-class examples k included in its support

set Vinyals et al. (2016). It is referred to as *n-way-k-shot* classification task. The FSL literature assumes access to a distribution of such FSL tasks $P(\mathcal{T}_{train})$ that is available for training. Given $P(\mathcal{T}_{train})$, the goal of few-shot learning is to enable a model θ to learn an unseen task using only its small support set, i.e., the trained model generalizes to the unseen query set of the same task.

Recently, a plethora of works was proposed to address few-shot learning. In the following, we discuss four main categories of few-shot learning approaches and refer to Wang et al. (2020a) for an extensive survey. Metric-based FSL techniques involve embedding the data into a representation space where examples belonging to the same class are similar according to a pre-defined metric, which facilitates classification Koch (2015); Vinyals et al. (2016); Snell et al. (2017); Sung et al. (2018); Oreshkin et al. (2018); Bertinetto et al. (2018); Lee et al. (2019). Optimization-based FSL approaches leverage meta-learning to yield a learning strategy tailored for learning FSL tasks. The learning strategies proposed include optimization algorithms Ravi and Larochelle (2016), model initializations Finn et al. (2017); Nichol and Schulman (2018), and/or parameter-specific learning rates Li et al. (2017b). Hybrid methods that combine metric-learning and meta-learning were also developed Rusu et al. (2018a); Lee and Choi (2018); Triantafillou et al. (2019). Finally, the third category of approaches modulates Requeima et al. (2019); Vuorio et al. (2019) or generates some Qi et al. (2018); Gidaris and Komodakis (2018) or all Zhmoginov et al. (2022) the parameters of the predictive model in a task-specific way.

Model-Agnostic Meta-Learning

In this section, we introduce the Model-Agnostic Meta-Learning (MAML) algorithm Finn et al. (2017), since our work builds upon it.

While MAML can also be used for few-shot regression and policy gradient reinforcement learning, we focus on its usage for few-shot classification. The meta-learned learning strategy, i.e., the meta-learner, that MAML optimizes is a model initialization that enables quick task-specific adaptation with few datapoints. For this, MAML uses a training task-distribution $p(\mathcal{T}_{train})$ that contains few-shot learning tasks, i.e., tasks with a small support set.

$$\theta_i^{'} = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}^S(f_\theta), \tag{1.3}$$

MAML employs a bi-level optimization scheme to optimize the parameter initialization θ. In the inner optimization step, MAML adapts θ to a task $\mathcal{T}_i$ by taking one (or more) gra-

dient descent steps, yielding model parameters θ_i' specific to task i (Equation1.3). Hereby, α denotes the learning rate, f the model parametrized by θ, and $\mathcal{L}_{\mathcal{T}_i}^S$ the loss function computed on the support set of $\mathcal{T}_i$.

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}^Q(f_{\theta_i'}). \tag{1.4}$$

In the outer optimization step, MAML optimizes the parameter initialization θ explicitly for few-shot learning (Equation 1.4). In particular, it minimizes $\mathcal{L}_{\mathcal{T}_i}^Q(f_{\theta_i'})$, the loss of the task-specific model θ_i' computed on the query set of $\mathcal{T}_i$. Note that the loss minimized is a good measure of the initialization suitability for few-shot learning. The learning rate used for the outer optimization is denoted by β. Note that the MAML meta-objective consists in learning an initialization that yields a low loss on unseen examples from a task $\mathcal{T}$ after a task-specific adaptation with few examples from that task.

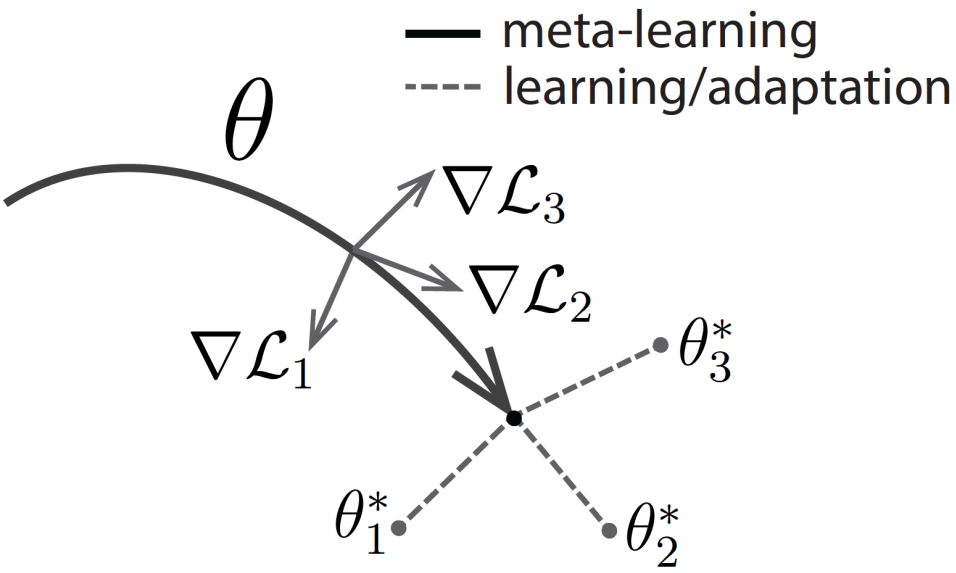


Figure 1.5: The meta-learned parameters θ allow fast adaptation to several tasks.

In each iteration, MAML samples a batch of training tasks from $p(\mathcal{T}_{train})$ and updates the model initialization θ following Equation 1.4. As depicted in Figure 1.5, the meta-learned parameter initialization lie in the vicinity of local optima of different tasks, hence enabling task-adaptation using only a few data examples. At test time, the meta-learned initialization is adapted to the target task by taking one (or more) gradient updates using its small support set.

Few-Shot Anomaly Detection via Meta-Learning

In this section, we summarize the motivation, method, and findings of our published work Frikha et al. (2021b) presented in Chapter 2.

On the one hand, most of the Anomaly Detection (AD) approaches developed in prior work (Section 1.2.1) require large datasets of normal examples to generalize Chandola et al. (2009). Such datasets are not available in data-scarce application scenarios (Section 1.2.2). On the other hand, the few-shot classification literature Wang et al. (2020a) focuses on class-balanced classification scenarios, where examples are available from all classes. Due to the extreme rarity of anomalous behavior, data examples from the anomalous class are usually not available, and One-Class Classification (OCC) techniques Khan and Madden (2014) are employed to perform AD.

Our work addresses the Few-Shot One-Class Classification (FS-OCC) problem, the underexplored intersection of the well-studied OCC, and few-shot learning problems. We formulate this problem as a meta-learning, where the tasks (Section 1.1.2) are AD tasks with different normal and anomalous classes. perspective Our contribution in this work is fourfold. Firstly, we empirically show that classical OCC methods fail in the low-data regime. Secondly, we theoretically analyze why the parameter initializations optimized by gradient-based meta-learning algorithms, e.g., MAML (Section 1.2.2), are not tailored for OCC, and why second-order derivatives are needed to optimize for such initializations. Thirdly, we propose an episode sampling technique that adapts any meta-learning algorithm that employs a bi-level optimization to the FS-OCC problem. Finally, we demonstrate the effectiveness of the proposed approach on eight datasets of images and time-series, including an industrial sensor readings dataset.

1.2.3 Continual Anomaly Detection

In this section, we first introduce the continual learning problem and its related works. Subsequently, we motivate and summarize our contribution from chapter 3 Frikha et al. (2021d).

Continual Learning

The ability to continuously learn new tasks without corrupting prior knowledge is a hallmark of human intelligence. When exposed to a sequence of tasks, deep learning methods exhibit a performance decrease on older tasks due to overwriting the previously acquired

knowledge. This problem is referred to as catastrophic forgetting McCloskey and Cohen (1989); French (1999); Goodrich (2015); Kirkpatrick et al. (2017). Continual learning (CL), also called lifelong learning, investigates ways of building models that are able to learn several tasks in an incremental fashion, i.e., reducing the impact of catastrophic forgetting. Ideally, the model should not only retain knowledge about past tasks but also leverage it while learning new tasks. Such models are essential for real-world applications where the data distribution changes frequently, e.g., quality control in industrial manufacturing, where the product portfolio is constantly evolving.

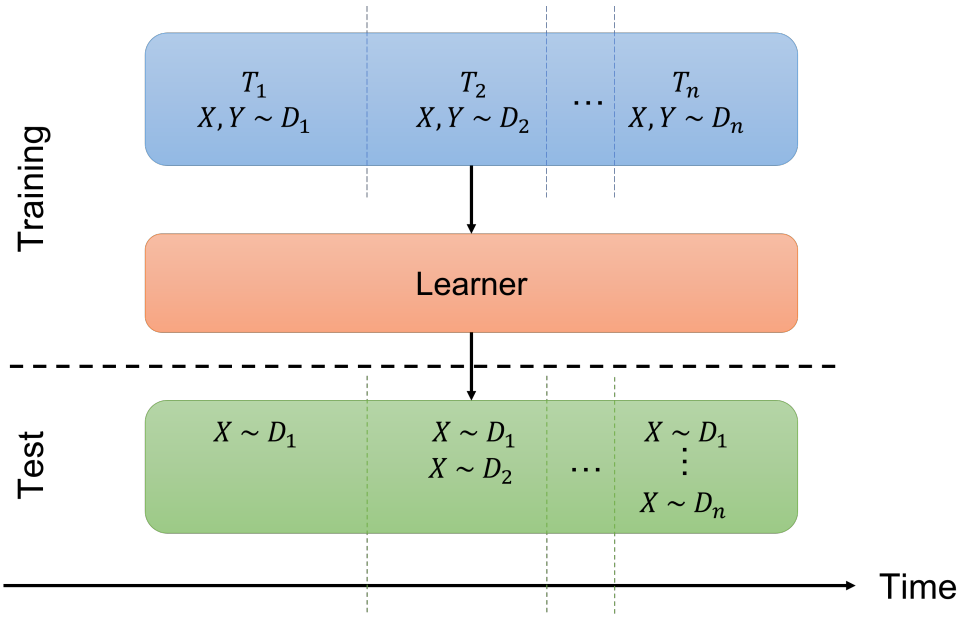


Figure 1.6: Schematic representation of the continual learning problem setting.

We present a schematic representation describing the continual learning problem setting, inspired by De Lange et al. (2021), in Figure 1.6. We focus on the continual learning framework introduced in Lopez-Paz and Ranzato (2017). Here, the learner is exposed to a series of different tasks $\mathcal{T}_i$ during training and can be evaluated at any time step, e.g., training iteration, on any of the tasks previously encountered. The evaluation is done using a test set, i.e., a separate set of data not seen during training, of the corresponding task. While some CL methods Lopez-Paz and Ranzato (2017) assume access to task descriptors, i.e., task identifiers that are fed to the learner at evaluation time, we address the more challenging setting where the learner needs to infer the task from the test samples Riemer et al. (2018).

The most commonly used performance metric used in CL is the *retained accuracy* (RA), which measures the task-averaged test performance after sequentially training on all tasks Lopez-Paz and Ranzato (2017). Other metrics include the *learning accuracy* (LA) and the *backward transfer and interference* (BTI). While LA computes the average test accuracy for each task immediately after training on it, BTI measures the average performance change for each task between the time step it was last learned and the end of the training, i.e., the difference between RA and LA. Note that a positive BTI indicates a positive transfer across the tasks, i.e., on average the tasks learned after task $\mathcal{T}_i$ were beneficial for the performance on that task. A negative BTI highlights catastrophic forgetting, i.e., the tasks learned after task $\mathcal{T}_i$ led to a deterioration of performance on that task.

Several works were conducted to investigate continual learning. We categorize the proposed approaches into four categories and refer to Parisi et al. (2019); Delange et al. (2021) for extensive reviews. The first category inhibits catastrophic forgetting by replaying experiences, e.g., data examples, from past tasks Schaul et al. (2015); Rebuffi et al. (2017); Lopez-Paz and Ranzato (2017); Riemer et al. (2018), while learning new tasks. Some approaches perform experience replay using synthetically generated examples Shin et al. (2017); Wang et al. (2018); Chaudhry et al. (2020). Another line of work regularizes parameter updates Kirkpatrick et al. (2017); Zenke et al. (2017); Lee et al. (2017) by penalizing changes to parameters that are important for previously learned tasks. Isolation-based methods prevent interference by assigning different model parameters for different tasks. This is done by increasing the model capacity Rusu et al. (2016); Aljundi et al. (2017); Xu and Zhu (2018), by using pruning masks Mallya and Lazebnik (2018), or learning sparse task-specific attention masks Serra et al. (2018). Finally, recent works tackle continual learning by leveraging meta-learning-based techniques which maximize transfer and minimize interference Riemer et al. (2018), learn an initialization for some Javed and White (2019); Beaulieu et al. (2020) or all Spigler (2019) model parameters, or continuously adapt class-prototypes Zhang et al. (2019).

Continual Anomaly Detection via Meta-Learning

In this section, we summarize the motivation, method, and findings of our published Frikha et al. (2021d) work presented in Chapter 3.

While the continual learning problem has been well-studied Parisi et al. (2019); Delange et al. (2021), the vast majority of works in this field focus on class-balanced classification. However, many real-world applications exhibit a high class-imbalance due to the rarity of

some categories, e.g., defective products in industrial manufacturing or the diagnosis of a rare disease in healthcare. As introduced in Section 1.2.1, anomaly detection problems are usually framed as one-class classification problems (OCC) Khan and Madden (2014), where only data from the normal class is available. To the best of our knowledge, we are the first to address the intersection of continual learning and anomaly detection, to which we refer to as Continual Anomaly Detection (CAD). CAD considers practical use-cases where a central anomaly detector for multiple applications is needed and new applications become available gradually in time. CAD is defined as a continual learning problem setting (Figure 1.6) where all the tasks are anomaly detection tasks, i.e., only data from their respective normal classes is available for training. The learner is expected to sequentially learn multiple anomaly detection classes, using only normal class examples from each task.

Our contribution in this work is threefold. Firstly, we introduce the novel and praxis-relevant continual anomaly detection problem and discuss its challenges: catastrophic forgetting and overfitting to the normal class. Secondly, we propose an effective and model-agnostic meta-learning approach to address CAD. Our method learns a learning strategy tailored for learning anomaly detection task-sequences with minimal forgetting. Finally, we empirically validate our approach on three datasets, where we significantly outperform previous class-balanced continual learning and anomaly detection methods.

1.3 Domain Generalization

This section presents an overview of research areas that are related to our contributions in chapters 4 and 5. Section 1.3.1 introduces the domain generalization problem which is relevant for both contributions. Thereafter, section 1.3.2 summarizes the motivation and contribution of our paper presented in chapter 4. Finally, in section 1.3.3, we present the data-free learning paradigm and summarize our contribution from chapter 5.

1.3.1 Introduction

While most deep learning achievements have been realized in the scenarios where the data is independent and identically distributed (i.d.d.), distribution shifts between training and testing are common in real-world applications. The performance of neural networks usually deteriorates substantially when evaluated on out-of-distribution (OOD) data Torralba and Efros (2011). In the healthcare sector, for instance, neural networks trained on MRI

images from one hospital fail to generalize to those from other hospitals that use different scanners Dou et al. (2019). Several domain adaptation (DA) approaches were developed to inhibit the performance degradation incurred by the domain shift present between the source domain used for training and the target domain faced at test time Wilson and Cook (2020).

While DA methods require access to a subset of data from the target domain, there exist scenarios where the target domain data is not available at development time. In fact, target domain data collection might be expensive, slow, e.g., in a cold start situation, or infeasible, e.g., collecting images from all streets in all countries to train autonomous vehicles. Sometimes, target domains cannot be known *a priori*. The Domain Generalization (DG) problem Blanchard et al. (2011); Muandet et al. (2013) was introduced to address such settings. In particular, a model trained on multiple source domains is directly evaluated on target domains without any modification or exposure to their data, i.e., there is no conditioning on the target domain. DG can be viewed as a zero-shot version of the domain adaptation problem.

We present a schematic representation describing the domain generalization problem setting in Figure 1.7. The DG problem assumes access to multiple datasets containing data from the same task but different domains. An example can be seen in Figure 1.2. These datasets are referred to as the *source domains* and can be used to train a model that is expected to be resilient to domain shift. In particular, the trained model is expected to perform well on data D_{target} from a new domain unseen during training, the *target domain*. Hereby the model is tested without any modification or adaptation to the target domain, which is unknown at training time, as opposed to domain adaptation.

In the last decade, a wide variety of DG approaches were proposed. In the following, we broadly classify them into three categories and refer to Zhou et al. (2021a) for an extensive review. The first category methods aim to align the different domains in a common embedding space by learning domain-invariant representations. The distribution mismatch between the domain-specific representations can be reduced by minimizing the distance between the means Tzeng et al. (2014), covariance matrices Sun and Saenko (2016) or the maximum mean discrepancy (MMD) criteria Gretton et al. (2012); Li et al. (2018b) across different domains in the embedding space. For the same purpose, contrastive learning techniques were employed as a regularization Motiian et al. (2017); Yoon et al. (2019); Mahajan et al. (2020), and the alignment of loss gradients across the domains using inner-product maximization Shi et al. (2021) or masking Parascandolo et al. (2020);

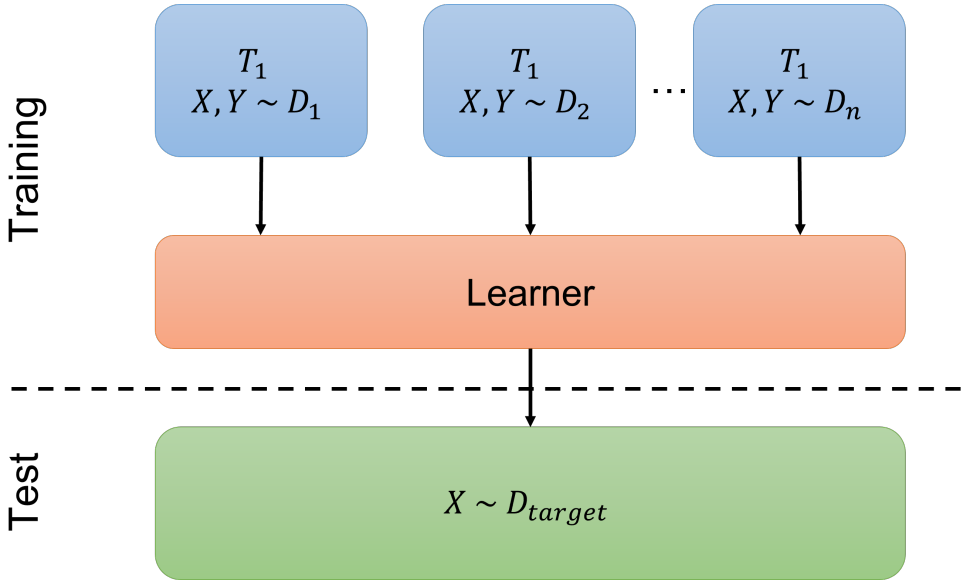


Figure 1.7: Schematic representation of the domain generalization problem setting.

Shahtalebi et al. (2021) was considered. Another line of work learn domain-invariant features by maximizing the error of a domain-discriminator model Ganin et al. (2016); Li et al. (2018c); Albuquerque et al. (2019); Shao et al. (2019); Rahman et al. (2020); Deng et al. (2020). The second category includes data augmentation approaches. While some methods augment the training data by leveraging Mixup Zhang et al. (2017) to synthesize cross-domain examples Xu et al. (2020); Yan et al. (2020); Wang et al. (2020b), others synthesize new images by using generative models to augment the source domains Rahman et al. (2019); Somavarapu et al. (2020); Borlino et al. (2021) or create novel domains Maria Carlucci et al. (2019); Zhou et al. (2020a,b). Adversarial perturbations Goodfellow et al. (2014b) were also used to perturb the training data based on the outputs of a class classifier Sinha et al. (2017); Volpi et al. (2018); Qiao et al. (2020) or a domain classifier Shankar et al. (2018), or to perturb learned representations of the data Huang et al. (2020); Zhou et al. (2021b). Finally, meta-learning methods (Section 1.2) that involve a bi-level optimization scheme were developed to address DG, by learning a regularizer of the last layer Balaji et al. (2018), explicitly optimizing for quick adaptation Li et al. (2018a), and regularizing the embedding space via inter-class and intra-class similarity optimization Dou et al. (2019).

1.3.2 Multi-Level Feature Discovery via Corruption

In this section, we summarize the motivation and contributions of our published work Frikha et al. (2021c) presented in Chapter 4.

Highly parametrized deep learning models trained with gradient-descent achieve impressive performance in a variety of tasks including computer vision Guo et al. (2016) and natural language processing Chai and Li (2019), sometimes even outperforming humans Silver et al. (2018); Madani et al. (2018). However, these models were found to rely on learning only a subset of (spurious) features, failing to capture further (more predictive) features present in the training data Geirhos et al. (2018); Shah et al. (2020); Pezeshki et al. (2021). This results in a performance deterioration when exposed to data that exhibits domain shift. Many terms are used to describe this phenomenon including shortcut learning Geirhos et al. (2018), simplicity bias Shah et al. (2020) and gradient starvation Pezeshki et al. (2021).

In our work, we propose a domain generalization (DG) approach that addresses this phenomenon by incentivizing the model to capture as many features as possible. This is based on the assumption that a richer set of features improves the knowledge transfer to a wider variety of unseen domains. Our method leverages methods from the explainable machine learning literature to identify the features captured by the model. Thereafter, these learned features are corrupted and the model is trained on the corrupted version of the data, hence enforcing new feature discovery. We evaluated our method on a DG testbed Gulrajani and Lopez-Paz (2020) that fairly compares DG algorithms by including the same pre-processing pipeline and hyperparameter search. We found that our algorithm outperforms 18 DG approaches on three different DG benchmark datasets.

1.3.3 Data-Free Domain Generalization

This section introduces the data-free learning paradigm including the motivation behind it and its related works. Thereafter, we provide an overview of our contribution presented in chapter 5.

Data-Free Learning

While machine learning methods require data to learn, in many real-world scenarios, data access is not possible. For instance, some companies might not be willing to share their data to avoid commercial disadvantage and/or reverse engineering. Moreover, General

Data Protection Regulations (GDPR) disallow the usage of personal information from individuals, e.g., biometric data or other confidential information. Likewise, some data might be accessible due to security or safety concerns. Furthermore, as datasets become larger and larger, their release and transfer become utterly costly. While federated learning McMahan et al. (2017) techniques enable learning from decentralized data and were extensively studied in this context Kairouz et al. (2021), we focus on the alternative more recent family of approaches of data-free learning Liu et al. (2021b).

In data-free learning, we consider the scenario where the data owners are willing to share a model trained on their data instead of releasing the original dataset. Recently, this setting has enjoyed a surge of interest in the machine learning research community Micaelli and Storkey (2019); Chen et al. (2019); Nayak et al. (2019); Liang et al. (2020); Li et al. (2020b); Kundu et al. (2020); Yin et al. (2020); Ahmed et al. (2021). Data-Free Knowledge Distillation methods were proposed to transfer knowledge from one Micaelli and Storkey (2019); Nayak et al. (2019); Chen et al. (2019); Yin et al. (2020); Zhang et al. (2021) or several trained teacher models Li et al. (2021b) to other untrained student models without any access to the original data. This is done by training the student model on synthetic data generated by a generative model Micaelli and Storkey (2019); Chen et al. (2019) or via Inceptionism-style Mahendran and Vedaldi (2015) image synthesis Nayak et al. (2019); Yin et al. (2020); Zhang et al. (2021), i.e., optimization of random noise examples to be recognized by trained models. We note that in classical knowledge distillation Hinton et al. (2015) the original dataset is used to distill the teacher knowledge into the student model.

While the aforementioned DFKD methods address domain-specific scenarios, many real-world applications exhibit domain shift between training and test data. Recently, the Source-Free Domain Adaptation (SFDA) problem Liang et al. (2020); Li et al. (2020b); Kundu et al. (2020) was proposed to investigate data-free learning settings involving domain shift. In particular, it addresses the situation where one or multiple models trained on the source domains are available instead of the data itself, along with data from the target domain.

The approaches developed to tackle SFDA rely on weighting the target domain examples by their similarity to the source domains Kundu et al. (2020), combining generative models with a regularization loss Li et al. (2020b), pairing an information maximization loss with pseudo-labeling Liang et al. (2020); Ahmed et al. (2021), or replacing the source-domain batch normalization Ioffe and Szegedy (2015) statistics with those computed on the target domain examples Li et al. (2016). Recently, federated learning approaches were developed

to cope with domain shift Li et al. (2021a); Liu et al. (2021a); Chen et al. (2022).

Data-Free Domain Generalization via Multi-Teacher Knowledge Amalgamation

In this section, we summarize the motivation and contributions of our published work Frikha et al. (2021a) presented in Chapter 5.

On the one hand, the aforementioned DFKD methods were designed to tackle domain-specific single-teacher scenarios with no domain shift. On the other hand, the previously mentioned SFDA methods require access to data from target domains. In our work, we address the unexplored intersection of domain generalization and data-free learning, which we define as the Data-Free Domain Generalization (DFDG) problem. DFDG investigates the practical setting where a model that is robust to domain shift is needed and only models trained on the source domains are available. The key difference to the SFDA problem is the absence of the target domain data, which is motivated by the fact that in many real-world scenarios the target domains are not known *a priori* and there is no access to their data.

The contribution of our work on DFDG is threefold: Firstly, we introduce and define the novel DFDG problem. Secondly, we propose Domain Entanglement via Knowledge Amalgamation from domain-specific Networks (DEKAN), an effective approach for this problem, as well as several baseline methods. Our algorithm extracts and merges the knowledge contained in the available domain-specific teacher model by generating domain-specific and cross-domain synthetic examples. The latter are optimized by maximizing the agreement of different domain-specific teachers and minimizing a cross-domain feature distribution matching loss. The generated images are then used to transfer the knowledge to a student model via multi-teacher knowledge distillation. The student is tested on the target domain without any modification or prior exposure to their data. Thirdly, we evaluate DEKAN on 2 DG benchmark datasets and find that it outperforms all the baselines including ensemble-based and multi-teacher DFKD methods, hence achieving state-of-the-art results on this challenging problem. Moreover, DEKAN substantially reduces the gap between the best DFDG baseline and the upper-bound oracle method that uses the original source domain data.

Chapter 2

Few-Shot One-Class Classification via Meta-Learning

The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

Few-Shot One-Class Classification via Meta-Learning

Ahmed Frikha [1, 2, 4], **Denis Krompaß** [1, 2], **Hans-Georg Köpken** [3], **Volker Tresp** [2, 4]
[1]Siemens AI Lab
[2]Siemens Technology
[3]Siemens Digital Industries
[4]Ludwig Maximilian University of Munich
ahmed.frikha@siemens.com

Abstract

Although few-shot learning and one-class classification (OCC), i.e., learning a binary classifier with data from only one class, have been separately well studied, their intersection remains rather unexplored. Our work addresses the few-shot OCC problem and presents a method to modify the episodic data sampling strategy of the model-agnostic meta-learning (MAML) algorithm to learn a model initialization particularly suited for learning few-shot OCC tasks. This is done by explicitly optimizing for an initialization which only requires few gradient steps with one-class minibatches to yield a performance increase on class-balanced test data. We provide a theoretical analysis that explains why our approach works in the few-shot OCC scenario, while other meta-learning algorithms fail, including the unmodified MAML. Our experiments on eight datasets from the image and time-series domains show that our method leads to better results than classical OCC and few-shot classification approaches, and demonstrate the ability to learn unseen tasks from only few normal class samples. Moreover, we successfully train anomaly detectors for a real-world application on sensor readings recorded during industrial manufacturing of workpieces with a CNC milling machine, by using few normal examples. Finally, we empirically demonstrate that the proposed data sampling technique increases the performance of more recent meta-learning algorithms in few-shot OCC and yields state-of-the-art results in this problem setting.

Introduction

The anomaly detection (AD) task (Chandola, Banerjee, and Kumar 2009; Aggarwal 2015) consists in differentiating between normal and abnormal data samples. AD applications are common in various domains that involve different data types, including medical diagnosis (Prastawa et al. 2004), cybersecurity (Garcia-Teodoro et al. 2009) and quality control in industrial manufacturing (Scime and Beuth 2018). Due to the rarity of anomalies, the data underlying AD problems exhibits high class-imbalance. Therefore, AD problems are usually formulated as one-class classification (OCC) problems (Moya, Koch, and Hostetler 1993), where either only a few or no anomalous data samples are available for training the model (Khan and Madden 2014). While most of the developed approaches (Khan and Madden 2014) require a substantial amount of normal data to yield good generalization, in many real-world applications, e.g., in industrial manufacturing, only small datasets are available. Data scarcity can have many reasons: data collection itself might be expensive, e.g., in healthcare, or happens only gradually, such as in a cold-start situation, or the domain expertise required for annotation is scarce and expensive.

To enable learning from few examples, viable approaches (Lake et al. 2011; Ravi and Larochelle 2017; Finn, Abbeel, and Levine 2017) relying on meta-learning (Schmidhuber 1987) have been developed. However, they rely on having examples from each of the task's classes, which prevents their application to OCC tasks. While recent meta-learning approaches focused on the few-shot learning problem, i.e., learning to learn with few examples, we extend their use to the OCC problem, i.e., learning to learn with examples from only one class. To the best of our knowledge, the few-shot OCC (FS-OCC) problem has only been addressed in (Kozerawski and Turk 2018; Kruspe 2019) in the image domain.

Our contribution is fourfold: Firstly, we show that classical OCC approaches fail in the few-shot data regime. Secondly, we provide a theoretical analysis showing that classical gradient-based meta-learning algorithms do not yield parameter initializations suitable for OCC and that second-order derivatives are needed to optimize for such initializations. Thirdly, we propose a simple episode generation strategy to adapt any meta-learning algorithm that uses a bi-level optimization scheme to FS-OCC. Hereby, we first focus on modifying the model-agnostic meta-learning (MAML) algorithm (Finn, Abbeel, and Levine 2017) to learn initializations useful for the FS-OCC scenario. The resulting One-Class MAML (OC-MAML) maximizes the inner product of loss gradients computed on one-class and class-balanced minibatches, hence maximizing the cosine similarity between these gradients. Finally, we demonstrate that the proposed data sampling technique generalizes beyond MAML to other metalearning algorithms, e.g., MetaOptNet (Lee et al. 2019) and Meta-SGD (Li et al. 2017), by successfully adapting them to the understudied FS-OCC.

We empirically validate our approach on eight datasets from the image and time-series domains, and demonstrate its robustness and maturity for real-world applications by successfully testing it on a real-world dataset of sensor read-

ings recorded during manufacturing of metal workpieces with a CNC milling machine. Furthermore, we outperform the concurrent work One-Way ProtoNets (Kruspe 2019) and achieve state-of-the-art performance in FS-OCC.

Approach

The primary contribution of our work is to propose a way to adapt meta-learning algorithms designed for class-balanced FS learning to the underexplored FS-OCC problem. In this section, as a first demonstration that meta-learning is a viable approach to this challenging learning scenario, we focus on investigating it on the MAML algorithm. MAML was shown to be a universal learning algorithm approximator (**?**), i.e., it could approximate a learning algorithm tailored for FS-OCC. Later, we validate our methods on further meta-learning algorithms (Table 4).

Problem Statement

Our goal is to learn a one-class classification task using only a *few* examples. In the following, we first discuss the unique challenges of the few-shot one-class classification (FS-OCC) problem. Subsequently, we discuss the formulation of the FS-OCC problem as a meta-learning problem.

To perform one-class classification, i.e., differentiate between in-class and out-of-class examples using only in-class data, approximating a *generalized* decision boundary for the normal class is necessary. Learning such a class decision boundary in the few-shot regime can be especially challenging for the following reasons. On the one hand, if the model overfits to the few available datapoints, the class decision boundary would be too restrictive, which would prevent generalization to unseen examples. As a result, some normal samples would be predicted as anomalies. On the other hand, if the model overfits to the majority class, i.e., predicting almost everything as normal, the class decision boundary would overgeneralize, and out-of-class (anomalous) examples would not be detected.

In the FS classification context, N-way K-shot learning tasks are used to test the learning procedure yielded by the meta-learning algorithm. An N-way K-shot classification task includes K examples from *each* of the N classes that are used for learning this task, after which the trained classifier is tested on a disjoint set of data (Vinyals et al. 2016). When the target task is an OCC task, only examples from one class are available for training, which can be viewed as a 1-way K-shot classification task. To align with the anomaly detection problem, the available examples must belong to the normal (majority) class, which usually has a lower variance than the anomalous (minority) class. This problem formulation is a prototype for a practical use case where an application-specific anomaly detector is needed and only few normal examples are available.

Model-Agnostic Meta-Learning

MAML is a meta-learning algorithm that we focus on adapting to the FS-OCC problem before validating our approach on further meta-learning algorithms (Table 4). MAML learns a model initialization that enables quick adaptation to unseen tasks using only few data samples. For that, it trains a model explicitly for few-shot learning on tasks T_i coming from the same task distribution $p(T)$ as the unseen target task T_{test}. In order to assess the model's adaptation ability to *unseen* tasks, the available tasks are divided into mutually disjoint task sets: one for meta-training S^{tr}, one for meta-validation S^{val} and one for meta-testing S^{test}. Each task T_i is divided into two disjoint sets of data, each of which is used for a particular MAML operation: D^{tr} is used for adaptation and D^{val} is used for validation, i.e., evaluating the adaptation. The adaptation of a model f_θ to a task T_i consists in taking few gradient descent steps using *few* datapoints sampled from D^{tr} yielding $\theta_i^{'}$.

A good measure for the suitability of the initialization parameters θ for few-shot adaptation to a considered task T_i is the loss $L_{T_i}^{val}(f_{\theta_i^{'}})$, which is computed on the validation set D_i^{val} using the task-specific adapted model $f_{\theta_i^{'}}$. To optimize for few-shot learning, the model parameters θ are updated by minimizing the aforementioned loss across all meta-training tasks. This *meta-update*, can be expressed as:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i \sim p(T)} L_{T_i}^{val}(f_{\theta_i^{'}}). \tag{1}$$

Here β is the learning rate used for the meta-update. To avoid overfitting to the meta-training tasks, model selection is done via validation using tasks from S^{val}. At meta-test time, the FS adaptation to unseen tasks from S^{test} is evaluated. We note that, in the case of few-shot classification, K datapoints from *each* class are sampled from D^{tr} for the adaptation, during training and testing.

One-Class Model-Agnostic Meta-Learning

Algorithm. MAML learns a model initialization suitable for *class-balanced* (CB) FS classification. To adapt it to FS-OCC, we aim to find a model initialization from which taking few gradients steps with a few one-class (OC) examples yields the same effect as doing so with a CB minibatch. We achieve this by adequately modifying the objective of the inner loop updates of MAML. Concretely, this is done by modifying the data sampling technique during meta-training, so that the class-imbalance rate (CIR) of the inner loop minibatches matches the one of the test task.

MAML optimizes explicitly for FS adaptation by creating and using auxiliary tasks that have the same characteristic as the target tasks, in this case tasks that include only few datapoints for training. It does so by reducing the size of the batch used for the adaptation (via the hyperparameter K (**?**)). Analogously, OC-MAML trains explicitly for quick adaptation to OCC tasks by creating OCC auxiliary tasks for meta-training. OCC problems are binary classification scenarios where only few or no minority class samples are available. In order to address both of theses cases, we introduce a hyperparameter (c) which sets the CIR of the batch sampled for the inner updates. Hereby, c gives the percentage of the samples belonging to the minority (anomalous) class w.r.t. the total number of samples, e.g., setting $c = 0\%$ means only majority class samples are contained in the data batch. We focus on this extreme case, where no anomalous

Algorithm 1 Meta-training of OC-MAML

Require: S^{tr}: Set of meta-training tasks
Require: α, β: Learning rates
Require: K, Q: Batch size for the inner and outer updates
Require: c: CIR for the inner-updates
1: Randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks T_i from S^{tr}; $T_i = \{D^{tr}, D^{val}\}$
4: **for each** sampled T_i **do**
5: Sample K examples B from D^{tr} such that CIR$= c$
6: Initialize $\theta_i^{'} = \theta$
7: **for** number of adaptation steps **do**
8: Compute adapted parameters with gradient descent using B: $\theta_i^{'} = \theta_i^{'} - \alpha \nabla_{\theta_i^{'}} L_{T_i}^{tr}(f_{\theta_i^{'}})$
9: **end for**
10: Sample Q examples $B^{'}$ from D^{val} w/ CIR$= 50\%$
11: Compute outer loop loss $L_{T_i}^{val}(f_{\theta_i^{'}})$ using $B^{'}$
12: **end for**
13: Update θ: $\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i} L_{T_i}^{val}(f_{\theta_i^{'}})$
14: **end while**
15: **return** meta-learned parameters θ

samples are available for learning. In order to evaluate the performance of the adapted model on both classes, we use a *class-balanced* validation batch $B^{'}$ for the meta-update. This way, we maximize the performance of the model in recognizing both classes after having *seen* examples from only one class during adaptation. The OC-MAML meta-training is described in Algorithm 1, and the cross-entropy loss was used for L. At test time, the adaptation to an unseen task is done by applying steps 5-9 in Algorithm 1, starting from the meta-learned initialization.

We note that the proposed episode sampling strategy, i.e., training on a one-class batch then using the loss computed on a class-balanced validation batch to update the meta-learning strategy (e.g., model initialization), is applicable to any meta-learning algorithm that incorporates a bi-level optimization scheme (examples in Table 4).

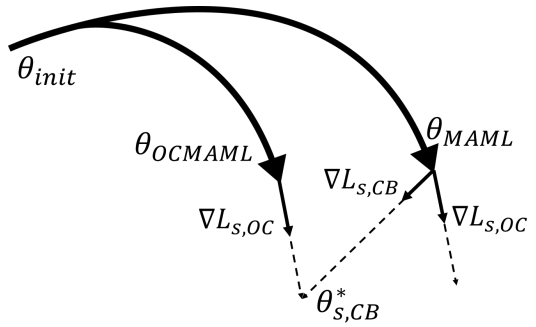


Figure 1: Adaptation to task T_s from the model initializations yielded by OC-MAML and MAML

Using OCC tasks for adaptation during meta-training favors model initializations that enable a quick adaptation to OCC tasks over those that require CB tasks. The schematic visualization in Figure 1 shows the difference between the model initializations meta-learned by MAML and OC-MAML. Hereby, we consider the adaptation to an unseen binary classification task T_s. $\theta^*_{s,CB}$ denotes a local optimum of T_s. The parameter initializations yielded by OC-MAML and MAML are denoted by θ_{OCMAML} and θ_{MAML} respectively. When starting from the OC-MAML parameter initialization, taking a gradient step using an OC support set $D_{s,OC}$ (gradient direction denoted by $\nabla L_{s,OC}$), yields a performance increase on T_s (by moving closer to the local optimum). In contrast, when starting from the parameter initialization reached by MAML, a class-balanced support set $D_{s,CB}$ (gradient direction denoted by $\nabla L_{s,CB}$) is required for a performance increase on T_s.

Theoretical Analysis: Why Does OC-MAML Work ? In this section we give a theoretical explanation of why OC-MAML works and why it is a more suitable approach than MAML for the FS-OCC setting. To address the latter problem, we aim to find a model parameter initialization, from which adaptation using few data examples from only *one* class yields a good performance on both classes, i.e., good generalization to the class-balanced task. We additionally demonstrate that adapting first-order meta-learning algorithms, e.g., First-Order MAML (FOMAML) (Finn, Abbeel, and Levine 2017) and Reptile (Nichol and Schulman 2018), to the OCC scenario as done in OC-MAML, does not yield initializations with the desired characteristics.

By using a Taylor series expansion the gradient used in the MAML update can be approximated to Equation 2 (Nichol and Schulman 2018), where the case with only 2 gradient-based updates is considered, i.e., one adaptation update on a minibatch (1), the support set including K examples from D^{tr}, and one meta-update on a minibatch (2), the query set including Q examples from D^{val}. We use the notation from (Nichol and Schulman 2018), where $\overline{g}_i$ and $\overline{H}_i$ denote the gradient and Hessian computed on the i^{th} minibatch at the initial parameter point ϕ_1, and α the learning rate. Here it is assumed that the same learning rate is used for the adaptation and meta-updates.

$$\begin{aligned} g_{MAML} &= \overline{g}_2 - \alpha \overline{H}_2 \overline{g}_1 - \alpha \overline{H}_1 \overline{g}_2 + O(\alpha^2) \\ &= \overline{g}_2 - \alpha \frac{\partial(\overline{g}_1 . \overline{g}_2)}{\partial \phi_1} + O(\alpha^2) \end{aligned} \tag{2}$$

Equation 2 shows that MAML maximizes the inner product of the gradients computed on different minibatches (Nichol and Schulman 2018). Under the assumption of local linearity of the loss function (which is the case around small optimization steps), and when gradients from different minibatches have a positive inner product, taking a gradient step using one minibatch yields a performance increase on the other (Nichol and Schulman 2018). Maximizing the inner product leads to a decrease in the angle between the gradient vectors and thus to an increase in their cosine similarity. Hence, MAML optimizes for an initialization where gradients computed on *small* minibatches have similar directions, which enables few-shot learning.

Equation 2 is independent of the data strategy adopted and

hence holds also for OC-MAML. However, in OC-MAML the minibatches 1 and 2 have different class-imbalance rates (CIRs), since the first minibatch includes examples from only one class and the second minibatch is class-balanced. So, it optimizes for increasing the inner product between a gradient computed on a one-class minibatch and a gradient computed on class-balanced data. Thus, OC-MAML optimizes for an initialization where gradients computed on one-class data have similar directions, i.e., a high inner product and therefore a high cosine similarity, to gradients computed on class-balanced data (Figure 1). Consequently, taking one (or few) gradient step(s) with one-class minibatch(es) from such a parameter initialization results in a performance increase on class-balanced data. This enables one-class classification. In contrast, MAML uses only class-balanced data during meta-training, which leads to a parameter initialization that requires class-balanced minibatches to yield the same effect. When adapting to OCC tasks, however, only examples from one class are available. We conclude, therefore, that the proposed data sampling technique modifies MAML to learn parameter initializations that are more suitable for adapting to OCC tasks.

A natural question is whether applying the same data sampling method to other gradient-based meta-learning algorithms would yield the same desired effect. We investigate this for First-Order MAML (FOMAML), a first-order approximation of MAML that ignores the second derivative terms and Reptile (Nichol and Schulman 2018), which is also a first-order meta-learning algorithm that learns an initialization that enables fast adaptation to test tasks using few examples from *each* class. We refer to the versions of these algorithms adapted to the FS-OCC setting as OC-FOMAML and OC-Reptile. We note that for OC-Reptile, the first $N-1$ batches contain examples from only one class and the last (N^{th}) batch is class-balanced. The approximated FOMAML and Reptile gradients are given by Equations 3 and 4 (Nichol and Schulman 2018), respectively.

$$g_{FOMAML} = \overline{g}_2 - \alpha \overline{H}_2 \overline{g}_1 + O(\alpha^2) \tag{3}$$

$$g_{Reptile} = \overline{g}_1 + \overline{g}_2 - \alpha \overline{H}_2 \overline{g}_1 + O(\alpha^2) \tag{4}$$

We note that these equations hold also for OC-FOMAML and OC-Reptile. By taking the expectation over minibatch sampling $\mathbb{E}_{\tau,1,2}$ for a task τ and two *class-balanced* minibatches 1 and 2, it is established that $\mathbb{E}_{\tau,1,2}[\overline{H}_1 \overline{g}_2] = \mathbb{E}_{\tau,1,2}[\overline{H}_2 \overline{g}_1]$ (Nichol and Schulman 2018). Averaging the two sides of the latter equation results in

$$\begin{aligned} \mathbb{E}_{\tau,1,2}[\overline{H}_2 \overline{g}_1] &= \frac{1}{2}\mathbb{E}_{\tau,1,2}[\overline{H}_1 \overline{g}_2 + \overline{H}_2 \overline{g}_1] \\ &= \frac{1}{2}\mathbb{E}_{\tau,1,2}[\frac{\partial(\overline{g}_1 . \overline{g}_2)}{\partial \phi_1}]. \end{aligned} \tag{5}$$

Equation 5 shows that, FOMAML and Reptile, like MAML, in expectation optimize for increasing the inner product of the gradients computed on different minibatches with the *same* CIR. However, when the minibatches 1 and 2 have different CIRs, which is the case for OC-FOMAML and OC-Reptile, $\mathbb{E}_{\tau,1,2}[\overline{H}_1 \overline{g}_2] \neq \mathbb{E}_{\tau,1,2}[\overline{H}_2 \overline{g}_1]$ and therefore $\mathbb{E}_{\tau,1,2}[\overline{H}_2 \overline{g}_1] \neq \frac{1}{2}\mathbb{E}_{\tau,1,2}[\frac{\partial(\overline{g}_1 . \overline{g}_2)}{\partial \phi_1}]$. Hence, despite using the same data sampling method as OC-MAML, OC-FOMAML and OC-Reptile do *not* explicitly optimize for increasing the inner product, and therefore the cosine similarity, between gradients computed on one-class and class-balanced minibatches. The second derivative term $\overline{H}_1 \overline{g}_2$ is, thus, necessary to optimize for an initialization from which performance increase on a class-balanced task is yielded by taking few gradient steps using one class data.

Related Works

Our proposed method addresses the FS-OCC problem, i.e., solving binary classification problems using only *few* datapoints from only *one* class. To the best of our knowledge, this problem was only addressed in (Kozerawski and Turk 2018) and (Kruspe 2019), and exclusively in the image data domain. In (Kozerawski and Turk 2018) a feed-forward neural network is trained on ILSVRC 2012 to learn a transformation from feature vectors, extracted by a CNN pre-trained on ILSVRC 2014 (Russakovsky et al. 2015), to SVM decision boundaries. At test time, an SVM boundary is inferred by using one image of one class from the test task which is then used to classify the test examples. This approach is specific to the image domain since it relies on the availability of very large, well annotated datasets and uses data augmentation techniques specific to the image domain, e.g., mirroring. Meta-learning algorithms offer a more general approach to FS-OCC since they are data-domain-agnostic, and do not require a pre-trained feature extraction model, which may not be available for some data domains, e.g., sensor readings.

The concurrent work One-Way ProtoNets (Kruspe 2019) adapts ProtoNets (Snell, Swersky, and Zemel 2017) to address FS-OCC by using 0 as a prototype for the *null* class, i.e., non-normal examples, since the embedding space is 0-centered due to using batch normalization (BN) (Ioffe and Szegedy 2015) as the last layer. Given the embedding of a query example, its distance to the normal-class prototype is compared to its norm. This method constraints the model architecture by requiring the usage of BN layers. We propose a model-architecture agnostic data sampling technique to adapt meta-learning algorithms to the FS-OCC problem. The resulting meta-learning algorithms substantially outperform One-Way ProtoNets (Kruspe 2019) (Table 4).

Class-Balanced Few-Shot Classification

Meta-learning approaches for FS classification approaches may be broadly categorized in 2 categories. Optimization-based approaches aim to learn an optimization algorithm (Ravi and Larochelle 2017) and/or a parameter initialization (Finn, Abbeel, and Levine 2017; Nichol and Schulman 2018), learning rates (Li et al. 2017), an embedding network (Lee et al. 2019) that are tailored for FS learning. Metric-based techniques learn a metric space where samples belonging to the same class are close together, which facilitates few-shot classification (**?**Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Sung et al. 2018; Oreshkin, López, and Lacoste 2018; Lee et al. 2019). Hybrid methods (**?**Lee and

Choi 2018) combine the advantages of both categories. Prior meta-learning approaches to FS classification addressed the *N*-way *K*-shot classification problem described in the problem statement section, i.e they require examples from *each* class of the test tasks. We propose a method to adapt meta-learning algorithm to the *1*-way *K*-shot scenario, where only few examples from *one* class are available.

One-Class Classification

Classical OCC approaches rely on SVMs (Schölkopf et al. 2001; Tax and Duin 2004) to distinguish between normal and abnormal samples. Hybrid approaches combining SVM-based techniques with feature extractors were developed to compress the input data in lower dimensional representations (Xu et al. 2015; Erfani et al. 2016; Andrews et al. 2016). Fully deep methods that jointly perform the feature extraction step and the OCC step have also been developed (Ruff et al. 2018). Another category of approaches to OCC uses the reconstruction error of antoencoders (Hinton and Salakhutdinov 2006) trained with only normal examples as an anomaly score (Hawkins et al. 2002; An and Cho 2015; Chen et al. 2017). Yet, determining a decision threshold for such an anomaly score requires labeled data from both classes. Other techniques rely on GANs (Goodfellow et al. 2014) to perform OCC (Schlegl et al. 2017; Ravanbakhsh et al. 2017; Sabokrou et al. 2018). The aforementioned hybrid and fully deep approaches require a considerable amount of data from the OCC task to train the typically highly parametrized feature extractors specific to the normal class, and hence fail in the scarce data regime (Table 1).

Experimental Evaluation

The conducted experiments [1] use some modules of the pyMeta library (Spigler 2019) and aim to address the following key questions: (a) How do meta-learning-based approaches using the proposed episode sampling technique perform compared to classical OCC approaches in the few-shot (FS) data regime? (b) Do the findings of our theoretical analysis about the differences between the MAML and OC-MAML initializations hold in practice? (c) Does the proposed episode sampling strategy to adapt MAML to the FS-OCC setting yield the expected performance increase and does this hold for further meta-learning algorithms?

Baselines and Datasets

We compare OC-MAML, with the classical OCC approaches One-Class SVM (OC-SVM) (Schölkopf et al. 2001) and Isolation Forest (IF) (Liu, Ting, and Zhou 2008) (Question (a)), which we fit to raw features and embeddings of the support set of the test task. Here, we explore two types of embedding networks which are trained on the meta-training tasks as follows: one is trained in a Multi-Task-Learning (MTL) (Caruana 1997) setting using one-class-vs-all tasks and the other trained using the "Finetune" baseline (FB) (Triantafillou et al. 2019). i.e., using multi-class classification on all classes available.

[1] Code available under https://github.com/AhmedFrikha/Few-Shot-One-Class-Classification-via-Meta-Learning

Moreover, we compare first-order (FOMAML and Reptile) and second-order (MAML) class-balanced meta-learning algorithms to their adapted versions to the OCC scenario, i.e., OC-FOMAML and OC-Reptile and OC-MAML (Question (b)). Finally, we compare MetaOptNet (Lee et al. 2019) and meta-SGD (Li et al. 2017) to their one-class counterparts that use our sampling strategy (Question (c)). We conducted a hyperparameter search for each baseline separately and used the best performing setting for our experiments. We evaluate our approach on 8 datasets from the image and time-series data domains, including two synthetic time-series (STS) datasets that we propose as a benchmark for FS-OCC on time-series, and a real-world sensor readings dataset of CNC Milling Machine Data (CNC-MMD). To adapt the image datasets to the OCC scenario, we create binary classification tasks, where the normal class is one class of the initial dataset and the anomalous class contains examples from *multiple* other classes.

Results and Discussion

In this section, we first discuss the performance of classical OCC approaches and the meta-learning algorithms in the FS-OCC problem setting, as well as the impact of the proposed data sampling strategy. Subsequently, we demonstrate the maturity of our approach on a real-world dataset. Thereafter, we further confirm our theoretical analysis with empirical results of cosine similarity between gradients. Finally, we show the generalizability of our sampling technique to further meta-learning algorithms beyond MAML, and compare the resulting algorithms to One-Way ProtoNets.

Table 1 shows the results averaged over 5 seeds of the classical OCC approaches (Top) and the meta-learning approaches, namely MAML, FOMAML, Reptile and their one-class versions (Bottom), on 3 image datasets and on the STS-Sawtooth dataset. For the meta-learning approaches, models were trained with and without BN layers and the results of the best architecture were reported for each dataset. The results of all the methods on the other 8 MT-MNIST task-combinations and on the STS-Sine dataset, are consistent with the results in Table 1.

While classical OCC methods yield chance performance in almost all settings, OC-MAML achieves very high results, consistently outperforming them across all datasets and on both support set sizes. Likewise, we observe that OC-MAML consistently outperforms the class-balanced and one-class versions of the meta-learning algorithms in all the settings, showing the benefits of our modification to MAML.

Moreover, OC-FOMAML and OC-Reptile yield poor results, especially without BN, confirming our theoretical findings that adapting first-order meta-learning algorithms to the OCC setting does not yield the desired effect. We found that using BN yields a substantial performance increase on the 3 image datasets and explain that by the gradient orthogonalizing effect of BN (Suteu and Guo 2019). In fact, gradient orthogonalization reduces interference between gradients computed on one-class and class-balanced batches. OC-MAML achieves high performance even without BN, as it reduces interference between these gradients by the means of its optimization objective (see theoretical analysis).

Adaptation set size	$K = 2$				$K = 10$			
Model \ Dataset	MIN	Omn	MNIST	Saw	MIN	Omn	MNIST	Saw
FB	50.0	50.6	56.5	50.0	50.0	51.2	50.3	50.0
MTL	50.0	50.0	49.7	50.0	50.2	50.0	45.3	50.0
OC-SVM	50.2	50.6	51.2	50.1	51.2	50.4	53.6	50.5
IF	50.0	50.0	50.0	50.0	50.7	50.0	50.9	49.9
FB + OCSVM	50.0	50.0	55.5	50.4	51.4	58.0	86.6	58.3
FB + IF	50.0	50.0	50.0	50.0	50.0	50.0	76.1	51.5
MTL + OCSVM	50.0	50.0	50.0	50.0	50.0	50.1	53.8	86.9
MTL + IF	50.0	50.0	50.0	50.0	50.0	55.7	84.2	64.0
Reptile	51.6	56.3	71.1	69.1	57.1	76.3	89.8	81.6
FOMAML	53.3	78.8	80.7	75.1	59.5	93.7	91.1	80.2
MAML	62.3	91.4	85.5	81.1	65.5	96.3	92.2	86
OC-Reptile	51.9	52.1	51.3	51.6	53.2	51	51.4	53.2
OC-FOMAML	55.7	74.7	79.1	58.6	66.1	87.5	91.8	73.2
OC-MAML (ours)	**69.1**	**96.6**	**88**	**96.6**	**76.2**	**97.6**	**95.1**	**95.7**

Table 1: Accuracies (in $\%$) computed on the class-balanced test sets of the test tasks of MiniImageNet (MIN), Omniglot (Omn), MT-MNIST with $T_{test} = T_0$ and STS-Sawtooth (Saw).

Several previous meta-learning approaches, e.g., MAML (Finn, Abbeel, and Levine 2017), were evaluated in a transductive setting, i.e., the model classifies the whole test set at once which enables sharing information between test examples via BN (Nichol and Schulman 2018). In anomaly detection applications, the CIR of the encountered test set batches, and therefore the statistics used in BN layers, can massively change depending on the system behavior (normal or anomalous). Hence, we evaluate all methods in a non-transductive setting: we compute the statistics of all BN layers using the few one-class adaptation examples and use them for predictions on test examples. This is equivalent to classifying each test example separately. We also use this method during meta-training. We note that the choice of the BN scheme heavily impacts the performance of several meta-learning algorithms (Bronskill et al. 2020).

Validation on the CNC-Milling Real-World Dataset. We validate OC-MAML on the industrial sensor readings dataset CNC-MDD and report the results in Table 2. We compute F1-scores for evaluation since the test sets are class-imbalanced. Depending on the type of the target milling operation (e.g., roughing), tasks created from *different* operations from the same type are used for meta-training. OC-MAML consistently achieves high F1-scores between 80% and 95.9% across the 6 milling processes. The high performance on the minority class, i.e., in detecting anomalous data samples, is reached by using only $K = 10$ non-anomalous examples ($c = 0\%$). These results show that OC-MAML yielded a parameter initialization suitable for learning OCC tasks in the time-series data domain and the maturity of this method for industrial real-world applications. Due to the low number of anomalies, it is not possible to apply MAML with the standard sampling, which would require K anomalous examples in the inner loop during meta-training. With OC-MAML, the few anomalies available are only used for the outer loop updates. We note that despite the high class-imbalance in the data of the meta-training processes, class-balanced query batches were sampled for the outer loop updates. This can be seen as an under-sampling of the majority class.

F_1	F_2	F_3	F_4	R_1	R_2
80.0%	89.6%	95.9%	93.6%	85.3%	82.6%

Table 2: OC-MAML F1-scores, averaged over 150 tasks sampled from the test operations, on finishing (F_i) and roughing (R_j) operations of the real-world CNC-MMD dataset, with only $K = 10$ normal examples ($c = 0\%$).

Model \ Dataset	MIN	Omn	MNIST	Saw
Reptile	0.05	0.02	0.16	0.02
FOMAML	0.13	0.14	0.31	-0.02
MAML	0.28	0.16	0.45	0.01
OC-Reptile	0.09	0.05	-0.09	0.03
OC-FOMAML	0.26	0.12	0.36	0.07
OC-MAML	**0.42**	**0.23**	**0.47**	**0.92**

Table 3: Cosine similarity between the gradients of one-class and class-balanced minibatches averaged over test tasks of MiniImageNet, Omniglot, MT-MNIST and STS-Sawtooth.

Cosine Similarity Analysis. We would like to directly verify that OC-MAML maximizes the inner product, and therefore the cosine similarity, between the gradients of one-class and class-balanced batches of data, while the other meta-learning baselines do not (see theoretical analysis). For this, we use the initialization meta-learned by each algorithm to compute the loss gradient of K normal examples and the loss gradient of a disjoint class-balanced batch. We use the best performing initialization for each meta-learning algorithm and compute the cosine similarities using on test tasks.

Support set size	$K = 2$			$K = 10$		
Model \ Dataset	MIN	CIFAR-FS	FC100	MIN	CIFAR-FS	FC100
MAML	62.3	62.1	55.1	65.5	69.1	61.6
OC-MAML (ours)	**69.1**	**70**	**59.9**	**76.2**	**79.1**	**65.5**
MetaOptNet	50	56	51.2	56.6	74.8	53.3
OC-MetaOptNet (ours)	**51.8**	**56.3**	**52.2**	**67.4**	**75.5**	**59.9**
MetaSGD	65	58.4	55	73.6	71.3	61.3
OC-MetaSGD (ours)	**69.6**	**71.4**	**60.3**	**75.8**	**77.8**	**64.3**
One-Way ProtoNets (Kruspe 2019)	67	70.9	56.9	74.4	76.7	62.1

Table 4: Test accuracies (in %) computed on the class-balanced test sets of the test tasks of MiniImageNet (MIN), CIFAR-FS and FC100 after using a one-class support set for task-specific adaptation

We report the mean cosine similarity on 3 image datasets and one time-series dataset in Table 3. The significant differences in the mean cosine similarity found between OC-MAML and the other meta-learning algorithms consolidate our theoretical findings.

Applicability to Further Meta-Learning Algorithms and Comparison to One-Way ProtoNets. To investigate whether the benefits of our sampling strategy generalize to further meta-learning algorithms beyond MAML, we apply it to MetaOptNet (Lee et al. 2019) and Meta-SGD (Li et al. 2017). Like MAML, these algorithms use a bi-level optimization scheme (inner and outer loop optimization) to perform few-shot learning. This enables the application of our proposed data strategy which requires two sets of data with different CIRs to be used. We refer to the OC versions of these algorithms as OC-MetaOptNet and OC-MetaSGD.

MetaOptNet trains a representation network to extract feature embeddings that generalize well in the FS regime when fed to linear classifiers, e.g., SVMs. For that, a differentiable quadratic programming (QP) solver (Amos and Kolter 2017) is used to fit the SVM (Lee et al. 2019) (inner loop optimization). The loss of the fitted SVM on a held-out validation set of the same task is used to update the representation network (outer loop optimization). Since solving a binary SVM requires examples from both classes and our sampling strategy provides one-class examples in the inner loop, we use an OC-SVM (Schölkopf et al. 2000) classifier instead. The embeddings extracted for few normal examples by the representation network are used to fit the OC-SVM, which is then used to classify the class-balanced validation set and to update the embedding network, analogously to the class-balanced scenario. To fit the OC-SVM, we solve its dual problem (Schölkopf et al. 2000) using the same differentiable quadratic programming (QP) solver (Amos and Kolter 2017) used to solve the multi-class SVM in (Lee et al. 2019). The ResNet-12 architecture is used for the embedding network. We use the meta-validation tasks to tune the OC-SVM hyperparameters.

Meta-SGD meta-learns an inner loop learning rate for each model parameter besides the initalization. Our episode sampling method is applied as done for MAML. Unlike the class-balanced MetaSGD, the meta-learning optimization assigns negative values to some parameter-specific learning rates to counteract overfitting to the majority class, which leads to performing gradient ascent on the adaptation loss. To prevent this, we clip the learning rates between 0 and 1.

Table 4 shows that applying the proposed sampling technique to MetaOptNet and Meta-SGD results in a significant accuracy increase in FS-OCC on the MiniImageNet, CIFAR-FS and FC100 datasets. Eventhough MetaOptNet substantially outperforms MAML and Meta-SGD in the class-balanced case (Lee et al. 2019), it fails to compete in the FS-OCC setting, suggesting that meta-learning a suitable initialization for the classifier is important in this scenario.

Finally, we compare to One-Way ProtoNets [2] and find that OC-MAML and OC-MetaSGD significantly outperform it on all three datasets. The poorer performance of One-Way ProtoNets and OC-MetaOptNet could be explained by the absence of a mechanism to adapt the feature extractor (the convolutional layers) to the unseen test tasks. OC-MAML and OC-MetaSGD finetune the parameters of the feature extractor by the means of gradient updates on the few normal examples from the test task. We conducted experiments using 5 different seeds and present the average in Table 4.

Conclusion

This work addressed the novel and challenging problem of few-shot one-class classification (FS-OCC). We proposed an episode sampling technique to adapt meta-learning algorithms designed for class-balanced FS classification to FS-OCC. Our experiments on 8 datasets from the image and time-series domains, including a real-world dataset of industrial sensor readings, showed that our approach yields substantial performance increase on three meta-learning algorithms, significantly outperforming classical OCC methods and FS classification algorithms using standard sampling. Moreover, we provided a theoretical analysis showing that class-balanced gradient-based meta-learning algorithms (e.g., MAML) do not yield model initializations suitable for OCC tasks and that second-order derivatives are needed to optimize for such initializations. Future works could investigate an unsupervised approach to FS-OCC, as done in the class-balanced scenario (Hsu, Levine, and Finn 2018).

[2] We re-implemented One-Way ProtoNets to conduct the experiments, since the code from the original paper was not made public.

References

Aggarwal, C. C. 2015. Outlier analysis. In *Data mining*, 237–263. Springer.

Amos, B.; and Kolter, J. Z. 2017. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 136–145. JMLR. org.

An, J.; and Cho, S. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE* 2: 1–18.

Andrews, J. T.; Tanay, T.; Morton, E. J.; and Griffin, L. D. 2016. Transfer representation-learning for anomaly detection. ICML.

Bronskill, J.; Gordon, J.; Requeima, J.; Nowozin, S.; and Turner, R. E. 2020. TaskNorm: Rethinking Batch Normalization for Meta-Learning. *arXiv preprint arXiv:2003.03284* .

Caruana, R. 1997. Multitask learning. *Machine learning* 28(1): 41–75.

Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41(3): 15.

Chen, J.; Sathe, S.; Aggarwal, C.; and Turaga, D. 2017. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, 90–98. SIAM.

Erfani, S. M.; Rajasegarar, S.; Karunasekera, S.; and Leckie, C. 2016. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition* 58: 121–134.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1126–1135. JMLR. org.

Garcia-Teodoro, P.; Diaz-Verdejo, J.; Maciá-Fernández, G.; and Vázquez, E. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security* 28(1-2): 18–28.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Hawkins, S.; He, H.; Williams, G.; and Baxter, R. 2002. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, 170–180. Springer.

Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *science* 313(5786): 504–507.

Hsu, K.; Levine, S.; and Finn, C. 2018. Unsupervised learning via meta-learning. *arXiv preprint arXiv:1810.02334* .

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* .

Khan, S. S.; and Madden, M. G. 2014. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review* 29(3): 345–374.

Kozerawski, J.; and Turk, M. 2018. CLEAR: Cumulative LEARning for One-Shot One-Class Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3446–3455.

Kruspe, A. 2019. One-Way Prototypical Networks. *arXiv preprint arXiv:1906.00820* .

Lake, B.; Salakhutdinov, R.; Gross, J.; and Tenenbaum, J. 2011. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10657–10665.

Lee, Y.; and Choi, S. 2018. Gradient-based meta-learning with learned layerwise metric and subspace. *arXiv preprint arXiv:1801.05558* .

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835* .

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, 413–422. IEEE.

Moya, M. M.; Koch, M. W.; and Hostetler, L. D. 1993. One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N* 93.

Nichol, A.; and Schulman, J. 2018. Reptile: a Scalable Metalearning Algorithm. *arXiv preprint arXiv:1803.02999* .

Oreshkin, B.; López, P. R.; and Lacoste, A. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, 721–731.

Prastawa, M.; Bullitt, E.; Ho, S.; and Gerig, G. 2004. A brain tumor segmentation framework based on outlier detection. *Medical image analysis* 8(3): 275–283.

Ravanbakhsh, M.; Nabi, M.; Sangineto, E.; Marcenaro, L.; Regazzoni, C.; and Sebe, N. 2017. Abnormal event detection in videos using generative adversarial nets. In *2017 IEEE International Conference on Image Processing (ICIP)*, 1577–1581. IEEE.

Ravi, S.; and Larochelle, H. 2017. Optimization as a Model for Few-Shot Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL https://openreview.net/forum?id=rJY0-Kcll.

Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S. A.; Binder, A.; Müller, E.; and Kloft, M. 2018. Deep one-class classification. In *International Conference on Machine Learning*, 4393–4402.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.;

Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3): 211–252. doi:10.1007/s11263-015-0816-y.

Sabokrou, M.; Khalooei, M.; Fathy, M.; and Adeli, E. 2018. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3379–3388.

Schlegl, T.; Seeböck, P.; Waldstein, S. M.; Schmidt-Erfurth, U.; and Langs, G. 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, 146–157. Springer.

Schmidhuber, J. 1987. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. Ph.D. thesis, Technische Universität München.

Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J.; Smola, A. J.; and Williamson, R. C. 2001. Estimating the support of a high-dimensional distribution. *Neural computation* 13(7): 1443–1471.

Schölkopf, B.; Williamson, R. C.; Smola, A. J.; Shawe-Taylor, J.; and Platt, J. C. 2000. Support vector method for novelty detection. In *Advances in neural information processing systems*, 582–588.

Scime, L.; and Beuth, J. 2018. Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. *Additive Manufacturing* 19: 114–126.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 4077–4087.

Spigler, G. 2019. Meta-learnt priors slow down catastrophic forgetting in neural networks. *arXiv preprint arXiv:1909.04170* .

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Suteu, M.; and Guo, Y. 2019. Regularizing Deep Multi-Task Networks using Orthogonal Gradients. *arXiv preprint arXiv:1912.06844* .

Tax, D. M.; and Duin, R. P. 2004. Support vector data description. *Machine learning* 54(1): 45–66.

Triantafillou, E.; Zhu, T.; Dumoulin, V.; Lamblin, P.; Xu, K.; Goroshin, R.; Gelada, C.; Swersky, K.; Manzagol, P.; and Larochelle, H. 2019. Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples. *CoRR* abs/1903.03096. URL http://arxiv.org/abs/1903.03096.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080* .

Xu, D.; Ricci, E.; Yan, Y.; Song, J.; and Sebe, N. 2015. Learning Deep Representations of Appearance and Motion for Anomalous Event Detection. *Procedings of the British Machine Vision Conference 2015* doi:10.5244/c.29.8. URL http://dx.doi.org/10.5244/C.29.8.

A Experimental Details

In the following we provide details about the model architectures used. For MT-MNIST, we use the same 4-block convolutional architecture as used by (Hsu, Levine, and Finn 2018) for their multi-class MNIST experiments. Each convolutional block includes a 3 x 3 convolutional layer with 32 filters, a 2 x 2 pooling and a ReLU non-linearity. The same model architecture is used for the MiniImageNet experiments as done by (Ravi and Larochelle 2016). For the Omniglot experiments, we use the same architecture used in (Finn, Abbeel, and Levine 2017).

On the STS datasets, the model architecture used is composed of 3 modules, each including a 5 x 5 convolutional layer with 32 filters, a 2 x 2 pooling and a ReLU non-linearity. The model architecture used for the CNC-MMD experiments is composed of 4 of these aforementioned modules, except that the convolutional layers in the last two modules include 64 filters. The last layer of all architectures is a linear layer followed by softmax. We note that in the experiments on the time-series datasets (STS and CNC-MMD) 1-D convolutional filters are used.

We conducted a hyperparameter grid search for each meta-learning algorithm separately. The hyperparameters with the most effect on the algorithm performance were identified and variated. These are: the inner and outer learning rates (α and β), the number of inner updates (adaptation steps). We also conducted a separate hyperparameter search for the case, where BN layers are used. Our results are averaged over 5 runs with different seeds, using the best hyperparameter values. For the outer learning rate β we searched over the grid $\{0.1, 0.01, 0.001\}$ for all datasets. Regarding the inner learning rate (α) we searched over the grids $\{0.1, 0.01, 0.001\}$ for MiniImageNet and the STS datasets, and $\{0.1, 0.05, 0.01\}$ for MT-MNIST and Omniglot. As for the number of adaptation steps, we search over the grids $\{1, 3, 5\}$ for MiniImageNet, MT-MNIST and Omniglot, and $\{1, 3, 5, 10\}$ for the STS datasets.

For the meta-learning algorithms, including OC-MAML, we used vanilla SGD in the inner loop and the Adam optimizer (Kingma and Ba 2014) in the outer loop, as done by (Finn, Abbeel, and Levine 2017). For (FO)MAML and OC-(FO)MAML, the size of the query set, also called outer loop minibatch, (Q), was set to 60, 20, 100 and 50, for MiniImageNet, Omnigot, MT-MNIST and the STS datasets respectively. Since the outer loop data is class-balanced, it includes $Q/2$ examples per class. Reptile uses the same batch size for all updates (Nichol and Schulman 2018). Hence, we set the outer loop minibatch size to be equal to the inner loop minibatch size, i.e., $Q = K$. The number of meta-training tasks used in each meta-training iteration (also called meta-batch size) was set to 8 for all datasets.

The MTL and FB baselines were also trained with the Adam optimizer. Here, the batch size used is 32 for all datasets, and the learning rate was set to 0.05 for MiniImageNet and Omniglot and 0.01 for MT-MNIST and STS.

In the following we give the hyperparameters used for the real-world CNC-MMD dataset of industrial sensor readings. The outer learning rate (β) was set to 0.001, the inner learning rate (α) was set to 0.0001, the number of adaptation steps was set to 5 and the meta-batch size was set to 16. Since the sizes and CIRS of the validation sets D^{val} differ across the meta-training tasks in this dataset, we could not fix the outer loop size Q. Here we sample the Q datapoints with the biggest possible size, under the constraint that these datapoints are class-balanced. The resulting Q values are between 4 and 16, depending on the meta-training task.

In the following, we provide details about the meta-training procedure adopted in the meta-learning experiments. We use disjoint sets of data for adaptation (D^{tr}) and validation (D^{val}) on the meta-training tasks, as it was empirically found to yield better final performance (Nichol and Schulman 2018). Hereby, the same sets of data are used in the OC-MAML and baseline experiments. In the MT-MNIST, Omniglot, MiniImageNet and STS experiments, the aforementioned sets of data are class-balanced. The sampling of the batch used for adaptation B ensures that this latter has the appropriate CIR ($c = 50\%$ for MAML, FO-MAML and Reptile, and $c = c_{target}$ for OC-MAML, OC-FOMAML and OC-Reptile). For the one-class meta-learning algorithms, $c_{target} = 0\%$, i.e., no anomalous samples of the target task are available, sothat only normal examples are sampled from D^{tr} during meta-training. In order to ensure that class-balanced and one-class meta-learning algorithms are exposed to the same data during meta-training, we move the anomalous examples from the adaptation set of data (D^{tr}) to the validation set of data (D^{val}). We note that this is only done in the experiments using one-class meta-learning algorithms.

During meta-training, meta-validation episodes are conducted to perform model selection. In order to mimic the adaptation to unseen FS-OCC tasks with CIR $c = c_{target}$ at test time, the CIR of the batches used for adaptation during meta-validation episodes is also set to $c = c_{target}$. We note that the hyperparameter K denotes the total number of datapoints, i.e., batch size, used to perform the adaptation updates, and not the number of datapoints *per class* as done by (Finn, Abbeel, and Levine 2017). Hence, a task with size $K = 10$ and CIR $c = 50\%$ is equivalent to a 2-way 5-shot classification task.

In the following, we provide details about the adaptation to the target task(s) and the subsequent evaluation. In the MT-MNIST and MiniImageNet experiments, we randomly sample 20 adaptation sets from the target task(s)' data, each including K examples with the CIR corresponding to the experiment considered. After each adaptation episode conducted using one of these sets, the adapted model is evaluated on a disjoint class-balanced test set that includes 4,000 images for MT-MNIST and 600 for MiniImageNet. We note that the samples included in the test sets of the test tasks are not used nor for meta-training neither for meta-validation. This results in 20 and 400 (20 adaptation sets created from each of the 20 test classes) different test tasks for MT-MNIST and MiniImageNet, respectively. All the results presented give the mean over all adaptation episodes. Likewise, in the STS experiments, we evaluate the model on 10 different adaptation sets from each of the 5 test tasks. In the CNC-MMD experiments, the 30 tasks created from the tar-

get operation are used for adaptation and subsequent evaluation. For each of these target tasks, we randomly sample K datapoints belonging to the normal class that we use for adaptation, and use the rest of the datapoints for testing. We do this 5 times for each target task, which results in 150 testing tasks. For MTL and FB baselines, as well as all the baseline combining these model with shallow models, i.e., IF and OC-SVM, we use the meta-validation task(s) for model choice, like in the meta-learning experiments. For the MTL baseline, for each validation task, we finetune a fully connected layer on top of the shared multi-task learned layers, as it is done at test time.

B Datasets and task creation procedures

In this Section we first provide general information about the datasets used in our experiments. Subsequently, we present more detailed information about the original datasets, the procedures adopted for creating OCC tasks, and the steps adopted to create the proposed STS datasets.

We evaluate our approach on 8 datasets from the image and time-series data domains. From the image domain we use 4 few-shot learning benchmarks, namely MiniImageNet (Ravi and Larochelle 2016), Omniglot (Lake, Salakhutdinov, and Tenenbaum 2015), CIFAR-FS (Bertinetto et al. 2018) and FC100 (Oreshkin, López, and Lacoste 2018) and 1 OCC benchmark dataset, the Multi-Task MNIST (MT-MNIST) dataset. To adapt the datasets to the OCC scenario, we create binary classification tasks, where the normal class contains examples from one class of the initial dataset and the anomalous class contains examples from *multiple* other classes. We note that the class-balanced versions of the meta-learning baselines, e.g., MAML and Reptile, are trained with class-balanced data batches from such AD tasks in the inner loop of meta-training. We create 9 sub-datasets based on MNIST, where the meta-testing task of each consists in differentiating between a certain digit and the others, and the same (10^{th}) task for meta-validation in all sub-datasets.

Since most of the time-series datasets for anomaly detection include data from only one domain and only one normal class, it is not possible them to the meta-learning problem formulation where several different tasks are required. Therefore, we create two synthetic time-series (STS) datasets, each including 30 synthetically generated time-series that underlie 30 different anomaly detection tasks. The time-series underlying the datasets are sawtooth waveforms (STS-Sawtooth) and sine functions (STS-Sine). We propose the STS-datasets as benchmark datasets for the few-shot (one-class) classification problem in the time-series domain and will publish them upon paper acceptance. Finally, we validate OC-MAML on a real-world anomaly detection dataset of sensor readings recorded during industrial manufacturing using a CNC milling machine. Various consecutive roughing and finishing operations (pockets, edges, holes, surface finish) were performed on ca. 100 aluminium workpieces to record the CNC Milling Machine Data (CNC-MMD). The temporal dimension is handled using 1-D convolutions.

In the following, we give details about all datasets, the task creation procedures adopted to adapt them to the OCC case, as well as the generation of the STS-datasets.

Multi-task MNIST (MT-MNIST): We derive 10 binary classification tasks from the MNIST dataset (LeCun, Cortes, and Burges 2010), where every task consists in recognizing one of the digits. This is a classical one-class classification benchmark dataset. For a particular task T_i, images of the digit i are labeled as normal samples, while out-of-distribution samples, i.e., the other digits, are labeled as anomalous samples. We use 8 tasks for meta-training, 1 for meta-validation and 1 for meta-testing. Hereby, images of digits to be recognized in the validation and test tasks are not used as anomalies in the meta-training tasks. This ensures that the model is not exposed to normal samples from the test task during meta-training. Moreover, the sets of anomalous samples of the meta-training, meta-validation and meta-testing tasks are mutually disjoint. We conduct experiments on 9 MT-MNIST datasets, each of which involves a different target task ($T_0 - T_8$). The task T_9 is used as a meta-validation task across all experiments. Each image has the shape 28x28.

Omniglot: This dataset was proposed in (Lake, Salakhutdinov, and Tenenbaum 2015) and includes 20 instances of 1623 hand-written characters from 50 different alphabets. We generate our meta-training and meta-testing tasks based on the official data split (Lake, Salakhutdinov, and Tenenbaum 2015), where 30 alphabets are reserved for training and 20 for evaluation. For each character class, we create a binary classification task, which consists in differentiating between this character and other characters from the same set (meta-training or meta-testing), i.e., the anomalous examples of a task T_i are randomly sampled from the remaining characters. By removing 80 randomly sampled tasks from the meta-training tasks, we create the meta-validation tasks set. Each image has the shape 28x28.

MiniImageNet: This dataset was proposed in (Ravi and Larochelle 2016) and includes 64 classes for training, 16 for validation and 20 for testing, and is a classical challenging benchmark dataset for few-shot learning. 600 images per class are available. To adapt it to the few-shot *one-class* classification setting, we create 64 binary classification tasks for meta-training, each of which consists in differentiating one of the training classes from the others, i.e., the anomalous examples of a task T_i are randomly sampled from the 63 classes with labels different from i. We do the same to create 16 meta-validation and 20 meta-testing tasks using the corresponding classes. Each image has the shape 84x84x3.

CIFAR-FS: This dataset was proposed in (Bertinetto et al. 2018) and includes 64 classes for training, 16 for validation and 20 for testing, derived from CIFAR-100, and is a benchmark dataset for few-shot learning. 600 images of size 32x32x3 are available per class. To adapt it to the few-shot *one-class* classification setting, we proceeded exactly as we did for miniImageNet (see above).

FC100: This dataset was proposed in (Oreshkin, López, and Lacoste 2018) and also includes 64 classes for training, 16 for validation and 20 for testing derived from CIFAR-100, and is a benchmark dataset for few-shot learning. How-

ever, in this dataset, the classes for training, validation and testing belong to different superclasses to minimize semantic overlap. This dataset contains 600 images of size 32x32x3 per class. To adapt it to the few-shot *one-class* classification setting, we proceeded exactly as we did for mini-ImageNet (see above).

Synthetic time-series (STS): In order to investigate the applicability of OC-MAML to time-series (question (c)), we created two datasets, each including 30 synthetically generated time-series that underlie 30 different anomaly detection tasks. The time-series underlying the datasets are sawtooth waveforms (STS-Sawtooth) and sine functions (STS-Sine). Each time-series is generated with random frequencies, amplitudes, noise boundaries, as well as anomaly width and height boundaries. Additionally, the width of the rising ramp as a proportion of the total cycle is sampled randomly for the sawtooth dataset, which results in tasks having rising and falling ramps with different steepness values. The data samples of a particular task are generated by randomly cropping windows of length 128 from the corresponding time-series. We generate 200 normal and 200 anomalous data examples for each task. For each dataset, we randomly choose 20 tasks for meta-training, 5 for meta-validation and 5 for meta-testing. We propose the STS-datasets as benchmark datasets for the few-shot one-class classification problem in the time-series domain, and will make them public upon paper acceptance.

In the following, we give details about the generation procedure adopted to create the STS-Sawtooth dataset. The same steps were conducted to generate the STS-Sine dataset. First, we generate the sawtooth waveforms underlying the different tasks by using the Signal package of the Scipy library (Jones et al. 2001–). Thereafter, a randomly generated noise is applied to each signal. Subsequently, signal segments with window length $l = 128$ are randomly sampled from each noisy signal. These represent the normal, i.e., non-anomalous, examples of the corresponding task. Then, some of the normal examples are randomly chosen, and anomalies are added to them to produce the anomalous examples.

Figure 1 shows exemplary normal and anomalous samples from the STS-Sawtooth and STS-Sine datasets. In order to increase the variance between the aforementioned synthetic signals underlying the different tasks, we randomly sample the frequency, i.e., the number of periods within the window length l, with which each waveform is generated, as well as the amplitude and the vertical position (see Figure 1). For sawtooth waveforms, we also randomly sample the width of the rising ramp as a proportion of the total cycle between 0% and 100%, for each task. Setting this value to 100% and to 0% produces sawtooth waveforms with rising and falling ramps, respectively. Setting it to 50% corresponds to triangle waveforms.

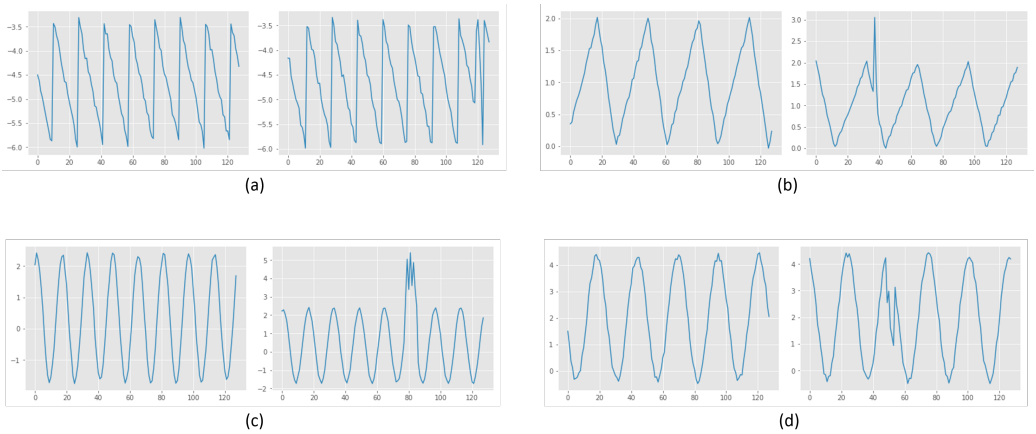
We note that the noise applied to the tasks are randomly sampled from *task-specific* intervals, the boundaries of which are also randomly sampled. Likewise, the width and height of each anomaly is sampled from a random task specific-interval. Moreover, we generate the anomalies of each task, such that half of them have a height between the signal's minimum and maximum (e.g., anomalies (a) and (d) in Figure 1), while the other half can surpass these boundaries, i.e., the anomaly is higher than the normal signal's maximum or lower than its minimum at least at one time step (e.g., anomalies (b) and (c) in Figure 1). We note that an anomalous sample can have more than one anomaly.

We preprocess the data by removing the mean and scaling to unit variance. Hereby, only the available *normal* examples are used for the computation of the mean and the variance. This means that in the experiments, where the target task's size $K = 2$ and only normal samples are available $c = 0\%$, only two examples are used for the mean and variance computation. We note that the time-series in Figure 1 are not preprocessed.

CNC Milling Machine Data (CNC-MMD): This dataset consists of ca. 100 aluminum workpieces on which various consecutive roughing and finishing operations (pockets, edges, holes, surface finish) are performed. The sensor readings which were recorded at a rate of 500Hz measure various quantities that are important for the process monitoring including the torques of the various axes. Each run of machining a single workpiece can be seen as a multivariate time-series. We segmented the data of each run in the various operations performed on the workpieces. e.g., one segment would describe the milling of a pocket where another describes a surface finish operation on the workpiece. Since most manufacturing processes are highly efficient, anomalies are quite rare but can be very costly if undetected. For this reason, anomalies were provoked for 6 operations during manufacturing to provide a better basis for the analysis. Anomalies were provoked by creating realistic scenarios for deficient manufacturing. Examples are using a workpiece that exhibits deficiencies which leads to a drop in the torque signal or using rather slightly decalibrated process parameters which induced various irritations to the workpiece surface which harmed production quality. The data was labeled by domain experts from Siemens Digital Industries. It should be noted that this dataset more realistically reflects the data situation in many real application scenarios from industry where anomalies are rare and data is scarce and for this reason training models on huge class-balanced datasets is not an option.

For our experiments, we created 30 tasks per operation by randomly cropping windows of length 2048 from the corresponding time-series of each operation. As a result, the data samples of a particular task T_i cropped from a milling operation O_j correspond to the same trajectory part of O_j, but to different workpieces. The task creation procedure ensures that at least two anomalous data samples are available for each task. The resulting tasks include between 15 and 55 normal samples, and between 2 and 4 (9 and 22) anomalous samples for finishing (roughing) operations. We validate our approach on all 6 milling operations in the case where only 10 samples belonging to the normal class ($K = 10, c = 0\%$) are available. Given the type of the target milling operation,e.g., finishing, we use the tasks from the other operations of the same type for meta-training. We note that the model is not exposed to any sample belonging to any task of the target operation during training. Each example has the

Figure 1: Exemplary normal (left) and anomalous (right) samples belonging to different tasks from the STS-Sawtooth (a and b) and the STS-Sine (c and d) datasets



shape 2048x3.

We preprocess each of the three signals separately by removing the mean and scaling to unit variance, as done for the STS datasets. Likewise, only the available *normal* examples are used for the computation of the mean and the variance.

Exemplary anomalous signals recorded from a finishing and a roughing operations are shown in Figure 2. These signals are not mean centered and scaled to unit variance. We note that we do not use the labels per time-step, but rather the label ”anomalous” is assigned to each time-series that contains at least an anomalous time-step.

C Further Experimental Results

In this section, we first present a more detailed overview of the experimental results on meta-learning algorithms, with and without using Batch Normalization (BN) layers in Table 1. Subsequently, we report the results of the experiments on the STS-Sine dataset (Tables 2 and 3) and the 8 further MT-MNIST task combinations (Tables 4, 5, 6, 7).

On the STS-Sine dataset and the 8 other MT-MNIST task combinations, we observe consistent results with the results from section **??**. OC-MAML yields high performance across all datasets. We also note that none of the meta-learning baselines consistently yields high performance across all datasets, as it is the case for OC-MAML.

D Speeding up OC-MAML

A concurrent work (Raghu et al. 2019) established that MAML's rapid learning of new tasks is dominated by feature reuse. The authors propose the Almost No Inner Loop (ANIL) algorithm, which consists in limiting the task-specific adaptation of MAML (inner loop updates) to the parameters of the model's last layer (the output layer), during meta-training and meta-testing. This leads to a speed up factor of 1.7 over MAML, since ANIL requires the computation of second-order derivative terms only for the last layer's parameters instead of all parameters. ANIL achieves very comparable performance to MAML.

We investigate, whether this simplification of MAML can also speed up OC-MAML, while retaining the same performance. In other words, could we also compute the second-order derivatives, which are required to explicitly optimize for few-shot one-class classification (FS-OCC) (section **??**), to the last layer's parameters and still reach a model initialization suitable for FS-OCC. Preliminary results of OC-ANIL on the MiniImageNet and Omniglot datasets were very comparable to the results of OC-MAML. Moreover, we conducted the same cosine similarity analysis described in section **??** with ANIL, FOANIL, OC-ANIL and OC-FOANIL and got very consistent results with our findings for the MAML-based algorithms (Table **??**). This confirms that second-order derivatives have only to be computed for the last layer of the neural network to optimize for FS-OCC, and that OC-ANIL is faster than OC-MAML by a factor of 1.7 (Raghu et al. 2019) with comparable performance. This modification significantly reduces the computational burden incurred by computing the second-order derivatives for all parameters as done in OC-MAML. Our implementation of OC-ANIL will be published upon paper acceptance.

Figure 2: Exemplary anomalous samples from a finishing (left) and a roughing (right) operations, where the anomalous time-steps are depicted in red.

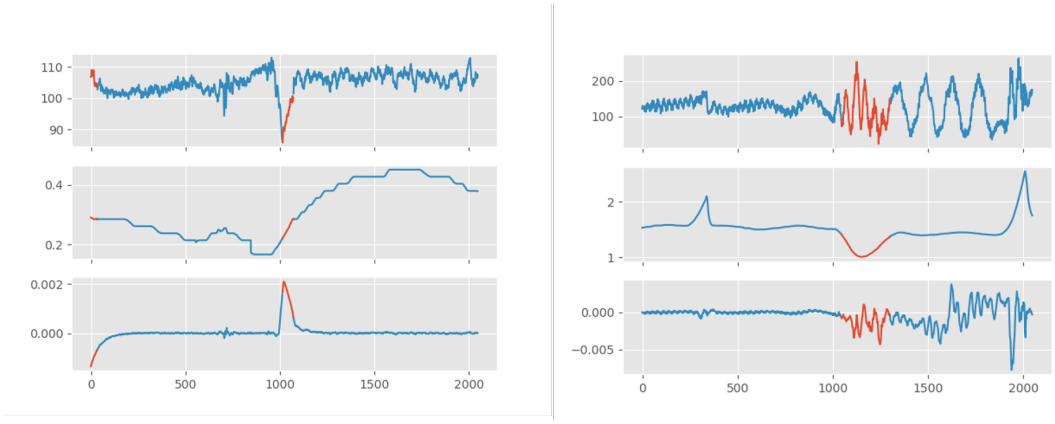


Table 1: Test accuracies (in %) computed on the class-balanced test sets of the test tasks of MiniImageNet (MIN), Omniglot (Omn), MT-MNIST with $T_{test} = T_0$ and STS-Sawtooth (Saw). The results are shown for models without BN (top) and with BN (bottom), and give the average over 5 different seeds. One-class support sets ($c = 0\%$) are used, unless otherwise specified.

Support set size	$K = 2$				$K = 10$			
Model \ Dataset	MIN	Omn	MNIST	Saw	MIN	Omn	MNIST	Saw
Reptile	50.2	50	71.1	50.6	50.2	56.2	85.2	72.8
FOMAML	50	50.9	80.7	52.5	50	50.6	83.8	50.4
MAML	51.4	87.2	80.7	81.1	50	92.3	91.9	72.9
OC-Reptile	50	50.4	50	50	50	50.7	50	50
OC-FOMAML	54.6	52.2	57.5	55.9	51	53.2	73.3	58.9
OC-MAML (ours)	66.4	95.6	85.2	**96.6**	73	96.8	**95.1**	**95.7**
Reptile (BN)	51.6	56.3	61.8	69.1	57.1	76.3	89.8	81.6
FOMAML (BN)	53.3	78.8	80	75.1	59.5	93.7	91.1	80.2
MAML (BN)	62.3	91.4	85.5	51.7	65.5	96.3	92.2	86
OC-Reptile (BN)	51.9	52.1	51.3	51.6	53.2	51	51.4	53.2
OC-FOMAML (BN)	55.7	74.7	79.1	58.6	66.1	87.5	91.8	73.2
OC-MAML (BN) (ours)	**69.1**	**96.6**	**88**	51.3	**76.2**	**97.6**	**95.1**	88.8

Table 2: Test accuracies (in %) computed on the class-balanced test sets of the test tasks of the STS-Sine dataset. One-class adaptation sets ($c = 0\%$) are used, unless otherwise specified.

Model \ Adaptation set size	$K = 2$	$K = 10$
FB ($c = 50\%$)	68.9	77.7
MTL ($c = 50\%$)	64.5	91.2
FB	73.8	76.6
MTL	50	50
OC-SVM	50.2	51.3
IF	50	49.9
FB + OCSVM	52.1	65.3
FB + IF	50	62.8
MTL + OCSVM	50	51.9
MTL + IF	50	64.7
OC-MAML (ours)	**99.9**	**99.9**

Table 3: Test accuracies (in %) computed on the class-balanced test sets of the test tasks of the STS-Sine dataset. The results are shown for models without BN (top) and with BN (bottom). One-class adaptation sets ($c = 0\%$) are used, unless otherwise specified.

Model \ Adaptation set size	$K = 2$	$K = 10$
Reptile	52.5	50.0
FOMAML	60.3	52.1
MAML	99.6	99.1
OC-Reptile	50.0	50.0
OC-FOMAML	78.7	58.1
OC-MAML (ours)	**99.9**	**99.9**
Reptile (w. BN)	90.9	98.6
FOMAML (w. BN)	90.8	97.3
MAML (w. BN)	51.4	99.0
OC-Reptile (w. BN)	52.6	53.4
OC-FOMAML (w. BN)	78.8	80.0
OC-MAML (w. BN) (ours)	50.5	95.5

Table 4: Test accuracies (in %) computed on the class-balanced test sets of the test tasks of the MT-MNIST datasets with $T_{test} = T_{1-4}$. One-class adaptation sets ($c = 0\%$) are used, unless otherwise specified.

Adaptation set size	$K = 2$				$K = 10$			
Model \ Dataset	1	2	3	4	1	2	3	4
FB ($c = 50\%$)	78.8	59.8	66.7	66.8	91.9	77.3	79.9	81.5
MTL ($c = 50\%$)	64.9	65	59.5	56.4	91	84.6	84.4	83.3
FB	53.7	56	50.7	57.1	53.6	50.7	50.2	59
MTL	54	46.8	41.5	52	49.4	49.6	54.7	46.1
OC-SVM	56.9	51.5	50.5	51.8	63.7	50.2	51.2	51.5
IF	50	50	50	50	50.9	50	50.1	50
FB + OCSVM	50.1	53.2	51.8	56.1	62.5	70.5	80.4	89.8
FB + IF	50	50	50	50	54.3	51.3	77.7	67.4
MTL + OCSVM	50	50	50	50	50.2	52.8	54.8	50.7
MTL + IF	50	50	50	50	76.5	75.5	69.3	74.4
OC-MAML (ours)	**87.1**	**86.3**	**86.8**	**85.9**	**92.5**	**92.4**	**91.7**	**92**

Table 5: Test accuracies (in %) computed on the class-balanced test sets of the test tasks of the MT-MNIST datasets with $T_{test} = T_{5-8}$. One-class adaptation sets ($c = 0\%$) are used, unless otherwise specified.

Adaptation set size	$K = 2$				$K = 10$			
Model \ Dataset	1	2	3	4	1	2	3	4
FB ($c = 50\%$)	64.6	69.8	68.9	62.9	64.4	83	87.8	72.8
MTL ($c = 50\%$)	60.5	71.4	65	60.6	88.4	91.4	82	79.1
FB	52.2	66.5	54.3	53.8	58.3	63.5	53.6	50.1
MTL	48.5	56.2	51.1	50.1	49.9	51.4	48.5	49.6
OC-SVM	51	53.4	53.9	50.1	50.5	54	54	52.2
IF	50	50	50	50	50	50.2	49.8	50.2
FB + OCSVM	52.2	51.2	50.5	58	86.2	75	84.5	80
FB + IF	50	50	50	50	80.4	87.2	79.2	71.4
MTL + OCSVM	50	50	50	50	51	59.1	71.3	75.9
MTL + IF	50	50	50	50	50	55.7	84.2	64
OC-MAML (ours)	**85.9**	**91.5**	**85.1**	**82.5**	**91.5**	**95.4**	**91.4**	**89.8**

Table 6: Test accuracies (in %) computed on the class-balanced test sets of the test tasks of the MT-MNIST datasets with $T_{test} = T_{1-4}$. The results are shown for models without BN (top) and with BN (bottom). One-class adaptation sets ($c = 0\%$) are used, unless otherwise specified.

Adaptation set size	$K = 2$				$K = 10$			
Model \ Dataset	1	2	3	4	1	2	3	4
Reptile	67.1	58.3	57	65.9	82.4	78.5	76.4	81.8
FOMAML	76.3	74.2	74.9	75.6	82.3	75.7	75.1	80.8
MAML	78.1	71.8	77	71.4	88.8	88.7	87.2	86.6
OC-Reptile	50	50	50	50	50	50	50	50
OC-FOMAML	56.6	52.6	55.6	50.1	50.7	50	53.8	64.3
OC-MAML (ours)	85.2	83.5	80.2	84.3	**92.5**	**92.4**	**91.7**	**92**
Reptile (w. BN)	58.9	56	56.6	62.2	90.4	84.6	88.9	86.7
FOMAML (w. BN)	75.4	72.3	72.2	74.5	91.7	88.6	86	87.8
MAML (w. BN)	83.5	81.3	83.9	77	91.9	90.3	88.7	87.3
OC-Reptile (w. BN)	52	53.2	51.9	51	51.5	51.2	51.1	50.3
OC-FOMAML (w. BN)	74.7	68.5	67	78.7	90.2	85.5	84.3	89.3
OC-MAML (w. BN) (ours)	**87.1**	**86.3**	**86.8**	**85.9**	92.1	90.7	90.2	91.8

Table 7: Test accuracies (in $\%$) computed on the class-balanced test sets of the test tasks of the MT-MNIST datasets with $T_{test} = T_{5-8}$. The results are shown for models without BN (top) and with BN (bottom). One-class adaptation sets ($c = 0\%$) are used, unless otherwise specified.

Adaptation set size	$K = 2$				$K = 10$			
Model \ Dataset	1	2	3	4	1	2	3	4
Reptile	60.3	65.4	59.9	57.2	73.6	86.3	79.1	72.3
FOMAML	76.5	80.2	77.9	72.8	66.2	84.4	76.9	72.5
MAML	74.8	82.1	73.6	70.7	86.1	93	90.2	85.7
OC-Reptile	50	50	50	50	50	50	50	50
OC-FOMAML	54.6	57.3	59	55	53.3	56.7	51	50
OC-MAML (ours)	80.6	**91.5**	82.1	77.5	**91.5**	94.2	91.3	**89.8**
Reptile (w. BN)	62.3	58.2	60.3	61	85.3	88	88.4	87.2
FOMAML (w. BN)	69.5	75.1	77.3	72.8	86.9	92.3	88.6	85
MAML (w. BN)	84.9	81.8	83.4	76.9	88.6	92.5	90.5	84
OC-Reptile (w. BN)	52	52.2	51.7	53.5	51.1	51.4	50.9	51.8
OC-FOMAML (w. BN)	68.3	85.7	78.5	67.1	83.2	94.7	89.9	82.6
OC-MAML (w. BN) (ours)	**85.9**	84.8	**85.1**	**82.5**	90.5	**95.4**	**91.4**	**89.8**

References

Bertinetto, L.; Henriques, J. F.; Torr, P. H.; and Vedaldi, A. 2018. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136* .

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1126–1135. JMLR. org.

Hsu, K.; Levine, S.; and Finn, C. 2018. Unsupervised Learning via Meta-Learning.

Jones, E.; Oliphant, T.; Peterson, P.; et al. 2001–. SciPy: Open source scientific tools for Python. URL http://www.scipy.org/. [Online; accessed ¡today¿].

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science* 350(6266): 1332–1338. ISSN 0036-8075. doi:10.1126/science.aab3050. URL https://science.sciencemag.org/content/350/6266/1332.

LeCun, Y.; Cortes, C.; and Burges, C. J. 2010. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

Nichol, A.; and Schulman, J. 2018. Reptile: a Scalable Metalearning Algorithm. *arXiv preprint arXiv:1803.02999* .

Oreshkin, B.; López, P. R.; and Lacoste, A. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, 721–731.

Raghu, A.; Raghu, M.; Bengio, S.; and Vinyals, O. 2019. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157* .

Ravi, S.; and Larochelle, H. 2016. Optimization as a model for few-shot learning .

Chapter 3

ARCADe: A Rapid Continual Anomaly Detector

ARCADe: A Rapid Continual Anomaly Detector

Ahmed Frikha
University of Munich
Siemens AG, Corporate Technology
Munich, Germany
ahmed.frikha@siemens.com

Denis Krompaß
Siemens AG, Corporate Technology
Munich, Germany
denis.krompass@siemens.com

Volker Tresp
University of Munich
Siemens AG, Corporate Technology
Munich, Germany
volker.tresp@siemens.com

***Abstract*—Although continual learning and anomaly detection have separately been well-studied in previous works, their intersection remains rather unexplored. The present work addresses a learning scenario where a model has to incrementally learn a sequence of anomaly detection tasks, i.e. tasks from which only examples from the normal (majority) class are available for training. We define this novel learning problem of continual anomaly detection (CAD) and formulate it as a meta-learning problem. Moreover, we propose *A Rapid Continual Anomaly Detector (ARCADe)*, an approach to train neural networks to be robust against the major challenges of this new learning problem, namely catastrophic forgetting and overfitting to the majority class. The results of our experiments on three datasets show that, in the CAD problem setting, ARCADe substantially outperforms baselines from the continual learning and anomaly detection literature. Finally, we provide deeper insights into the learning strategy yielded by the proposed meta-learning algorithm.**

I. Introduction

Humans can continually learn new tasks without corrupting their previously acquired abilities. Neural networks, however, tend to overwrite older knowledge and therefore fail at incrementally learning new tasks. This is called the catastrophic forgetting problem [1], [2], [3]. In fact, most deep learning achievements have been realized in the offline supervised single-task or multi-task learning [4] settings, where the availability of independent and identically distributed (i.i.d.) data can be assumed. Building intelligent agents that are able to incrementally acquire new capabilities while preserving the previously learned ones remains an old and long-standing goal in machine learning research.

Several approaches have been developed to enable continual learning, e.g. by alleviating interference between the sequentially learned tasks, [5], [3], [6] and/or encouraging knowledge transfer between them [7], [8], [9], [10], [11]. While most of the previous works addressed the continual learning problem with neatly class-balanced classification tasks, many real-world applications exhibit extreme class-imbalance, e.g. in anomaly detection [12] problems. For example, in industrial manufacturing, of all produced parts, only a few per million are faulty. And since the products and/or machines in the plant are continuously changing, building a central anomaly detector that incrementally improves by learning new anomaly detection tasks would relax this cold-start problem.

To the best of our knowledge, continual learning with class-imbalanced data has only been addressed in [13], [14]. Hereby the authors assume, however, access to examples from all classes, including the minority class. In the anomaly detection literature [12] most works address the unsupervised anomaly detection problem, where only examples from the majority (normal) class are available for training an anomaly detector. Learning a binary classifier using data samples from only one of its classes (usually the majority class) is referred to as One-Class Classification (OCC) [15], [16]. Our work addresses the novel and unexplored problem of *Continual Anomaly Detection (CAD)*, where different binary classification tasks have to be learned sequentially by using only examples from their respective majority classes for training. We also refer to this problem as *Continual One-Class Classification (ContOCC)*. In particular, we propose an approach that relies on meta-learning [17] to yield a parameter initialization that resists to the main challenges of CAD, namely catastrophic forgetting and overfitting to the majority class. Several state-of-the-art works introduced meta-learning algorithms to tackle continual learning problems [9], [10], [18], [19], [11]. Hereby, however, only class-balanced classification tasks were considered.

Our contribution in this work is threefold: Firstly, we introduce and define the novel and relevant CAD problem. Secondly, we propose a first, strong and model-agnostic approach to tackle it. Thirdly, we successfully validate our approach on three datasets, where we substantially outperform continual learning and anomaly detection baseline methods.

II. The Continual Anomaly Detection (CAD) Problem

The goal of *Continual Anomaly Detection (CAD)* or *Continual One-Class Classification (ContOCC)* is to sequentially learn multiple OCC tasks without forgetting previously learned one. More precisely, the target model should be able to sequentially learn binary classification tasks by using only examples from their respective normal classes for training, and then achieve a high performance in distinguishing between both classes of each of the learned tasks, when faced with unseen datapoints. The CAD problem is a prototype for a practical use case where a central anomaly detector for multiple applications is needed and new applications become available gradually in

time. In this section, we first discuss the unique challenges of the CAD learning scenario. Subsequently, we present a problem formulation for CAD. Finally, we introduce the meta-learning optimization technique, upon which our approach builds to tackle CAD.

A. Unique Challenges

In order to perform CAD, approximating *one* decision boundary that encompasses *all* the (normal) majority classes of the observed tasks is necessary. In fact, the examples belonging to the normal class of *any* observed task should be mapped inside the normal class boundary, and therefore classified as normal. Learning such a decision boundary can be especially challenging due to two inherent problems of neural networks: catastrophic forgetting and overfitting to the majority class, i.e. predicting the normal class label for any input. On the one hand, each model update that we perform using examples from a new task shifts our decision boundary away from the normal class of previously learned tasks, resulting in a poorer classification performance on the latter (catastrophic forgetting). On the other hand, since the model is only exposed to (normal) majority class examples, the decision boundary tends to over-generalize and classify any input as normal. This way the model overfits to the normal class and anomalies would not be detected.

B. Problem Formulation

We define a CAD task-sequence $S = \{T_1, ..., T_n\}$ as an ordered sequence of OCC tasks T_i. To learn S, the classification model is trained on the tasks included in it, one after another. Due to the sequential exposure to tasks, the model is trained with non i.i.d. samples. This setting is commonly used in class-balanced continual learning to define non-stationary conditions. It is also called *locally i.i.d.* [7], [9], since the model is exposed to a sequence of stationary distributions, defined by the tasks T_i. In contrast, offline single-task and multi-task learning assume that a fixed training dataset is available at all points in time. We note that for an OCC task T_i, the training set T_i^{tr} and the validation set T_i^{val} have different data distributions, since T_i^{tr} includes only examples from one class and T_i^{val} is class-balanced.

In the following we formulate the CAD problem as a meta-learning problem. We consider separate sets of task-sequences for meta-training (D_{tr}), meta-validation (D_{val}) and meta-testing (D_{test}). Hereby all the tasks in these sequences belong to the same domain, i.e. come from a task distribution $p(T)$. To prevent leakage between D_{tr}, D_{val} and D_{test}, these sets of data must have mutually exclusive classes, i.e. none of the classes building the tasks T included in D_{tr} is used to build a task in D_{val} or D_{test} and vice versa.

Each sequence in D_{tr}, D_{val} and D_{test} is composed of a training and a test set. Let $S_{tr} = \{S_{tr}^{tr}, S_{tr}^{val}\}$ denote a meta-training task-sequence from D_{tr}, where $S_{tr}^{tr} = \{T_1^{tr}, ..., T_n^{tr}\}$ is a sequence of the training sets of the tasks composing S, and $S_{tr}^{val} = \{T_1^{val}, ..., T_n^{val}\}$ is a sequence of their validation sets. Following the terminology introduced in [11] to formulate meta-learning problems, we refer to S_{tr}^{tr} as a *meta-training training* task-sequence and S_{tr}^{val} as a *meta-training validation* task-sequence. We call the set of all meta-training training (validation) sequences the meta-training training (validation) set D_{tr}^{tr} (D_{tr}^{val}). We note that the sequences in D_{tr}^{tr} include examples from only the majority class of each task, while the sequences in D_{tr}^{val} contain disjoint class-balanced sets of data from each task. The same holds for the meta-validation and meta-testing sets D_{val} and D_{test}.

We aim to find an algorithm that, by using D_{tr}, yields a learning strategy that enables a classification model to sequentially learn anomaly detection tasks without (or with minimal) forgetting. Applying this learning strategy to a random task-sequence from D_{test}^{tr} would then provide a model that has high performance on D_{test}^{val}, hence performing CAD. In this work the learning strategy yielded by the proposed meta-learning algorithm consists in a model initialization and a learning rate for each model parameter, which are suitable to perform CAD. Starting from the meta-learned model initialization, taking few gradient descent steps with the meta-learned learning rates to learn each of the OCC tasks in a sequence S leads to a proficient anomaly detector on all tasks.

C. Continual Learning via Meta-Learning a Parameter Initialization

The proposed meta-learning approach to tackle the CAD problem learns a model initialization and parameter-specific learning rates by building upon a bi-level optimization scheme. In this section we explain this optimization mechanism which was introduced in the MAML algorithm [20] to address the few-shot learning problem [21], [22]. Since then this optimization scheme was used by multiple meta-learning algorithms to address several problems, e.g. few-shot learning [20], [23], [24], [25], few-shot one-class classification [26], resisting to adversarial examples [27] and continual learning [10], [18], [11].

Let θ denote the set of model parameters. The aforementioned bi-level optimization mechanism aims to optimize these model parameters to be easily adaptable to unseen tasks T_i which have certain characteristics, e.g. few-shot learning tasks, anomaly detection tasks or continual learning tasks. After adaptation to a task T_i, e.g. by taking few gradient steps using its training set, the adapted parameters $\theta_i^{'}$ yield high performance on a held-out test set of the same task. In that sense the meta-learned model parameters θ can be viewed as a parameter initialization that enables quick learning of unseen tasks. The meta-learned parameter initialization represents an inductive bias that facilitates learning tasks with certain characteristics.

To find such a model initialization, a model is explicitly trained for quick adaptation using a set of meta-training tasks.

Hereby, these tasks belong to the same domain and have the same characteristics as the test tasks, e.g. if the unseen test tasks are expected to have only few examples, the meta-training tasks should be few-shot learning tasks [20]. In each meta-training iteration, two operations are performed for each task, parameter adaptation and evaluation. Adapting the model initialization θ to a task T_i is done by taking few gradient descent steps using its training set T_i^{tr}, yielding a task-specific model $\theta_i^{'}$. The evaluation of the task-specific model uses the task's validation set T_i^{val}. The resulting loss $L_{T_i}^{val}(f_{\theta_i^{'}})$ is used to update the initialization θ as shown in Equation 1, where β is the learning rate used for this update.

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i \sim p(T)} L_{T_i}^{val}(f_{\theta_i^{'}}). \tag{1}$$

For a model initialization to be suitable for continual learning, i.e. to inhibit catastrophic forgetting, each meta-training and meta-testing task is built as a sequence of classification tasks [18], [10]. Hereby, adapting the parameter initialization to a task-sequence consists in taking a few gradient descent steps on the tasks included in it, sequentially. The parameter initialization is then updated as shown in Equation 1, where $L_{T_i}^{val}(f_{\theta_i^{'}})$ is the sum of the losses computed on the validation set of each task in the task-sequence.

III. Related Work

The present work addresses the Continual Anomaly Detection (CAD) problem, which represents the intersection of the continual learning and anomaly detection problems. To the best of our knowledge no prior works addressed the CAD problem. Therefore, in this section we review related continual learning and anomaly detection work separately.

A. Continual Learning

Several Continual learning (CL) approaches inhibit catastrophic forgetting by retaining past knowledge. This can be done by increasing the model capacity [28] or by regularizing the parameter updates [3], [5], [6]. Another category of CL methods relies on replaying previous experiences, e.g. datapoints, by interleaving them between new experiences [29], [7], [9]. Recent works developed meta-learning based approaches to tackle CL [9], [19], [30], [10], [18], [11]. In [9], a method that maximizes transfer and minimizes interference between the sequentially learned tasks is developed by combining the meta-learning algorithm Reptile [31] with a reservoir sampling. The CL approach proposed in [19] learns and continuously adapts class prototypes, by building upon the meta-learning algorithm ProtoNets [32].

Using the bi-level optimization scheme introduced in Section II-C, methods were developed to meta-learn a parameter initialization that inhibits catastrophic forgetting [10], [18], [11]. Here, it is possible to learn an initialization for all model parameters [18] or learn an embedding network and an initialization for only the classifier network [10], [11]. In [11], a separate network is additionally trained to perform a task-specific feature weighting by modulating the output of the embedding network. The aforementioned works address CL by assuming that the classification tasks, which have to be learned, are *class-balanced*. The absence of any mechanism to cope with the extreme setting, where all the tasks are OCC tasks as in CAD, makes these approaches prone to overfitting to the majority class. In contrast, our approach inhibits this undesired phenomenon besides reducing catastrophic forgetting. We compare to the meta-learning based continual learning algorithm SeqFOMAML [18] in our experiments and show that it overfits to the majority class in the CAD problem setting.

B. Anomaly Detection and One-Class Classification

Typical anomaly detection (AD) approaches use SVMs to detect anomalous examples [33], [34], i.e. examples that do not belong to the normal class. When faced with high-dimensional data, e.g. images, feature extractors are used to embed the data into a lower-dimensional space before they are fed to the SVM-based classifier [35], [36], [37]. End-to-end deep learning methods were also proposed to tackle AD, by jointly training a feature extractor and a one-class classifier [38] or by using the reconstruction loss of autoencoders [39] to distinguish anomalies [40], [41], [42]. GAN-based [43] approaches were also used for AD [44], [45], [46]. Recently, an episodic data sampling strategy was proposed to adapt various class-balanced meta-learning algorithms to the AD setting [26]. Hereby, the bi-level optimization mechanism explained in Section II-C is used to find a model (initialization) that enables few-shot AD, i.e. learning a classification task by using only few examples from only its normal class.

All the aforementioned approaches yield a classification model that can detect the anomalies of a *single* AD task. In fact, they do not incorporate any feature to promote learning multiple tasks sequentially or inhibit catastrophic forgetting, which makes them unsuitable for the CAD problem setting. We propose a method that enables a model to sequentially learn *multiple* AD tasks with only minimal forgetting. In our experiments, we compare to the meta-learning algorithm OC-MAML [26], which yields an initialization tailored for learning AD tasks. Our results (Section V-B) show that it fails at sequentially learning several tasks.

IV. Approach: A Rapid Continual Anomaly Detector (ARCADe)

This work introduces *A Rapid Continual Anomaly Detector (ARCADe)*, a meta-learning algorithm designed to tackle the Continual Anomaly Detection (CAD) problem (Section II). ARCADe builds upon the bi-level optimization scheme introduced in Section II-C. Since meta-learning algorithms that use this optimization mechanism have been shown to be universal learning algorithm approximators [47], ARCADe should be able to approximate a learning algorithm tailored for the CAD

problem. In this section, we first present ARCADe using the CAD problem formulation from Section II-B. Subsequently, we explain the intuition behind meta-learning parameter-specific learning rates. Finally, we destinguish between two variants of ARCADe.

A. Algorithm

Our algorithm uses the meta-training set D_{tr} to learn an initialization θ as well as a learning rate α for each model parameter, as done in [23] to address the few-shot learning problem. Starting from this meta-learned initialization, learning a sequence S_{test} of unseen OCC tasks (by taking few gradient descent steps) using the meta-learned learning rates yields a model that has a high performance on all tasks included in S_{test}. The meta-training procedure of ARCADe is presented in Algorithm 1.

Algorithm 1 ARCADe Meta-training Procedure

Require: D_{tr}: Set of meta-training task-sequences
Require: β: Learning rate for the meta-update
Require: K: Adaptation set size
1: Randomly initialize model parameters $\boldsymbol{\theta}$ and parameter-specific learning rates $\boldsymbol{\alpha}$
2: **while** not done **do**
3: Sample a batch of task-sequences S_i from D_{tr}
4: Initialize meta-learning loss $L_{meta} = 0$
5: **for each** sampled S_i **do**
6: Initialize sequence adaptation loss $L_s = 0$
7: Initialize $\boldsymbol{\theta}'_{\boldsymbol{i,0}} = \boldsymbol{\theta}$ ($\boldsymbol{\theta}'_{\boldsymbol{i,0}} = \boldsymbol{\theta_{head}}$ if ARCADe-H)
8: **for** T_j in S_i with j in $\{1, ..., J = \textit{length}(S_i)\}$ **do**
9: Compute adapted parameters using K (normal) majority class examples from T_j^{tr}: $\boldsymbol{\theta}'_{\boldsymbol{i,j}} = \boldsymbol{\theta}'_{\boldsymbol{i,j-1}} - \boldsymbol{\alpha} \circ \nabla_{\boldsymbol{\theta}'_{\boldsymbol{i,j-1}}} L_{T_j^{tr}}(f_{\boldsymbol{\theta}'_{\boldsymbol{i,j-1}}})$
10: Compute $L_{T_j^{val}}(f_{\boldsymbol{\theta}'_{\boldsymbol{i,j}}})$ with the current adapted parameters $\boldsymbol{\theta}'_{\boldsymbol{i,j}}$ on the class-balanced val set T_j^{val}
11: $L_s = L_s + L_{T_j^{val}}(f_{\boldsymbol{\theta}'_{\boldsymbol{i,j}}})$
12: **end for**
13: **for** T_j in S_i **do**
14: Compute loss $L_{T_j^{val}}(f_{\boldsymbol{\theta}'_{\boldsymbol{i,J}}})$ with the final adapted parameters $\boldsymbol{\theta}'_{\boldsymbol{i,J}}$ on the val set T_j^{val}
15: $L_s = L_s + L_{T_j^{val}}(f_{\boldsymbol{\theta}'_{\boldsymbol{i,J}}})$
16: **end for**
17: $L_{meta} = L_{meta} + L_s$
18: **end for**
19: Update $(\boldsymbol{\theta}, \boldsymbol{\alpha})$: $(\boldsymbol{\theta}, \boldsymbol{\alpha}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{\alpha}) - \beta \nabla_{(\boldsymbol{\theta}, \boldsymbol{\alpha})} L_{meta}$
20: **end while**
21: **return** Meta-learned parameters $\boldsymbol{\theta}$ and learning rates $\boldsymbol{\alpha}$

In each meta-training iteration of ARCADe a batch of task-sequences is randomly sampled from D_{tr}. The current parameter initialization $\boldsymbol{\theta}$ is adapted to each sequence S_i by taking one (or more) gradient step(s) on the training sets T_j^{tr} of the tasks included in S_i sequentially. Hereby the gradient descent steps are performed using the current parameter-specific learning rates $\boldsymbol{\alpha}$. We note that in Algorithm 1 only one gradient descent update is performed (Operation 9) for simplicity of notation. Extending it to multiple updates is straightforward. We use the binary cross-entropy loss for all loss functions mentioned in Algorithm 1.

In the CAD problem setting (Section II-B), we consider anomaly detection tasks (or OCC tasks), i.e. each task T_j includes a training set T_j^{tr} with only majority class examples and a class-balanced validation set T_j^{val}. For each task T_j we compute the loss on the class-balanced held-out validation set T_j^{val} twice. The first time (Operation 10) is done directly after learning T_j by using the adapted model $\boldsymbol{\theta}'_{\boldsymbol{i,j}}$. This ensures a high model performance on the task immediately after it is learned. The second time (Operation 14) is conducted after learning all the tasks in S_i, i.e. using the final model adapted to that sequence $\boldsymbol{\theta}'_{\boldsymbol{i,J}}$. This maximizes the last model's performance on all the tasks in the sequence, hence minimizing catastrophic forgetting. These two losses are computed for *each* task in S_i and added to the sequence adaptation loss L_s. The model initialization and learning rates are updated in each meta-training iteration by minimizing L_{meta} which is the sum of the adaptation losses L_s of each sampled task-sequence S_i (Operation 19). In that sense, we can say that ARCADe explicitly optimizes for having a high performance on all tasks contained in a sequence, immediately after learning them and after having learned them all sequentially, while using only examples from their majority class.

In order to ensure that the model has a high performance on a task T_j at all points in time after learning it, one could compute the loss on its validation set T_j^{val} after learning *each* task T_k subsequent to T_j and add it to L_s. Here the loss would be computed using the current model parameters $\boldsymbol{\theta}'_{\boldsymbol{i,k}}$ after learning a task T_k. Doing this would minimize forgetting task T_j in all points in time while incrementally learning new tasks (T_k). However, in this case, the computational cost for computing L_s would increase exponentially with the length of the task-sequence, which does not scale for long task-sequences. Instead we approximate this additional optimization objective by adding to L_s the validation loss of *one* randomly sampled previous task T_j, every time a new task T_k in the sequence is learned. We note that this cannot be performed for the first task in the sequence, since it has no previous tasks. Even though we compute these additional loss terms and use them for our experimental evaluation, we do not mention them in Algorithm 1 for simplicity of notation.

Once meta-training is done, the best performing initialization and learning rates are used to learn task-sequences from the meta-testing set D_{test}. Here, the model initialization is sequentially adapted to the tasks from the test task-sequence using their training sets and the meta-learned learning rates, as

done during meta-training (Operations 8 and 9 in Algorithm 1). Thereafter the adapted model is evaluated on the class-balanced validation sets of these tasks, as done in meta-training (Operations 13 and 14 in Algorithm 1). We note that the selection of the best performing model initialization and learning rates is done by conducting validation episodes (adaptation and evaluation) using the task-sequences from the meta-validation set D_{val}, throughout meta-training.

B. Meta-Learning Parameter-Specific Learning Rates

In the following, we explain the intuition behind additionally meta-learning parameter-specific learning rates to tackle the CAD problem and not only the model initialization as it was done in [18], [10] and [11] in the class-balanced continual learning setting. We hypothesize that meta-learning parameter-specific learning rate enables the optimization algorithm to identify the parameters that are responsible for overfitting to the majority class and/or for catastrophic forgetting, and reduce their learning rates. Our results (Section V-B) confirm our intuition and show that additionally meta-learning parameter-specific learning rates leads to a more effective inductive bias for the CAD problem.

Before performing the adaptation updates (Operation 9 in Algorithm 1), we clip the learning rates to have values between 0 and 1. We do this to prevent them from having negative values, which would lead to taking gradient ascent steps on the task adaptation loss $L_{T_j^{tr}}$. The meta-update (Operation 19 in Algorithm 1) can indeed update the learning rates to have negative values since performing gradient ascent on the one-class training set of a task prevents overfitting to that class (by increasing the loss on that class). The lower overfitting to the majority class leads to a lower loss on the class-balanced validation set ($L_{T_j^{val}}(f_{\boldsymbol{\theta}'_{i,j}})$), which results in a lower L_{meta}. By clipping the negative learning rates to 0, we ensure that the corresponding parameter (responsible to overfitting to the majority class) is not updated during task-adaptation. It is considered as a task-agnostic parameter and is used as-is for all tasks, as opposed to other parameters which are updated to task-specific values. To speed-up meta-training, it is possible to conduct the first n meta-training iterations with constant learning rates, before meta-learning them along with the initialization (Operation 13 in Algorithm 1).

C. Variants of ARCADe

We distinguish two variants of ARCADe: ARCADe-M, which we introduced up to now, meta-learns an initialization and a learning rate for *all* model parameters, and ARCADe-H, which does the same but only for the parameters of the classification head, i.e. the output layer. For the parameters of the backbone layers, ARCADe-H does not learn an initialization but rather task-agnostic end values, which do not have to be updated depending on the task-sequence that has to be learned. When learning tasks sequentially ARCADe-H updates only the parameters of the output layer with their corresponding meta-learned learning rates. The only difference in the meta-learning procedure can be seen in Operation 7 from Algorithm 1. Meta-learning approaches that adapt only the classification head to learn unseen tasks were proposed in [25] and [10] to address the few-shot learning and the class-balanced continual learning problems, respectively.

V. Experimental Evaluation

We conduct experiments [1] in an attempt to answer the following key questions: (1) Can the proposed meta-learning algorithm cope with the challenges of the CAD problem, i.e. catastrophic forgetting and overfitting to the majority class, and how do its two variants, ARCADe-M and ARCADe-H, compare to each other? (2) How do previous meta-learning approaches for anomaly detection and class-balanced continual learning perform in the CAD setting? (3) Does meta-learning a learning rate for each parameter, besides the initialization, boost performance in a CAD context? (4) If yes, does the distribution of the meta-learned learning rates follow a pattern across datasets?

A. Baselines and Datasets

We evaluate the two variants of the proposed meta-learning Algorithm (ARCADe-M and ARCADe-H) on three different datasets which range from grey-scale images of letters to more challenging RGB natural images (Question 1). Besides we compare ARCADe to OC-MAML [26] and SeqFOMAML [18], which meta-learn model initializations that are tailored for anomaly detection and continual learning, respectively (Questions 2). We use the same evaluation procedure for ARCADe and the baselines: Task-sequences are sampled from the meta-testing set D_{test} and their tasks are learned sequentially using gradient descent. For a fairer comparison, we adapt SeqFOMAML to the anomaly detection scenario by using anomaly detection tasks for its meta-training. Note that SeqFOMAML samples the same number of examples from each class during the adaptation phase of its meta-training, i.e. it uses normal and anomalous examples for model adaptation during meta-training. Furthermore, we train ARCADe without meta-learning learning rates to investigate their impact when addressing a CAD problem (Question 3). Finally, we analyze the distribution and properties of the learning rates meta-learned by ARCADe (Question 4).

We evaluate ARCADe on three meta-learning benchmark datasets: Omniglot [48], MiniImageNet [49] and CIFAR-FS [50]. Omniglot is composed of 20 instances of 1623 hand-written character classes from 50 different alphabets. The images have the size 28x28 pixels. We use 25 alphabets for meta-training, 5 for meta-validation and 20 for meta-testing. MiniImageNet contains 100 classes from ImageNet where each class includes 600 images of size 84x84x3. We use the official data split of 64 classes for meta-training, 16 for meta-validation and 20 for meta-testing. CIFAR-FS was derived from CIFAR-100 by dividing its classes into 64

[1] Our code is made public under: https://github.com/AhmedFrikha/ARCADe-A-Rapid-Continual-Anomaly-Detector

classes for meta-training, 16 for meta-validation and 20 for meta-testing to make it suitable for meta-learning problems. Here, each class includes 600 images of size 32x32x3. The same data splits are used for ARCADe and the baselines.

To create meta-learning tasks for CAD, i.e. sequences of anomaly detection tasks as explained in Section II-B, we proceed as follows. First, we divide the classes available, e.g. the meta-training classes, into L disjoint sets of classes, where L is the task-sequence length. By building tasks using these sets we ensure that the tasks do not share any class. Subsequently, to create a task, one class from its set of classes is randomly chosen to be the normal class, i.e. its datapoints are labeled as non-anomalous, while the remaining classes are all labeled as anomalous. This ensures that the anomaly class has a higher variance than the normal class, which is usually the case in AD problems. Two disjoint sets of examples are then created from this task: a training set T^{tr} containing only normal class examples and a class-balanced validation set T^{val}. The tasks are then concatenated in a random order into a task-sequence. This task-sequence creation procedure is adopted to create meta-training, meta-validation and meta-testing task-sequences for the three datasets.

For ARCADe as well as for the baselines we use the same 4-module architecture used in [18] for continual learning. Each module includes a 3x3 convolutional layer, a 2x2 max-pooling layer, a batch-normalization [51] layer and a ReLU activation function. The 4 modules are followed by a linear layer and a sigmoid activation function. For omniglot, the convolutional layers include 64 filters, while for MiniImageNet and CIFAR-FS they include 32 filters. Since the meta-update of ARCADe requires backpropagating the gradients through all updates of all tasks, which is computationally expensive, we use a first-order approximation for our experiments. Hereby, the second-order terms of the derivatives are ignored, as done in [18].

B. Results and discussion

In this section we present and discuss the results of our experimental evaluation. Following previous continual learning works [7], [9] we consider the final retained accuracy, i.e. the average of the accuracies of the final model on the validation sets of all test tasks, as our main metric. We use task-sequences composed of 10 tasks for meta-training on Omniglot and 5 tasks for meta-training on MiniImageNet and CIFAR-FS. For meta-testing task-sequence lengths between 1 and 100 are used for Omniglot and between 1 and 5 for the more challenging MiniImageNet and CIFAR-FS. During meta-training and meta-testing, each task is learned by performing only 3 gradient descent updates and using only 10 normal examples, across all datasets. This extends ARCADe's applicability to *few-shot* CAD problems, i.e. CAD problems that exhibit extreme data scarcity. The performance of the two ARCADe variants and the baselines is shown in Figure 1 on Omniglot and in Figure 2 on MiniImageNet and CIFAR-FS. For all datasets, we report the retained accuracy averaged over 500 task-sequences from the meta-testing set D_{test}.

Fig. 1. Retained accuracy on Omniglot

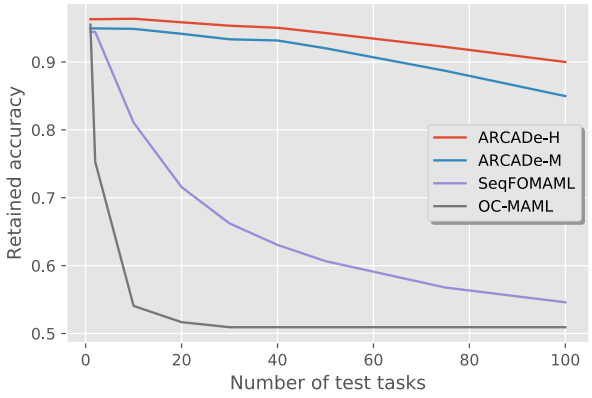
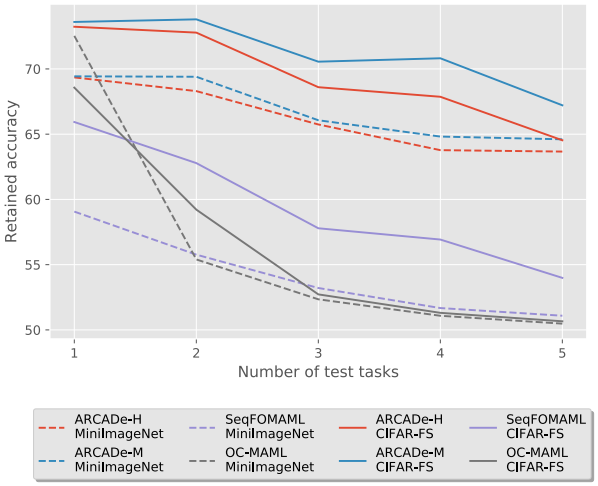


Fig. 2. Retained accuracy on MiniImageNet and CIFAR-FS



We find that both ARCADe variants substantially outperform the baselines for all sequences that include more than one task on all three datasets. While the model initialization meta-learned by SeqFOMAML slows down catastrophic forgetting when adapted to class-balanced tasks [18], it fails at retaining a high accuracy in the CAD problem setting i.e. when adapted to a sequence of *OCC* tasks. The quick decrease in retained accuracy suggests an important overfitting to the majority class. While OC-MAML yields a higher accuracy on the first task on MiniImageNet and Omniglot, it is not able to preserve this performance while learning the subsequent tasks in the sequence. In a CAD situation, the OC-MAML model quickly forgets the first task learned and collapses to a model that predicts only the majority class. We note that the lower performance of OC-MAML on the first task compared to the results reported in [26] is due to the different evaluation setting in the CAD problem, where the identifiers and training sets of the learned

tasks are not available at test time. OC-MAML uses the training set of the learned test task to overwrite the batch normalization statistics (mean and variance) before testing on the validation set.

Surprisingly, we find that ARCADe can learn up to 100 OCC tasks sequentially on Omniglot, while losing only 6% accuracy, even though it was trained with only 10-tasks sequences. We observe that ARCADe-H outperforms ARCADe-M on Omniglot, while ARCADe-M achieves higher retained accuracy on MiniImageNet and CIFAR-FS. Our explanation for this is that since MiniImageNet and CIFAR-FS have a higher variance in the input space, adapting the parameters of the feature extractor to the normal classes of the test tasks is beneficial. However, ARCADe-H can only adapt the parameters of the output layer, which results in a lower performance. The features meta-learned on the meta-training set of Omniglot, which includes by far more classes than the ones of MiniImageNet and CIFAR-FS, require less adaptation to perform well on the meta-testing set.

To assess the impact of meta-learning parameter specific learning rates, we evaluate ARCADe with constant learning rates, i.e. only parameter initializations are meta-learned. In Table I, we present the results in terms of retained accuracy on test task-sequences with the same length as the ones used for meta-training. We find that additionally meta-learning learning rates boosts the performance of both ARCADe variants across all datasets. This validates our hypothesis that additionally meta-learning learning rates leads to a more effective inductive bias for the addressed CAD problem.

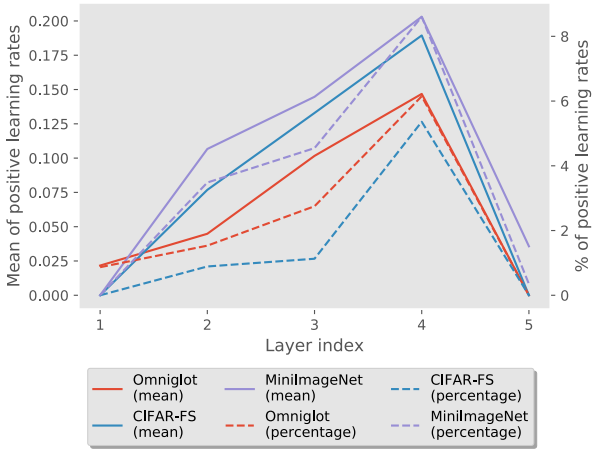
TABLE I
RETAINED TEST ACCURACIES OF ARCADE WITH AND WITHOUT META-LEARNING LEARNING RATES

Model \ Dataset	Omniglot	CIFAR-FS	MIN
ARCADe-M	**96.1**	**68.1**	**64.5**
ARCADe-M (constant α)	95.7	66.4	63.1
ARCADe-H	**96**	**67.8**	**64.1**
ARCADe-H (constant α)	95.6	66.8	63.0

Finally, we would like to investigate the characteristics of the meta-learned learning rates in order to gain a deeper insight into the learning strategy to which ARCADe-M converges. As mentioned in Section IV, we clip the learning rates between 0 and 1. Thus, only positive learning rates are active. We measure the percentage and mean of the positive (active) learning rate per neural network layer and present them in Figure 3.

The following observations and interpretations hold for all three datasets. We find that, for all layers, the majority of the learning rates are chosen to be not active, i.e. they converge to negative values. This suggests that most parameters are task-agnostic and can be used as-is independently of the task-sequence to be learned. As we progress in the layers of the embedding network (CNN), the percentage of active learning rates increases to reach its maximum at the last convolutional

Fig. 3. Layer-wise mean and percentage of positive learning rates meta-learned by ARCADe-M



layer. This means that, while the basic features (layer 1) can be reused without adaptation across tasks, the more sophisticated features that are used for classification (layer 4) have to be task-specifically adapted. Moreover, ARCADe-M freezes almost all the parameters in the linear output layer (less than 1% of the parameters are updated). This suggests that, each time it learns a new task, ARCADe-M does not update its normal class decision boundary to include the embeddings of the normal class examples of this new task, but rather changes the embedding of the latter to fit inside a frozen decision boundary. We hypothesize that ARCADe-M does this since the output layer is more prone to the CAD challenges, i.e. overfitting to the majority class and catastrophic forgetting.

Furthermore, we analyze the means of the active learning rates and find a similar trend across the layers. In fact, the few active learning rates in the first and last layer have substantially lower values than those of the other convolutional layers, especially layer 4. This shows that, during adaptation, bigger update steps are performed on the last layer of the embedding network than on the output layer, which backs our previous interpretation of the ARCADe-M's learning strategy. On the other hand, ARCADe-H cannot update the parameters of the embedding network by design, and learns therefore how to adapt the parameter of the output layer. Analyzing the parameter-specific learning rates meta-learned by ARCADe-H shows also that some parameters are chosen to be task-agnostic (due to negative learning rates), while other are chosen to be task-specific. This further explains the performance increase of ARCADe-H when additionally meta-learning learning rates (Table I).

VI. CONCLUSION

In this work we addressed the novel and challenging problem of Continual Anomaly Detection (CAD). After formulating this learning scenario as a meta-learning problem, we proposed *A Rapid Continual Anomaly Detector (ARCADe)*

to serve as a first and strong baseline in this research context. On the Omniglot dataset, our meta-learning approach enables sequentially learning up to 100 anomaly detection tasks using only examples from their normal (majority) class, with minimal forgetting and overfitting to the majority class. Our method substantially outperformed continual learning and anomaly detection baselines on three datasets.

REFERENCES

[1] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, 1999.

[2] B. F. Goodrich, "Neuron clustering for mitigating catastrophic forgetting in supervised and reinforcement learning," 2015.

[3] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, 2017.

[4] R. Caruana, "Multitask learning," *Machine learning*, 1997.

[5] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," *Proceedings of machine learning research*, 2017.

[6] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Advances in neural information processing systems*, 2017.

[7] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Advances in neural information processing systems*, 2017.

[8] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[9] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," 2018.

[10] K. Javed and M. White, "Meta-learning representations for continual learning," in *Advances in Neural Information Processing Systems*, 2019.

[11] S. Beaulieu, L. Frati, T. Miconi, J. Lehman, K. O. Stanley, J. Clune, and N. Cheney, "Learning to continually learn," 2020.

[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, 2009.

[13] R. Aljundi, K. Kelchtermans, and T. Tuytelaars, "Task-free continual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[14] S. Wang, L. L. Minku, and X. Yao, "A systematic study of online class imbalance learning with concept drift," *IEEE transactions on neural networks and learning systems*, 2018.

[15] M. M. Moya, M. W. Koch, and L. D. Hostetler, "One-class classifier networks for target recognition applications," *NASA STI/Recon Technical Report N*, 1993.

[16] S. S. Khan and M. G. Madden, "One-class classification: taxonomy of study and review of techniques," *The Knowledge Engineering Review*, 2014.

[17] J. Schmidhuber, "Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook," Ph.D. dissertation, Technische Universität München, 1987.

[18] G. Spigler, "Meta-learnt priors slow down catastrophic forgetting in neural networks," 2019.

[19] M. Zhang, T. Wang, J. H. Lim, G. Kreiman, and J. Feng, "Variational prototype replays for continual learning," 2019.

[20] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[21] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," 2016.

[22] Y. Wang and Q. Yao, "Few-shot learning: A survey," 2019.

[23] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few-shot learning," 2017.

[24] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[25] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, "Rapid learning or feature reuse? towards understanding the effectiveness of maml," 2019.

[26] A. Frikha, D. Krompaß, H.-G. Köpken, and V. Tresp, "Few-shot one-class classification via meta-learning," 2020.

[27] C. Yin, J. Tang, Z. Xu, and Y. Wang, "Adversarial meta-learning," 2018.

[28] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016.

[29] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015.

[30] X. He, J. Sygnowski, A. Galashov, A. A. Rusu, Y. W. Teh, and R. Pascanu, "Task agnostic continual learning via meta learning," 2019.

[31] A. Nichol and J. Schulman, "Reptile: a scalable metalearning algorithm," 2018.

[32] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017.

[33] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, 2001.

[34] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, 2004.

[35] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," *Procedings of the British Machine Vision Conference 2015*, 2015.

[36] J. T. Andrews, T. Tanay, E. J. Morton, and L. D. Griffin, "Transfer representation-learning for anomaly detection." ICML, 2016.

[37] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, 2016.

[38] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*, 2018.

[39] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, 2006.

[40] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 2002.

[41] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, 2015.

[42] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 2017.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014.

[44] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," *2017 IEEE International Conference on Image Processing (ICIP)*, 2017.

[45] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International Conference on Information Processing in Medical Imaging*.

[46] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[47] C. Finn and S. Levine, "Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm," 2017.

[48] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the annual meeting of the cognitive science society*, 2011.

[49] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.

[50] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," 2018.

[51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

Chapter 4

Discovery of New Multi-Level Features for Domain Generalization via Knowledge Corruption

Discovery of New Multi-Level Features for Domain Generalization via Knowledge Corruption

Ahmed Frikha
Siemens Technology
University of Munich (LMU)
ahmed.frikha@siemens.com

Denis Krompaß
Siemens Technology
denis.krompass@siemens.com

Volker Tresp
Siemens Technology
University of Munich (LMU)
volker.tresp@siemens.com

***Abstract*—Machine learning models that can generalize to unseen domains are essential when applied in real-world scenarios involving strong domain shifts. We address the challenging domain generalization (DG) problem, where a model trained on a set of source domains is expected to generalize well in unseen domains without any exposure to their data. The main challenge of DG is that the features learned from the source domains are not necessarily present in the unseen target domains, leading to performance deterioration. We assume that learning a richer set of features is crucial to improve the transfer to a wider set of unknown domains. For this reason, we propose COLUMBUS, a method that enforces new feature discovery via a targeted corruption of the most relevant input and multi-level representations of the data. We conduct an extensive empirical evaluation to demonstrate the effectiveness of the proposed approach which achieves new state-of-the-art results by outperforming 18 DG algorithms on multiple DG benchmark datasets in the DomainBed framework.**

I. Introduction

Deep learning models have achieved tremendous success when applied to independent and identically distributed (i.i.d.) data. However, in real-world applications, distribution shifts between training and test data are commonly encountered. For instance, data distributions might differ from one hospital to another [1], and from one production plant to another. Similarly, the models in self-driving cars are exposed to different urban and rural environments in different countries with changing weather conditions [2] and object poses [3].

Approaches to make machine learning models resilient to such data distribution changes were studied for different domain shift settings. For example, several domain adaptation methods were developed to address the case where, besides the data from the source domain(s), a set of labeled [5] or unlabeled [6] data is available from a specific target domain. However, in real-world scenarios, collecting data from the target domain(s) is often slow, e.g., a new hospital or production site, expensive, or even infeasible, e.g., collecting images from every street of every country in the context of self-driving cars. Sometimes, the target domains cannot be known beforehand.

The Domain Generalization (DG) problem [7], [8] was introduced to address such cases. Specifically, a model trained on multiple source domains is expected to directly perform well in unseen target domains without requiring any exposure to its data. This problem setting can be interpreted as multi-source 0-shot domain adaptation.

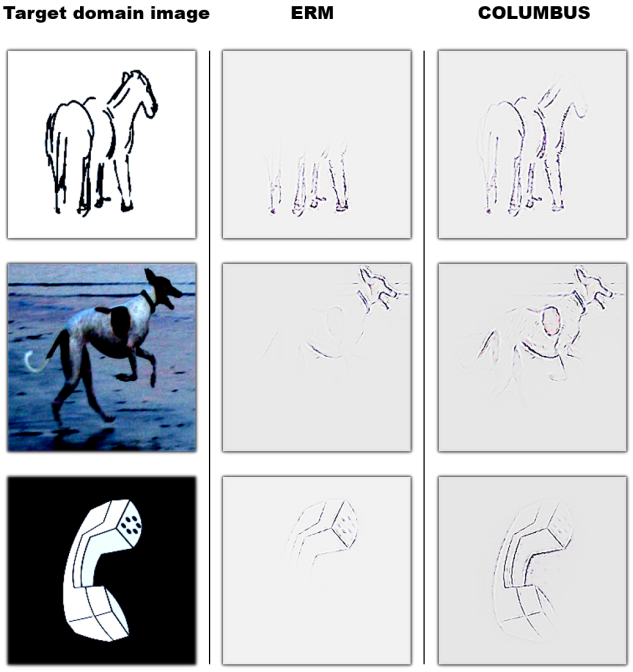


Fig. 1. Relevance maps computed with GuidedGrad-CAM [4] for ERM and COLUMBUS using images from the target domains, PACS *Sketch*, VLCS *VOC* and OfficeHome *Clipart*. COLUMBUS recognizes more features than ERM, including horse back and muzzle, dog legs and tail, and phone shape.

Training a model to generalize across several related but unseen data distributions remains arguably one of the most challenging open problems in machine learning. In the last decade, a plethora of widely different methods were developed to address the DG problem. We refer to [9] for an extensive overview of DG algorithms. Despite these efforts, [10] found that carefully tuning the baseline, which simply applies Empirical Risk Minimization (ERM) on the data of the source domains, achieves a high performance that is competitive with state-of-the-art methods.

One major challenge of DG is that the model can only observe and learn features from the source domains, which may not be present in the unseen target domains, limiting generalization. We presume that learning a wider set of different features would increase the chance of learning features that are useful for a larger set of unseen domains. Hence, we introduce

COLUMBUS, a training procedure for automated new feature discovery, which leads to a better feature recognition in unseen domains (Figure 1). During training on the source domains, COLUMBUS incentivizes the model to discover new features, even in data examples on which it already performs well. To achieve this, COLUMBUS prevents the model from using the features it deems most relevant for the source domains by corrupting them during training. To identify the most relevant features for a model, we leverage attribution methods [11] usually used for model explainability purposes.

We evaluate our approach on the recently proposed DomainBed framework [10] which includes several DG datasets and algorithm implementations to promote a fair and reproducible comparison of different approaches. Our method outperforms 18 DG algorithms evaluated on 3 datasets in the DomainBed framework, achieving new state-of-the-art results using 2 different model selection methods. Furthermore, our method achieves the highest performance when evaluated on unseen data from the source domains used for training (in-domain generalization), which confirms its effectiveness and ability to learn new features useful for unseen data.

II. Related Work

A. Domain Generalization

This section presents an overview of domain generalization (DG) approaches. Methods to which we compare in our experiments (Section IV) are highlighted in bold. We refer to [9] for an extensive overview of DG algorithms. The simplest approach to DG is to train one model via Empirical Risk Minimization (**ERM**) [12] on the training datasets of all source domains. **GroupDRO** [13] additionally increases the importance of source domains where the model yields a lower performance. In the following, we broadly categorize DG approaches into three categories.

Domain alignment methods aim to learn domain-invariant representations of the data by aligning features across the source domains. The reduction of the representation distribution mismatch across source domains can be achieved by minimizing the maximum mean discrepancy criteria [14] combined with an adversarial autoencoder (**MMD**) [15], minimizing the difference between the means [16] or covariance matrices (**CORAL**) [17] in the embedding space across different domains, or minimizing a contrastive loss [18]–[20], e.g., **SelfReg** [21]. Domain alignment is also performed by aligning the loss gradients across source domains via inner product maximization (**Fish**) [22], or binary (**AND-mask**) [23], [24] or continuous gradient masking (**SAND-mask**) [24].

Another line of works optimizes for features that confuse a domain discriminator model [25]–[28], and includes **DANN** [29] and its class-conditional extension **C-DANN** [30]. Other works additionally involve the classifier in the representation alignment, either by optimizing for an embedding space such that the optimal linear classifier on top of it is the same across different domains (**IRM**) [31], or by passing a domain-specific mean embedding to the classifier as a second argument (**MTL**) [32]. **VREx** [33] is an approximation of IRM via a variance penalty and **ARM** [34] is an extension of MTL that employs a separate embedding CNN.

Meta-learning techniques were applied to DG by training a model in a bi-level optimization scheme on meta-train and meta-test sets sampled from the source domains. Hereby, **MLDG** [35] optimizes for parameters that can be quickly adapted to different domains, MASF [1] adds inter-class and intra-class losses to regularize the embedding space, and MetaReg meta-learns a regularizer for the output layer [36].

Data augmentation approaches were proposed to tackle DG and our method falls into this category. Some works use **Mixup** [37] to compute inter-domain examples to augment the training set [38]–[40]. **SagNets** [41] reduce the domain gap by randomizing the style of images while keeping their content. Another line of works generate images by using adversarial attacks [42] to perturb input images based on a class classifier [43]–[45] or a domain classifier [46], by training CNNs to generate images within the source domains [47]–[49] or novel domains [50]–[52]. Other works apply such perturbations on a feature level [53], [54].

Our approach corrupts the raw input data as well as the multi-level representations that the model learns in order to enforce new feature discovery. Instead of using visually undetectable adversarial attacks or highly parametrized generative models, we employ attribution methods, e.g., Guided-Grad-CAM [4], to identify and corrupt the most relevant features. Our approach shares similarities with **RSC** [53] which discards the most dominant features fed to the output layer to promote the activation of the remaining features. The key difference of our approach is that we corrupt features not only in the last high-level representation space, i.e., the input to the output layer, but also in the raw input space and other low-level representation spaces. We argue that by discarding the features only in the representation space (e.g., elephant trunk detector), as done in RSC, the same silenced feature detectors can be relearned as long as the model is exposed to the corresponding features in the input space (e.g., the pixels of the elephant trunk). We hypothesize that corrupting the features in the input space is crucial to enforce the discovery of new features. Our empirical results show that our method outperforms RSC by a significant margin (Section IV) on unseen data from source and target domains, hence confirming our hypothesis.

B. Relevance Attribution

In an attempt to explain and interpret the predictions of deep learning models, several attribution methods that assign relevance scores to input features have been developed [4], [55], [56]. In Saliency Maps [55] the relevance scores are given by the gradient of the output neuron corresponding to the ground truth w.r.t. the input. Better attributions were achieved by averaging these gradients over local neighborhood patches in SmoothGrad [57] and over brightness level interpolations in IntegratedGradients [58]. Another category of approaches modifies the backpropagation procedure by considering only positive gradients [59] or to satisfy the relevance conservation property through the layers [60], [61]. Class Activation Maps

(CAM) [62] leverages the activations in the last convolutional layer to produce a heatmap highlighting the relevance of each feature in the raw input. Gradient-weighted CAM (Grad-CAM) [4] generalizes CAM to a variety of CNNs by using the gradient information flowing into the last convolutional layer. This method can be combined with GuidedBP [59] to yield GuidedGrad-CAM [4]. IBA [56] approximates attribution scores by restricting the information flow via noise injection to intermediate feature maps during the forward pass.

While prior works used attribution methods to explain and interpret model predictions, we leverage them for training purposes. To the best of our knowledge, we are the first to incorporate attribution methods combined with data corruption into training to improve the model's generalization ability. For a broader overview of attribution methods, we refer to [11].

III. Method

The proposed method improves the knowledge transfer to unknown data distributions by training a model to learn a rich set of features on several representation levels of the data via an automated new feature discovery.

Let F_s and F_t denote the sets of features learnable for the addressed classification task, which are present in the data of the source domains and in the target domain, respectively, and G their intersection. In the optimal case, the set of features L learned by the model on the source domains encompasses G fully. Since F_t is unknown at training time, our method maximizes the size of L by training the model to learn as many features as possible, resulting in a higher chance to capture features from G via the higher intersection between L and G. To achieve this, we propose COLUMBUS, a training procedure that enables automated new feature discovery. COLUMBUS prevents the model from using (a part of) the most relevant features for its current predictions during training. This is done in 3 major steps: identification of the most relevant features, their corruption, and training with the corrupted data representations. Figure 2 presents an overview of our approach. We apply this technique on several levels of representations of the data ranging from the raw input, e.g., pixels of the elephant trunk, to the high-level features fed to the output layers, e.g., elephant-trunk-detector, including the representations yielded by intermediate layers, hence fostering multi-level new feature discovery.

A. Identification

In each training iteration, we sample a method from a set of attribution methods A, and use it to compute an attribution map M that identifies the most relevant features. Any attribution method can be included in the set A. In this work, we use Saliency Maps [55] and GuidedGrad-CAM [4], since they are simple, fast, and model-architecture-agnostic. Other methods require modifications to support skip connections and batch normalization layers [60] or involve training additional parameters after each update [56]. Moreover, GuidedGrad-CAM was found to be competitive with the state-of-the-art attribution methods in the image degradation evaluation [56].

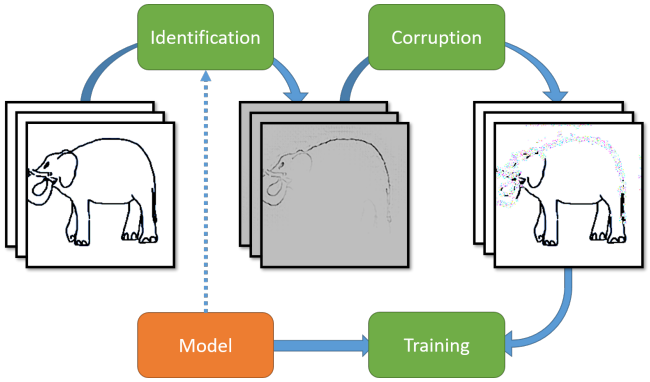


Fig. 2. Overview of the proposed COLUMBUS method. In the identification stage, the most class-discriminative features according to the current model are identified via a relevance attribution method, which in this case is applied to the raw input representation. In the corruption stage, the identified features, e.g., elephant trunk and back, are perturbed by using a corruption method, in this case a replacement by a random pixel. Finally, the model is trained with the batch of corrupted data, promoting the discovery of new features, e.g., elephant feet and toes. The image used belongs to the PACS Sketch domain.

While Saliency Maps and GuidedGrad-CAM were developed to assign relevance scores to features in the input space, we extend their usage to identify relevant features in representations extracted by intermediate layers. Let $\mathbf{y}$ and $\hat{\mathbf{y}}$ denote the ground truth and the model prediction for a datapoint $\mathbf{X}$. The attribution map M_l yielded by Saliency Maps for a representation R_l yielded by layer l is given by

$$M_{l,Saliency} = \frac{\partial(\hat{\mathbf{y}} \odot \mathbf{y})}{\partial R_l}. \tag{1}$$

Note that the original Saliency Maps method [55] corresponds to the case where $l = 0$, i.e., R_0 is the raw input representation.

The class-discriminative relevance map M_l yielded by Grad-CAM [4] for a layer l is given by a sum over the channels of the representation R_l weighted by importance factors α_c for each channel c, resulting in

$$M_{l,GradCAM} = ReLU(\sum_c \alpha_c R_l^c). \tag{2}$$

Hereby, the importance factors are given by the gradient of the model prediction for the correct class w.r.t. the global-average-pooled representation R_l. Formally,

$$\alpha_c = \frac{1}{Z} \sum_i \sum_j \frac{\partial(\hat{\mathbf{y}} \odot \mathbf{y})}{\partial R_l^{c,i,j}}. \tag{3}$$

Grad-CAM is applied to the representation yielded by the last convolutional layer to obtain a relevance map M which is upsampled to the input size [4]. For intermediate representation R_l, we use the corresponding relevance map M_l (Eq. 2). We use GuidedGrad-CAM [4] which yields more fine-grained maps than Grad-CAM by multiplying $M_{l,GradCAM}$ with the relevance maps determined by Guided Backpropagation (Guided BP) [59], for the same representation R_l. Guided BP modifies Saliency Maps by removing negative gradients when backpropagating through ReLU layers.

B. Corruption

In each training iteration, we sample a method from the set of corruption methods C and use it to corrupt the identified features based on the relevance attribution map M. Any technique that perturbs the information contained in the identified features can be used. We use different corruption methods depending on the sampled representation level l.

To corrupt the raw input ($l = 0$), the most relevant input features according to the attribution map M_0, e.g., the pixels corresponding to an elephant trunk, are perturbed using a corruption method. Hereby, we perturb the identified pixel values by setting them to a random value, to zero, i.e., black pixels, by applying the Fast Gradient Sign Method (FGSM) [42], or by applying Gaussian blurring. For an intermediate representation level $l > 0$, first the original input is fed through the model up to the corresponding layer l to yield the representation R_l. The latter is then corrupted based on the relevance attribution map M_l resulting from the identification stage and finally fed to the next layer. To corrupt intermediate embeddings, we drop the most relevant features, i.e., set their values to zero. This can be viewed as a targeted Dropout [63].

C. Training

The COLUMBUS training procedure is described by Algorithm 1. In each training iteration, a data batch B, a representation level l, an attribution method, and a corruption method are sampled. Subsequently, the aforementioned identification and corruption steps are performed. The model is trained on the corrupted data (representations). Hereby, the $p\%$ most relevant features are corrupted. To enable the model to learn some features at the beginning of training, p is set to 0, i.e., no corruption is applied. As training progresses, more features are corrupted in the data, forcing the model to discover and learn new features. Concretely, p is linearly increased throughout the training to reach p_{max}, a hyperparameter. Note that the resulting gradual learning of new features, independently from each other, promotes also feature disentanglement, which was found to be beneficial for visual reasoning [64]. We use different identification and corruption methods to increase the diversity of the corrupted datapoints used to train the model and prevent overfitting. We also found that sampling multiple methods leads to better empirical results. It should also be noted that COLUMBUS is adaptive to the model's learning, since the identification step is model-state-specific. In other words, if the model *forgets* a set of features during training, these will not be identified (again), and hence will not be corrupted (again), which enables the model to relearn them.

The model parameters θ are updated by minimizing a loss function f using a gradient-based optimization algorithm, e.g., Adam [65]. In algorithm 1, SGD is used for simplicity of notation. The loss function f used is a weighted sum comprising a classification loss L_{cls} and a domain-alignment regularization loss term L_{DA}. Formally,

$$f = L_{cls} + \lambda \sum_{i=1}^{N_s} \sum_{j=i+1}^{N_s} L_{DA}(i, j), \tag{4}$$

where λ is a weighting factor and N_s the number of source domains. We use cross-entropy as classification loss L_{cls}. L_{DA} regularizes the embedding space by minimizing the l_2-norm between the domain-specific embedding means and covariance matrices for each pair of source domains i and j, as in DDC [16] and CORAL [17] respectively. The normalization of the regularization loss by the number of source domain pairs is omitted for simplicity of notation. We note that, unlike prior works [10], [16], [17] that apply this loss on the representations of the original images, we use corrupted images and representations. This leads to an alignment in the embedding space not only across the domains, but also between class-specific features, e.g., by minimizing the difference between the embedding distribution of cartoon images of elephant feet and art painting images of elephant trunks.

Algorithm 1 The COLUMBUS training procedure

Require: D_s: Training data of all source domains
Require: f: Loss function
Require: α: Learning rate
Require: L: Set of representation levels including the raw input level
Require: A: Set of relevance attribution methods
Require: C: Set of corruption methods
Require: p_{max}: Max. % of representation to be corrupted

Initialize the model parameters θ randomly or from a pre-trained model
Initialize the % of representation to be corrupted $p = 0$
while not done **do**
  Sample a data batch $B = \{\mathbf{X}, \mathbf{y}\}$ from D_s
  Sample level of representation l from L
  Feed B through the model parametrized by θ
  Get the relevance attribution map M_l by applying a relevance attribution method randomly sampled from A on level l using the current model parameters θ
  if $l = 0$ **then**
    Corrupt the values in B corresponding to the $p\%$ highest values in M_0 by applying a corruption method randomly sampled from C, yielding the corrupted input B_c
    Feed B_c through the model to obtain the predictions $\hat{\mathbf{y}}$
  else
    Feed B through the model until level l to obtain the representation R_l
    Corrupt the values in R_l corresponding to the $p\%$ highest values in M_l, yielding the corrupted representation $R_{l,c}$
    Feed $R_{l,c}$ through the model starting from level $l+1$ to obtain the predictions $\hat{\mathbf{y}}$
  end if
  Update θ: $\theta \leftarrow \theta - \alpha \nabla_\theta f(\mathbf{y}, \hat{\mathbf{y}})$
  Increase p linearly towards p_{max}
end while
return Learned model parameters θ

In our experiments, we corrupt $q\%$ of the sampled data batch in each iteration, and increase q linearly during the training until q_{max} is reached, as done for the representation percentage to be corrupted p. This is omitted in Algorithm 1 for simplicity of notation. During training, we alternate between sampling an intermediate representation and the raw input for corruption. The intermediate representations correspond to the outputs of each ResNet block in the used ResNet-50 model [66]. At test time, the model trained with COLUMBUS is applied to the data from the target domains without any corruption.

IV. Experiments

A. Experimental Setup

We evaluate our approach empirically* on the recently proposed DomainBed framework [10] which includes several DG datasets, implementations of DG algorithms, and model selection methods. DomainBed promotes a fair and reproducible comparison of the different approaches by including a common automated hyperparameter search, i.e., a random search with the given seeds conducts the same experiments for all methods. For a fair comparison with the 18 DG algorithms, our experiments follow the same experimental setting adopted in DomainBed [10]: We use a ResNet-50 model [66] pretrained on ImageNet [67] with frozen batch normalization [68] statistics as suggested in [69], the same optimization algorithm, data augmentation techniques and number of training iterations used in DomainBed. The COLUMBUS-specific hyperparameters p_{max}, q_{max} and λ are included in the hyperparameter search of DomainBed, and the intervals used can be found in the Appendix. We noticed that the published code [10] with the provided seeds does not enable the reproduction of the published results, since the resulting points in the hyperparameter search space are different from the ones used for the published results. Therefore, for a fairer comparison, we additionally rerun the experiments of the best performing DG method in DomainBed, i.e., CORAL, with the published code and seeds that we used for COLUMBUS.

We conduct experiments on 3 challenging multi-domain datasets commonly used as DG benchmarks: VLCS [70], OfficeHome [71] and PACS [72]. VLCS contains images belonging to 5 classes from 4 photographic domains: VOC2007 (V), LabelMe (L), Caltech101 (C), and SUN09 (S). OfficeHome consists of images of 65 classes from the domains Art (A), Clipart (C), Product (P), and Real (R). PACS comprises images belonging to 7 classes from the domains Art-painting (A), Cartoon (C), Photo (P), and Sketch (S). DomainBed splits each source domain data into 80% for training and 20% for validation. Each experiment is run with the provided 3 seeds.

B. Results

Tables I, II and III show the results averaged over the 3 seeds pre-determined by DomainBed, on VLCS, PACS and OfficeHome respectively. Hereby, the unseen target domain is defined by the column name, i.e., the 3 other domains are used as source domains for training. The test accuracy is computed on the test set of the target domain. We provide results including standard deviations in the appendix. The average results over the domains of each dataset can be seen in Table IV. We select the model with the highest source-domain validation performance for the evaluation on the target domain.

TABLE I
Domain Generalization results on VLCS.

Algorithm	C	L	S	V	Avg
ERM	97.7	64.3	73.4	74.6	77.5
IRM	98.6	64.9	73.4	77.3	78.5
GroupDRO	97.3	63.4	69.5	76.7	76.7
Mixup	98.3	64.8	72.1	74.3	77.4
MLDG	97.4	65.2	71.0	75.3	77.2
CORAL	98.3	66.1	73.4	77.5	78.8
MMD	97.7	64.0	72.8	75.3	77.5
DANN	99.0	65.1	73.1	77.2	78.6
CDANN	97.1	65.1	70.7	77.1	77.5
MTL	97.8	64.3	71.5	75.3	77.2
SagNet	97.9	64.5	71.4	77.5	77.8
ARM	98.7	63.6	71.3	76.7	77.6
VREx	98.4	64.4	74.1	76.2	78.3
RSC	97.9	62.5	72.3	75.6	77.1
CORAL†	97.3	65.2	71.5	75.6	77.4
COLUMBUS	98.9	65.0	75.0	77.9	79.2

TABLE II
Domain Generalization results on PACS.

Algorithm	A	C	P	S	Avg
ERM	84.7	80.8	97.2	79.3	85.5
IRM	84.8	76.4	96.7	76.1	83.5
GroupDRO	83.5	79.1	96.7	78.3	84.4
Mixup	86.1	78.9	97.6	75.8	84.6
MLDG	85.5	80.1	97.4	76.6	84.9
CORAL	88.3	80.0	97.5	78.8	86.2
MMD	86.1	79.4	96.6	76.5	84.6
DANN	86.4	77.4	97.3	73.5	83.6
CDANN	84.6	75.5	96.8	73.5	82.6
MTL	87.5	77.1	96.4	77.3	84.6
SagNet	87.4	80.7	97.1	80.0	86.3
ARM	86.8	76.8	97.4	79.3	85.1
VREx	86.0	79.1	96.9	77.7	84.9
RSC	85.4	79.7	97.6	78.2	85.2
CORAL†	87.4	79.4	97.5	73.9	84.5
COLUMBUS	88.7	78.7	97.2	81.5	86.5

COLUMBUS achieves the highest results on all datasets on average, advancing the state-of-the-art by 1.6% and 1.2% compared to ERM and CORAL respectively. We note an impressive 5.5% improvement on OfficeHome's most challenging domain *Clipart* (**C**) compared to ERM and 3.3% compared to CORAL, on this 65-class classification task. Likewise, on the *Art* (**A**) domain of PACS, substantial 4% and 1.3% increases are observed compared to ERM and CORAL respectively. A significant performance increase is achieved

*Code under https://github.com/AhmedFrikha/columbus-domainbed.

†Results yielded by using published code [10] with the provided seeds.

TABLE III
DOMAIN GENERALIZATION RESULTS ON OFFICEHOME.

Algorithm	A	C	P	R	Avg
ERM	61.3	52.4	75.8	76.6	66.5
IRM	58.9	52.2	72.1	74.0	64.3
GroupDRO	60.4	52.7	75.0	76.0	66.0
Mixup	62.4	54.8	76.9	78.3	68.1
MLDG	61.5	53.2	75.0	77.5	66.8
CORAL	65.3	54.4	76.5	78.4	68.7
MMD	60.4	53.3	74.3	77.4	66.3
DANN	59.9	53.0	73.6	76.9	65.9
CDANN	61.5	50.4	74.4	76.6	65.8
MTL	61.5	52.4	74.9	76.8	66.4
SagNet	63.4	54.8	75.8	78.3	68.1
ARM	58.9	51.0	74.1	75.2	64.8
VREx	60.7	53.0	75.3	76.6	66.4
RSC	60.7	51.4	74.8	75.1	65.5
CORAL†	64.8	54.6	76.8	78.4	68.6
COLUMBUS	62.8	57.9	75.5	77.9	68.5

TABLE IV
AVERAGE DOMAIN GENERALIZATION RESULTS.

Algorithm	VLCS	PACS	OfficeHome	Avg
ERM	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	76.5
IRM	78.5 ± 0.5	83.5 ± 0.8	64.3 ± 2.2	75.5
GroupDRO	76.7 ± 0.6	84.4 ± 0.8	66.0 ± 0.7	75.7
Mixup	77.4 ± 0.6	84.6 ± 0.6	68.1 ± 0.3	76.7
MLDG	77.2 ± 0.4	84.9 ± 1.0	66.8 ± 0.6	76.3
CORAL	78.8 ± 0.6	86.2 ± 0.3	68.7 ± 0.3	77.9
MMD	77.5 ± 0.9	84.6 ± 0.5	66.3 ± 0.1	76.2
DANN	78.6 ± 0.4	83.6 ± 0.4	65.9 ± 0.6	76.0
CDANN	77.5 ± 0.1	82.6 ± 0.9	65.8 ± 1.3	75.3
MTL	77.2 ± 0.4	84.6 ± 0.5	66.4 ± 0.5	76.1
SagNet	77.8 ± 0.5	86.3 ± 0.2	68.1 ± 0.1	77.4
ARM	77.6 ± 0.3	85.1 ± 0.4	64.8 ± 0.3	75.8
VREx	78.3 ± 0.2	84.9 ± 0.6	66.4 ± 0.6	76.5
RSC	77.1 ± 0.5	85.2 ± 0.9	65.5 ± 0.9	75.9
SelfReg	77.8 ± 0.9	85.6 ± 0.4	67.9 ± 0.7	77.1
Fish	77.8 ± 0.3	85.5 ± 0.3	68.6 ± 0.4	77.3
AND-mask	78.1 ± 0.9	84.4 ± 0.9	65.6 ± 0.4	76.0
SAND-mask	77.4 ± 0.2	84.6 ± 0.9	65.8 ± 0.4	75.9
CORAL†	77.4 ± 0.3	84.5 ± 0.5	68.6 ± 0.2	76.9
COLUMBUS	79.2 ± 0.2	86.5 ± 0.4	68.5 ± 0.4	78.1

on PACS's challenging *Sketch* (**S**) domain as well. On all target domains, COLUMBUS consistently outperforms all the baselines or yields a competitive performance. The fact that COLUMBUS outperforms RSC [53] confirms our hypothesis, that corrupting the learned features in the raw input is crucial to prevent relearning the same high-level features, and hence enforce new feature discovery.

We also evaluate our approach using the oracle selection method [10], where the model is evaluated on a held-out validation set from the target domain. In order to limit access to the target domain, this evaluation is performed only once at the end of each training, disallowing early stopping. The average results are presented in Table V. We find that the performance advantage of COLUMBUS is increased when better proxies for model selection, e.g., a held-out set from the target domain, are available, further confirming the effectiveness of our approach. Our results on DOMAINBED using both model selection methods show that the additional features learned thanks to the corruption of the most relevant features are useful for generalization to unseen domains. This is backed by Figure 1, where COLUMBUS recognizes more features in examples from the unseen target domain than ERM.

TABLE V
DOMAIN GENERALIZATION RESULTS USING THE TEST-DOMAIN VALIDATION SET (ORACLE) AS A SELECTION METHOD.

Algorithm	VLCS	PACS	OfficeHome	Avg
ERM	77.6 ± 0.3	86.7 ± 0.3	66.4 ± 0.5	76.9
IRM	76.9 ± 0.6	84.5 ± 1.1	63.0 ± 2.7	74.8
GroupDRO	77.4 ± 0.5	87.1 ± 0.1	66.2 ± 0.6	76.9
Mixup	78.1 ± 0.3	86.8 ± 0.3	68.0 ± 0.2	77.6
MLDG	77.5 ± 0.1	86.8 ± 0.4	66.6 ± 0.3	77.0
CORAL	77.7 ± 0.2	87.1 ± 0.5	68.4 ± 0.2	77.7
MMD	77.9 ± 0.1	87.2 ± 0.1	66.2 ± 0.3	77.1
DANN	79.7 ± 0.5	85.2 ± 0.2	65.3 ± 0.8	76.8
CDANN	79.9 ± 0.2	85.8 ± 0.8	65.3 ± 0.5	77.0
MTL	77.7 ± 0.5	86.7 ± 0.2	66.5 ± 0.4	77.0
SagNet	77.6 ± 0.1	86.4 ± 0.4	67.5 ± 0.2	77.2
ARM	77.8 ± 0.3	85.8 ± 0.2	64.8 ± 0.4	76.1
VREx	78.1 ± 0.2	87.2 ± 0.6	65.7 ± 0.3	77.0
RSC	77.8 ± 0.6	86.2 ± 0.5	66.5 ± 0.6	76.8
AND-mask	76.4 ± 0.4	86.4 ± 0.4	66.1 ± 0.2	76.3
SAND-mask	76.2 ± 0.5	85.9 ± 0.4	65.9 ± 0.5	76.0
CORAL†	77.4 ± 0.6	85.6 ± 0.8	68.4 ± 0.4	77.1
COLUMBUS	77.7 ± 0.4	88.2 ± 0.2	69.6 ± 0.4	78.5

Finally, we investigate whether the richer set of features learned by COLUMBUS leads to a better in-domain generalization, i.e., whether a performance boost is also yielded on unseen source domain data. We evaluate COLUMBUS and the DG baselines on the held-out validation sets of the source domains and report the maximal mean validation accuracy across domains in the Appendix. COLUMBUS consistently achieves the highest validation performance on the training domains compared to the DG baselines. This shows that the richer set of learned features improves generalization to unseen in-distribution datapoints, suggesting that COLUMBUS might also be suitable for applications without domain shift.

V. CONCLUSION

In this work, we proposed COLUMBUS, a novel and strong domain generalization (DG) approach that enforces new feature discovery to improve the transfer to a wider set of unseen domains. During training, COLUMBUS corrupts the input and multi-level representations of the data most relevant for the model. For the identification of such features, relevance attribution methods that are usually used for model explainability purposes are leveraged. Our extensive empirical evaluation on DOMAINBED demonstrates the effectiveness of the proposed method, which outperforms 18 DG algorithms and achieves new state-of-the-art results on multiple DG benchmarks. Our results show that the richer set of learned features improves the generalization to unseen data from both seen and unseen domains, suggesting the suitability of our approach for applications beyond domain generalization to include scenarios without domain shift.

APPENDIX

Experimental Setting Details In this section we provide further details about the experiments conducted. The experiments were conducted on computing instances that include a Tesla T4 NVIDIA GPU, 8 custom Intel Cascade Lake CPUs and 32 Gb of memory. The operating system used is Ubuntu 20.04 LTS. The libraries PyTorch [73] and TorchVision were used with the versions 1.7.1 and 0.8.2, respectively.

In our experiments, the percentage of representation corrupted p and the percentage of the batch corrupted q are increased linearly towards p_{max} and q_{max}, respectively, during the first half of the training. In the second half of the training, the maximum values are used.

For a fair comparison, we used the automated hyperparameter search from DOMAINBED [10] for each domain and dataset. Hereby, each hyperparameter search involves 20 random search experiments, i.e., the hyperparameters are randomly sampled from the specified intervals. To distribute the hyperparameter search experiments over multiple devices (each experiment runs on a single GPU), we used the Ray Tune package [74], [75]. Our experiments follow the experimental setting: We use a ResNet-50 model [66] pretrained on ImageNet [67] with frozen batch normalization [68] statistics as suggested in [69], as well as the same optimization algorithm, ADAM [65], data augmentation techniques, and number of training iterations. An overview of the hyperparameter-specific intervals we used for COLUMBUS can be seen in Table VI. The algorithm-specific hyperparameter intervals used for the other DG algorithms can be found in [10]. Depending on whether the corruption is applied on the input or an intermediate representation, different value intervals were used for the percentage of the representation corrupted p and the percentage of the batch corrupted q. For the hyperparameters related to intermediate representations, i.e., $p_{max,intermediate}$ and $q_{max,intermediate}$, the interval upper bounds were chosen based on the results of RSC [53], which discards the most dominant features fed to the output layer, i.e., the last representation level. We used the same intervals used in DOMAINBED for the other algorithms for all hyperparameters.

Source Domain Generalization

In this section, we investigate whether the richer set of features learned by COLUMBUS leads to a better in-domain generalization, i.e., whether a performance boost is also yielded on unseen data from the source domains used for training. We evaluate COLUMBUS and the DG baselines on the held-out validation sets of the source domains and report the maximal average validation accuracy across domains in Table VII*.

COLUMBUS consistently achieves the highest validation performance on the training domains compared to the DG baselines, on every dataset. This shows that the richer set of learned features improves generalization to unseen in-distribution data examples as well, suggesting that COLUMBUS might be suitable for applications beyond domain generalization to include scenarios without domain shift.

Results including standard deviations We present the domain generalization results of COLUMBUS and the baselines, including the standard deviations computed over the 3 runs with the seeds provided by DOMAINBED in Tables VIII, IX and X. Hereby, for model selection, the *training-domain validation-set* from DOMAINBED is used.

*For the baselines, we computed the results using the logs made public in https://drive.google.com/file/d/16VFQWTble6-nB5AdXBtQpQFwjEC7CChM/view?usp=sharing.

TABLE VI
HYPERPARAMETER INTERVALS USED FOR THE HYPERPARAMETER SEARCH CONDUCTED WITH DOMAINBED

Hyperparameter	**Random Distribution**
Weighting coefficient λ	$10^{Uniform(-1,1)}$
Max. corruption % for input representation $p_{max,input}$	$Uniform(0.2, 0.5)$
Max. corruption % for intermediate representation $p_{max,intermediate}$	$Uniform(0.01, 0.333)$
Max. batch corruption % for input representation $q_{max,input}$	$Uniform(0.2, 1.0)$
Max. batch corruption % for intermediate representation $q_{max,intermediate}$	$Uniform(0.1, 0.5)$

TABLE VII
SOURCE DOMAIN VALIDATION PERFORMANCE.

Algorithm	**VLCS**	**PACS**	**OfficeHome**	**Avg**
ERM	86.4 ± 0.0	97.0 ± 0.1	82.1 ± 0.2	88.5
IRM	85.8 ± 0.2	96.5 ± 0.4	79.9 ± 2.0	87.4
GroupDRO	86.4 ± 0.0	96.9 ± 0.1	81.6 ± 0.2	88.3
Mixup	86.6 ± 0.1	97.4 ± 0.1	83.2 ± 0.3	89.0
MLDG	86.4 ± 0.1	97.1 ± 0.1	82.4 ± 0.3	88.6
CORAL	86.5 ± 0.0	97.1 ± 0.1	83.7 ± 0.2	89.1
MMD	86.4 ± 0.1	96.9 ± 0.0	82.0 ± 0.1	88.4
DANN	86.3 ± 0.0	96.4 ± 0.3	80.4 ± 0.9	87.7
CDANN	86.4 ± 0.1	96.4 ± 0.3	80.5 ± 0.9	87.8
MTL	86.3 ± 0.0	97.0 ± 0.0	81.7 ± 0.2	88.3
SagNet	86.4 ± 0.0	97.0 ± 0.2	82.9 ± 0.4	88.8
ARM	86.3 ± 0.0	96.5 ± 0.1	80.2 ± 0.2	87.7
VREx	86.2 ± 0.1	96.9 ± 0.1	81.8 ± 0.4	88.3
RSC	86.4 ± 0.3	96.8 ± 0.2	81.5 ± 0.3	88.2
CORAL†	86.6 ± 0.1	96.8 ± 0.2	83.6 ± 0.0	89.0
COLUMBUS	86.6 ± 0.1	97.3 ± 0.0	83.4 ± 0.1	89.1

TABLE VIII
DOMAIN GENERALIZATION RESULTS ON VLCS, INCLUDING STANDARD DEVIATION.

Algorithm	**C**	**L**	**S**	**V**	**Avg**
ERM	97.7 ± 0.4	64.3 ± 0.9	73.4 ± 0.5	74.6 ± 1.3	77.5
IRM	98.6 ± 0.1	64.9 ± 0.9	73.4 ± 0.6	77.3 ± 0.9	78.5
GroupDRO	97.3 ± 0.3	63.4 ± 0.9	69.5 ± 0.8	76.7 ± 0.7	76.7
Mixup	98.3 ± 0.6	64.8 ± 1.0	72.1 ± 0.5	74.3 ± 0.8	77.4
MLDG	97.4 ± 0.2	65.2 ± 0.7	71.0 ± 1.4	75.3 ± 1.0	77.2
CORAL	98.3 ± 0.1	66.1 ± 1.2	73.4 ± 0.3	77.5 ± 1.2	78.8
MMD	97.7 ± 0.1	64.0 ± 1.1	72.8 ± 0.2	75.3 ± 3.3	77.5
DANN	99.0 ± 0.3	65.1 ± 1.4	73.1 ± 0.3	77.2 ± 0.6	78.6
CDANN	97.1 ± 0.3	65.1 ± 1.2	70.7 ± 0.8	77.1 ± 1.5	77.5
MTL	97.8 ± 0.4	64.3 ± 0.3	71.5 ± 0.7	75.3 ± 1.7	77.2
SagNet	97.9 ± 0.4	64.5 ± 0.5	71.4 ± 1.3	77.5 ± 0.5	77.8
ARM	98.7 ± 0.2	63.6 ± 0.7	71.3 ± 1.2	76.7 ± 0.6	77.6
VREx	98.4 ± 0.3	64.4 ± 1.4	74.1 ± 0.4	76.2 ± 1.3	78.3
RSC	97.9 ± 0.1	62.5 ± 0.7	72.3 ± 1.2	75.6 ± 0.8	77.1
CORAL†	97.3 ± 0.3	65.2 ± 0.5	71.5 ± 0.6	75.6 ± 0.9	77.4
COLUMBUS	98.9 ± 0.2	65.0 ± 1.3	75.0 ± 0.2	77.9 ± 0.9	79.2

TABLE IX
DOMAIN GENERALIZATION RESULTS ON PACS, INCLUDING STANDARD DEVIATION.

Algorithm	**A**	**C**	**P**	**S**	**Avg**
ERM	84.7 ± 0.4	80.8 ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
IRM	84.8 ± 1.3	76.4 ± 1.1	96.7 ± 0.6	76.1 ± 1.0	83.5
GroupDRO	83.5 ± 0.9	79.1 ± 0.6	96.7 ± 0.3	78.3 ± 2.0	84.4
Mixup	86.1 ± 0.5	78.9 ± 0.8	97.6 ± 0.1	75.8 ± 1.8	84.6
MLDG	85.5 ± 1.4	80.1 ± 1.7	97.4 ± 0.3	76.6 ± 1.1	84.9
CORAL	88.3 ± 0.2	80.0 ± 0.5	97.5 ± 0.3	78.8 ± 1.3	86.2
MMD	86.1 ± 1.4	79.4 ± 0.9	96.6 ± 0.2	76.5 ± 0.5	84.6
DANN	86.4 ± 0.8	77.4 ± 0.8	97.3 ± 0.4	73.5 ± 2.3	83.6
CDANN	84.6 ± 1.8	75.5 ± 0.9	96.8 ± 0.3	73.5 ± 0.6	82.6
MTL	87.5 ± 0.8	77.1 ± 0.5	96.4 ± 0.8	77.3 ± 1.8	84.6
SagNet	87.4 ± 1.0	80.7 ± 0.6	97.1 ± 0.1	80.0 ± 0.4	86.3
ARM	86.8 ± 0.6	76.8 ± 0.5	97.4 ± 0.3	79.3 ± 1.2	85.1
VREx	86.0 ± 1.6	79.1 ± 0.6	96.9 ± 0.5	77.7 ± 1.7	84.9
RSC	85.4 ± 0.8	79.7 ± 1.8	97.6 ± 0.3	78.2 ± 1.2	85.2
CORAL†	87.4 ± 0.3	79.4 ± 0.3	97.5 ± 0.1	73.9 ± 1.8	84.5
COLUMBUS	88.7 ± 0.8	78.7 ± 1.0	97.2 ± 0.1	81.5 ± 1.5	86.5

TABLE X
DOMAIN GENERALIZATION RESULTS ON OFFICEHOME, INCLUDING STANDARD DEVIATION.

Algorithm	**A**	**C**	**P**	**R**	**Avg**
ERM	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5
IRM	58.9 ± 2.3	52.2 ± 1.6	72.1 ± 2.9	74.0 ± 2.5	64.3
GroupDRO	60.4 ± 0.7	52.7 ± 1.0	75.0 ± 0.7	76.0 ± 0.7	66.0
Mixup	62.4 ± 0.8	54.8 ± 0.6	76.9 ± 0.3	78.3 ± 0.2	68.1
MLDG	61.5 ± 0.9	53.2 ± 0.6	75.0 ± 1.2	77.5 ± 0.4	66.8
CORAL	65.3 ± 0.4	54.4 ± 0.5	76.5 ± 0.1	78.4 ± 0.5	68.7
MMD	60.4 ± 0.2	53.3 ± 0.3	74.3 ± 0.1	77.4 ± 0.6	66.3
DANN	59.9 ± 1.3	53.0 ± 0.3	73.6 ± 0.7	76.9 ± 0.5	65.9
CDANN	61.5 ± 1.4	50.4 ± 2.4	74.4 ± 0.9	76.6 ± 0.8	65.8
MTL	61.5 ± 0.7	52.4 ± 0.6	74.9 ± 0.4	76.8 ± 0.4	66.4
SagNet	63.4 ± 0.2	54.8 ± 0.4	75.8 ± 0.4	78.3 ± 0.3	68.1
ARM	58.9 ± 0.8	51.0 ± 0.5	74.1 ± 0.1	75.2 ± 0.3	64.8
VREx	60.7 ± 0.9	53.0 ± 0.9	75.3 ± 0.1	76.6 ± 0.5	66.4
RSC	60.7 ± 1.4	51.4 ± 0.3	74.8 ± 1.1	75.1 ± 1.3	65.5
CORAL†	64.8 ± 0.2	54.6 ± 0.7	76.8 ± 0.6	78.4 ± 0.3	68.6
COLUMBUS	62.8 ± 0.3	57.9 ± 0.8	75.5 ± 0.1	77.9 ± 0.5	68.5

REFERENCES

[1] Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker, "Domain generalization via model-agnostic learning of semantic features," *NeurIPS*, 2019.

[2] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in *IEEE/CVF ICCV*, 2019.

[3] M. A. Alcorn, Q. Li, Z. Gong, C. Wang, L. Mai, W.-S. Ku, and A. Nguyen, "Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects," in *IEEE/CVF CVPR*, 2019.

[4] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *IEEE ICCV*, 2017.

[5] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, 2018.

[6] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2020.

[7] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," *NeurIPS*, 2011.

[8] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *ICML*, 2013.

[9] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *arXiv:2103.02503*, 2021.

[10] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," *arXiv:2007.01434*, 2020.

[11] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[12] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, 1999.

[13] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization," *arXiv:1911.08731*, 2019.

[14] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *JMLR*, 2012.

[15] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *IEEE CVPR*, 2018.

[16] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv:1412.3474*, 2014.

[17] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *ECCV*, 2016.

[18] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *IEEE ICCV*, 2017.

[19] C. Yoon, G. Hamarneh, and R. Garbi, "Generalizable feature learning in the presence of data bias and domain class imbalance with application to skin lesion classification," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2019.

[20] D. Mahajan, S. Tople, and A. Sharma, "Domain generalization using causal matching," *arXiv:2006.07500*, 2020.

[21] D. Kim, S. Park, J. Kim, and J. Lee, "Selfreg: Self-supervised contrastive regularization for domain generalization," *arXiv:2104.09841*, 2021.

[22] Y. Shi, J. Seely, P. H. Torr, N. Siddharth, A. Hannun, N. Usunier, and G. Synnaeve, "Gradient matching for domain generalization," *arXiv:2104.09937*, 2021.

[23] G. Parascandolo, A. Neitz, A. Orvieto, L. Gresele, and B. Schölkopf, "Learning explanations that are hard to vary," *arXiv:2009.00329*, 2020.

[24] S. Shahtalebi, J.-C. Gagnon-Audet, T. Laleh, M. Faramarzi, K. Ahuja, and I. Rish, "Sand-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization," *arXiv:2106.02266*, 2021.

[25] I. Albuquerque, J. Monteiro, M. Darvishi, T. H. Falk, and I. Mitliagkas, "Generalizing to unseen domains via distribution matching," *arXiv:1911.00804*, 2019.

[26] R. Shao, X. Lan, J. Li, and P. C. Yuen, "Multi-adversarial discriminative deep domain generalization for face presentation attack detection," in *IEEE/CVF CVPR*, 2019.

[27] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Correlation-aware adversarial domain adaptation and generalization," *Pattern Recognition*, 2020.

[28] Z. Deng, F. Ding, C. Dwork, R. Hong, G. Parmigiani, P. Patil, and P. Sur, "Representation via representations: Domain generalization via adversarially learned invariant representations," *arXiv:2006.11478*, 2020.

[29] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *JMLR*, 2016.

[30] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao, "Deep domain generalization via conditional invariant adversarial networks," in *ECCV*, 2018.

[31] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv:1907.02893*, 2019.

[32] G. Blanchard, A. A. Deshmukh, U. Dogan, G. Lee, and C. Scott, "Domain generalization by marginal transfer learning," *arXiv:1711.07910*, 2017.

[33] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville, "Out-of-distribution generalization via risk extrapolation (rex)," in *ICML*, 2021.

[34] M. Zhang, H. Marklund, N. Dhawan, A. Gupta, S. Levine, and C. Finn, "Adaptive risk minimization: A meta-learning approach for tackling group distribution shift," *arXiv:2007.02931*, 2020.

[35] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *AAAI Conference on Artificial Intelligence*, 2018.

[36] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "Metareg: Towards domain generalization using meta-regularization," *NeurIPS*, 2018.

[37] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv:1710.09412*, 2017.

[38] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang, "Adversarial domain adaptation with domain mixup," in *AAAI Conference on Artificial Intelligence*, 2020.

[39] S. Yan, H. Song, N. Li, L. Zou, and L. Ren, "Improve unsupervised domain adaptation with mixup training," *arXiv:2001.00677*, 2020.

[40] Y. Wang, H. Li, and A. C. Kot, "Heterogeneous domain generalization via domain mixup," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[41] H. Nam, H. Lee, J. Park, W. Yoon, and D. Yoo, "Reducing domain gap via style-agnostic networks," *arXiv e-prints*, 2019.

[42] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv:1412.6572*, 2014.

[43] A. Sinha, H. Namkoong, R. Volpi, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," *arXiv:1710.10571*, 2017.

[44] R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," *arXiv:1805.12018*, 2018.

[45] F. Qiao, L. Zhao, and X. Peng, "Learning to learn single domain generalization," in *IEEE/CVF CVPR*, 2020.

[46] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi, "Generalizing across domains via cross-gradient training," *arXiv:1804.10745*, 2018.

[47] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Multi-component image translation for deep domain generalization," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[48] N. Somavarapu, C.-Y. Ma, and Z. Kira, "Frustratingly simple domain generalization via image stylization," *arXiv:2006.11207*, 2020.

[49] F. C. Borlino, A. D'Innocente, and T. Tommasi, "Rethinking domain generalization baselines," in *International Conference on Pattern Recognition (ICPR)*, 2021.

[50] F. Maria Carlucci, P. Russo, T. Tommasi, and B. Caputo, "Hallucinating agnostic images to generalize across domains," in *IEEE/CVF ICCV Workshops*, 2019.

[51] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Deep domain-adversarial image generation for domain generalisation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 025–13 032.

[52] ——, "Learning to generate novel domains for domain generalization," in *European conference on computer vision*. Springer, 2020, pp. 561–578.

[53] Z. Huang, H. Wang, E. P. Xing, and D. Huang, "Self-challenging improves cross-domain generalization," in *Computer Vision–ECCV 2020: 16th European Conference*, 2020.

[54] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," *arXiv:2104.02008*, 2021.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[56] K. Schulz, L. Sixt, F. Tombari, and T. Landgraf, "Restricting the flow: Information bottlenecks for attribution," *arXiv:2001.00396*, 2020.

[57] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv:1706.03825*, 2017.

[58] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *ICML*, 2017.

[59] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv:1412.6806*, 2014.

[60] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, 2015.

[61] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, 2017.

[62] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE CVPR*, 2016.

[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *JMLR*, 2014.

[64] S. van Steenkiste, F. Locatello, J. Schmidhuber, and O. Bachem, "Are disentangled representations helpful for abstract visual reasoning?" *arXiv:1905.12506*, 2019.

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016.

[67] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, 2015.

[68] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[69] S. Seo, Y. Suh, D. Kim, G. Kim, J. Han, and B. Han, "Learning to optimize domain specific normalization for domain generalization," in *ECCV*, 2020.

[70] C. Fang, Y. Xu, and D. N. Rockmore, "Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias," in *IEEE ICCV*, 2013.

[71] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *IEEE CVPR*, 2017.

[72] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *IEEE ICCV*, 2017.

[73] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[74] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan *et al.*, "Ray: A distributed framework for emerging {AI} applications," in *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, 2018, pp. 561–577.

[75] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, "Tune: A research platform for distributed model selection and training," *arXiv preprint arXiv:1807.05118*, 2018.

Chapter 5

Towards Data-Free Domain Generalization

Towards Data-Free Domain Generalization

Ahmed Frikha[*,†]
Siemens Technology
University of Munich

Haokun Chen[*,†]
Siemens Technology
Technical University of Munich

Denis Krompaß
Siemens Technology

Thomas Runkler
Siemens Technology
Technical University of Munich

Volker Tresp
Siemens Technology
University of Munich

Abstract

In this work, we investigate the unexplored intersection of domain generalization and data-free learning. In particular, we address the question: How can knowledge contained in models trained on different source data domains be merged into a single model that generalizes well to unseen target domains, in the absence of source and target domain data? Machine learning models that can cope with domain shift are essential for for real-world scenarios with often changing data distributions. Prior domain generalization methods typically rely on using source domain data, making them unsuitable for private decentralized data. We define the novel problem of Data-Free Domain Generalization (DFDG), a practical setting where models trained on the source domains separately are available instead of the original datasets, and investigate how to effectively solve the domain generalization problem in that case. We propose DEKAN, an approach that extracts and fuses domain-specific knowledge from the available teacher models into a student model robust to domain shift. Our empirical evaluation demonstrates the effectiveness of our method which achieves first state-of-the-art results in DFDG by significantly outperforming ensemble and data-free knowledge distillation baselines.

1 Introduction

Deep learning methods have achieved impressive performance in a wide variety of tasks where the data is independent and identically distributed. However, real-world scenarios usually involve a distribution shift between the training data used during development and the test data faced at deployment time. In such situations, deep learning models often suffer from a performance degradation and fail to generalize to the out-of-distribution (OOD) data from the target domain [62, 66, 17, 21]. For instance, this domain shift problem is encountered when applying deep learning models on MRI data from different clinical centers that use different scanners [10]. Domain Adaptation (DA) approaches [71, 73] assume access to data from the source domain(s) for training as well as target domain data for model adaptation. However, data collection from the target domain can sometimes be expensive, slow, or infeasible, e.g. self-driving cars have to generalize to a variety of weather conditions [80] and object poses [3] in urban and rural environments from different countries. In this work, we focus on the Domain Generalization (DG) [5, 48] setting, where a model trained on multiple source domains is applied without any modification to unseen target domains.

In the last decade, a plethora of DG methods requiring only access to the source domains were proposed [86]. Nevertheless, the assumption that access to source domain data can always be granted

[*]Equal contribution.
[†]Primary contact authors: ahmed.frikha@siemens.com and haokun.chen@siemens.com

Workshop on Distribution Shifts, 35th Conference on Neural Information Processing Systems (NeurIPS 2021).

does not hold in many cases. For instance, General Data Protection Regulation (GDPR) prohibits the access to sensitive data that might identify individuals, e.g. bio-metric data or other confidential information. Likewise, some commercial entities are not willing to share their original data to prevent competitive disadvantage. Furthermore, as datasets get larger, their release, transfer, storage and management can become prohibitively expensive [39]. To circumvent the concerns related to releasing the original dataset, the data owners might want to share a model trained on their data instead. In light of increasing data privacy concerns, this alternative has recently enjoyed a surge of interest [44, 7, 50, 37, 33, 28, 78, 1].

Although Data-Free Knowledge Distillation (DFKD) methods were developed to transfer knowledge from a teacher model to a student model without any access to the original data [39, 44, 7, 50, 78, 9], only single-teacher scenarios with no domain shift were studied. On the other hand, Source-Free Domain Adaptation (SFDA) approaches were proposed to tackle the domain shift problem setting where one [37, 33, 28, 70, 11] or multiple [1] models trained on source domain data are available instead of the original dataset(s). Nonetheless, they require access to data from the target domain. In this work, we investigate the unstudied intersection of Domain Generalization and Data-Free Learning. Data-Free Domain Generalization (DFDG) is a problem setting that assumes only access to models trained on the source domains, without requiring data from source or target domains. Hereby, the goal is to have a single model able to generalize to unseen domains without any modification or data exposure, as it is the case in DG. To the best of our knowledge, we are the first to address this problem setting. Works addressing related problems are discussed in Appendix A.

Our contribution is threefold: Firstly, we introduce and define the novel and practical DFDG problem setting. Secondly, we tackle it by proposing a first and strong approach that merges the knowledge stored in the domain-specific models via the generation of synthetic data and distills it into a single model. Thirdly, we demonstrate the effectiveness of our method by empirically evaluating it on two DG benchmark datasets.

2 Approach

2.1 Problem statement

Let D_s^i and D_t^j denote the datasets available from the source and target domains respectively with $i = 1, .., I$ and $j = 1, .., J$. Hereby, I and J denote the number of source and target domains respectively. In the Domain Generalization (DG) [5, 48] problem setting, the goal is to train a model on the source domain data D_s^i in a way that enables generalization to a priori unavailable target domain data D_t^j, without any model modification at test time. We consider the source-data-free scenario of this problem where the source domain datasets D_s^i are not accessible, e.g., due to privacy, security, safety or commercial concerns, and models trained on these domain-specific datasets separately are available instead.

We refer to the source domain models as teacher models T_i as in the knowledge distillation literature [22]. We assume that the teacher models were trained without the prior knowledge that they would be used in a DFDG setting, i.e., their training does not involve any domain shift robustness mechanism. Hence, the application scenarios where the source domain data is not accessible anymore, e.g., was deleted, are also considered. We refer to this novel learning scenario as *Data-Free Domain Generalization (DFDG)*. The major difference with Source-Free Domain Adaptation (SFDA) [37, 33, 28] is the absence of target domain data D_t^j in DFDG.

The DFDG problem is a prototype for a practical use case where a model robust to domain shifts is needed and models trained on the same task but different data domains are available. This problem definition is motivated by the question: How can we amalgamate the knowledge from multiple models trained on different domains into a single model that is able to generalize to unseen target domains without any data exposure?

2.2 Domain Entanglement via Knowledge Amalgamation from Domain-Specific Networks

We propose Domain Entanglement via Knowledge Amalgamation from domain-specific Networks (DEKAN). Our approach tackles the challenges of DFDG in 3 stages: Knowledge extraction, fusion and transfer. In the first stage, *Intra-Domain Data-Free Knowledge Extraction*, we extract the

knowledge from the different source domain teacher models separately. Hereby, we generate domain-specific synthetic datasets via inceptionism-style [46] image synthesis, i.e., we initialize random noise images $\hat{x}$ and optimize them to be recognized as a sample from a pre-defined class by a trained domain-specific model. In particular, we apply the data-free knowledge distillation method described in [78, 83] to invert each domain-specific teacher separately. In the second stage, *Cross-Domain Data-Free Knowledge Fusion*, DEKAN generates cross-domain synthetic data by leveraging all pairs of inter-domain model-dataset combinations. Here, the cross-domain examples are optimized to be recognizable by teacher models trained on different domains. In the final stage, *Multi-Domain Knowledge Distillation*, DEKAN transfers the extracted knowledge from the domain-specific teachers to a student model via knowledge distillation using the generated data. At test time, i.e., deployment phase, the resulting student model is evaluated on target domain data without any modification. Details about the first stage as well as DEKAN's complete algorithm can be found in Appendix B. In the following, we focus on the second and third stages.

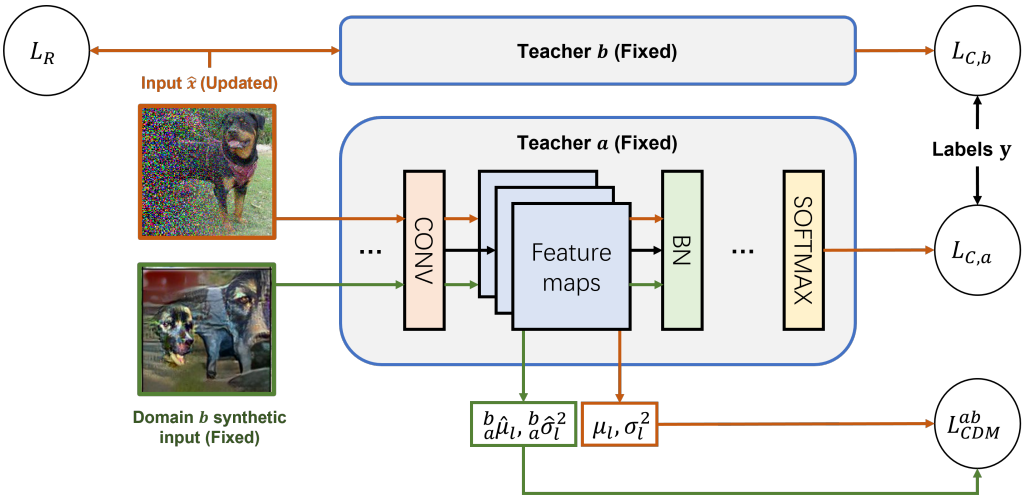


Figure 1: Overview of the Cross-Domain Data-Free Knowledge Fusion.

In the second stage, we propose a technique to merge the knowledge from two domains by generating cross-domain synthetic images that capture class-discriminative features present in the two domains, and match the distribution of intermediate features extracted by a domain-specific model from images of another domain. Let T_a and T_b denote the teacher models, and D_g^a and D_g^b the synthetic data generated in the first stage (Appendix B.0.1), specific to two domains a and b. We generate synthetic images D_g^{ab} by minimizing the cross-domain inversion loss L_{CD}^{ab}, that we formulate as

$$L_{CD}^{ab} = L_C(T_a(\hat{x}), y) + L_C(T_b(\hat{x}), y) + \alpha_1 L_R(\hat{x}) + \alpha_2 L_{CDM}^{ab}(\hat{x}), \tag{1}$$

where L_C denotes the classification loss, e.g., cross-entropy, L_R an image prior regularization, L_{CDM} the cross-domain feature moment matching loss, and α_1 and α_2 weighting coefficients. L_R penalizes the l_2-norm and the total variation of the image to ensure the convergence to valid natural images [42, 52, 46, 78]. We incentivize the generated images to contain class-discriminative features from both domains by minimizing the classification loss using both teachers. We hypothesize that images that can be recognized by models trained on different domains capture more domain-agnostic semantic features than those generated by inverting a single domain-specific model as done in prior works [78].

In addition, the cross-domain feature distribution matching loss L_{CDM}^{ab} optimizes the cross-domain synthetic images D_g^{ab} so that their feature distribution matches the distribution of the features extracted by T_a, the model trained on domain a, for images D_g^b synthesized from domain b. Note that $L_{CDM}^{ab} \neq L_{CDM}^{ba}$ and that using the model T_b and the data generated by inverting T_a in the first stage, i.e., D_g^a, would yield the cross-domain images D_g^{ba} that are different from D_g^{ab}. Formally,

$$L_{CDM}^{ab}(\hat{x}) = \sum_{l} max(\|\mu_l(\hat{x}) - {}_{a}^{b}\hat{\mu}_l\|_2 - {}_{a}^{b}\delta_l, 0) + \sum_{l} max(\|\sigma_l^2(\hat{x}) - {}_{a}^{b}\hat{\sigma}_l^2\|_2 - {}_{a}^{b}\gamma_l, 0). \quad (2)$$

L_{CDM}^{ab} minimizes the l_2-norm between the BN-statistics of the synthetic data, $\mu_l(\hat{x})$ and $\sigma_l^2(\hat{x})$, and target statistics, at each BN layer l. Here, the target statistics, ${}_{a}^{b}\hat{\mu}_l$ and ${}_{a}^{b}\hat{\sigma}_l^2$, are computed in a way that involves knowledge from different domains. In particular, they result from feeding the synthetic data specific to domain b through the teacher model trained on data from domain a, and computing the first two feature moments, i.e., mean and variance, for each BN layer. The intention behind this is to synthesize images that capture the features learned by the model on domain a that are activated and recognized when exposed to images from domain b. We hypothesize that such images would encompass domain-agnostic semantic information that would be useful for training a single model resilient to domain shift in the next stage.

We relax L_{CDM} by allowing the BN-statistics of the synthetic input to fluctuate within a certain interval. Here, we compute the relaxation constants ${}_{a}^{b}\delta_l$ and ${}_{a}^{b}\gamma_l$ as the ϵ_{CD} percentile of the distribution of differences between the stored BN-statistics, i.e., computed on the original domain a images, and those computed using the images D_g^b synthesized from the domain b teacher model in the first stage. Note that $\epsilon_{CD} = 100\%$ corresponds to synthesized images $\hat{x}$ yielding the BN-statistics from domain a, i.e., stored in model T_a, would not be penalized, i.e., $L_{CDM}^{ab} = 0$. This stage can be viewed as a domain augmentation, since the synthesized images D_g^{ab} do not belong neither to domain a nor to domain b. The synthesis of cross-domain data is applied to all possible domain pairs.

In the final DEKAN stage, the domain-specific and cross-domain knowledge, which is captured in the synthetic data generated in the first and second stages respectively, is transferred to a single student model S. To this end, we use knowledge distillation [22], i.e., we train the student model to mimic the predictions of the teachers for the synthetic data. As described in Equation 3, we minimize the Kullback-Leibler divergence D_{KL} between the predictions of the student S and the teacher(s) corresponding to the synthetic image $\hat{x}$. In particular, if the data example is domain-specific, i.e., it was generated in the first DEKAN stage, the predictions of the corresponding teacher are used as soft labels to train the student. For the cross-domain synthetic images that were generated in the second stage, the average predictions of the two corresponding teachers is used instead. The aggregation of the prediction distributions of two domain-specific teacher models contributes to the knowledge amalgamation across domains.

$$L_{KD} = D_{KL}(S(\hat{x}) \,||\, p) \quad \text{with} \quad p = \begin{cases} T_i(\hat{x}), & \text{if } \hat{x} \in D_g^i \quad \text{(domain-specific)} \\ \frac{1}{2}(T_i(\hat{x}) + T_j(\hat{x})), & \text{if } \hat{x} \in D_g^{ij} \quad \text{(cross-domain)} \end{cases} \quad (3)$$

3 Experiments and Results

The conducted experiments[3] aim to tackle the following key questions: (a) How does DEKAN compare to leveraging the domain specific models directly to make predictions on data from unseen domains? (b) How does our approach compare to data-free knowledge distillation methods applied to each domain separately? (c) How much does the unavailability of data cost in terms of performance?

We design baseline methods to address the novel DFDG problem, and compare them with DEKAN. The first category of baselines applies the available domain-specific models on the data from the target domains (Question (a)). We consider two ensemble baselines that aggregate the predictions of these models, e.g., by taking the average of the model predictions (**AvgPred**), or by taking the prediction of the most confident model, i.e., the model with the lowest entropy (**HighestConf**). Besides, we implement oracle methods that evaluate each of the domain-specific models separately on the target domain and then report the results of the best model (**BestTeacher**). Furthermore, we propose a baseline that applies an improved version [83] of DeepInversion (DI) [78] on each of the models separately to generate domain-specific synthetic images used to then train a student model via knowledge distillation (**Multi-DI**; Question (b)). Note that Multi-DI is equivalent to the application of DEKAN's first and third stage. Finally, we compare DEKAN to an upper-bound baseline that uses

[3]Code will be made public upon paper acceptance.

the original data from the source domains to train a single model via Empirical Risk Minimization (**ERM**) [68, 20], a common domain generalization baseline (Question (c)).

Algorithm	**Art Painting**	**Cartoon**	**Photo**	**Sketch**	**Average**
Ensemble - AvgPred	79.88	65.40	96.35	79.46	80.27
Ensemble - HighestConf	82.28	65.96	96.59	76.86	80.42
Multi-DI	82.13	72.14	95.57	73.75	80.90
DEKAN (ours)	83.01	75.94	96.29	80.17	**83.88**
BestTeacher (oracle)	75.24	62.80	96.41	69.76	76.05
ERM [20] (not data-free)	86.0	81.8	96.8	80.4	86.2

Algorithm	**MNIST**	**MNIST-M**	**SVHN**	**USPS**	**Average**
Ensemble - AvgPred	97.85	45.83	31.33	96.12	67.78
Ensemble - HighestConf	98.52	46.71	30.45	96.47	68.04
Multi-DI	93.31	54.04	36.72	96.53	70.15
DEKAN (ours)	94.64	55.86	39.15	96.77	**71.61**
BestTeacher (oracle)	99.27	48.33	38.11	97.73	70.86
ERM (not data-free)	98.22	55.18	50.13	96.54	75.02

Table 1: Domain Generalization results on PACS (top) and Digits (bottom).

We evaluate DEKAN and the baselines on two DG benchmark datasets, PACS [30] and Digits, which comprises images from MNIST [29], MNIST-M [15], SVHN [51] and USPS [24]. Table 1 shows the results of DEKAN and the baselines. Hereby, the column name refers to the unseen target domain, i.e., the 3 other domains are the source domains used to train the teacher models. The test accuracy is computed on the test set of the target domain.

DEKAN outperforms all data-free baselines on both datasets on average, setting a first state-of-the-art performance for the novel DFDG problem. We find that generative approaches, i.e., Multi-DI and DEKAN, outperform the ensemble methods on average, suggesting that training a single model on data from different domains enables a better aggregation of knowledge than the aggregation of domain-specific model predictions. Most importantly, DEKAN substantially outperforms Multi-DI, highlighting the importance of the synthesized cross-domain images. This is especially the case for the challenging domains, i.e., the domains where all the methods yield the lowest performance. In particular, the generation of cross-domain synthetic data leads to performance improvements of 6.4% and 3.8% on the Sketch and Cartoon PACS domains respectively, as well as a 2.4% increase on the SVHN domain of Digits. Additionally, we note the positive knowledge transfer across domains on the PACS dataset, as all the multi-domain methods outperform the oracle BestTeacher baseline that uses a single domain-specific teacher model, i.e., the teacher that achieves the highest performance on a validation set from the target domain. Finally, it is worth noting that while DEKAN significantly reduces the gap between the best data-free baseline and the upper-bound baseline that uses the original data, there is still potential for improvement.

4 Conclusion

This work addressed the unstudied intersection of domain generalization and data-free learning, a practical setting where a model robust to domain shifts is needed and the available models were trained on the same task but with data from different domains. We proposed DEKAN, an approach that fuses domain-specific knowledge from the available teacher models into a single student model that can generalize to data from a priori unknown domains. Our empirical evaluation demonstrated the effectiveness of our method which outperformed ensemble and data-free knowledge distillation baselines, hence achieving first state-of-the-art results in the novel and challenging data-free domain generalization problem.

References

[1] Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10103–10112, 2021.

[2] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

[3] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[4] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 2018.

[5] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 2011.

[6] Francesco Cappio Borlino, Antonio D'Innocente, and Tatiana Tommasi. Rethinking domain generalization baselines. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021.

[7] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3514–3522, 2019.

[8] Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2534–2543, 2021.

[9] Yoojin Choi, Jihwan Choi, Mostafa El-Khamy, and Jungwon Lee. Data-free network quantization with adversarial knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 710–711, 2020.

[10] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 2019.

[11] Cian Eastwood, Ian Mason, Christopher KI Williams, and Bernhard Schölkopf. Source-free adaptation to measurement shift via bottom-up feature restoration. *arXiv preprint arXiv:2107.05446*, 2021.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[13] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[14] Ahmed Frikha, Denis Krompaß, and Volker Tresp. Columbus: Automated discovery of new multi-level features for domain generalization via knowledge corruption. *arXiv preprint arXiv:2109.04320*, 2021.

[15] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[16] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 2016.

[17] Robert Geirhos, Carlos R Medina Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. *arXiv preprint arXiv:1808.08750*, 2018.

[18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[19] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 2012.

[20] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[21] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[23] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference*, 2020.

[24] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 2015.

[26] Daehee Kim, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. *arXiv preprint arXiv:2104.09841*, 2021.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020.

[29] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[30] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[31] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[32] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[33] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

[34] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[35] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[36] Yuhang Li, Feng Zhu, Ruihao Gong, Mingzhu Shen, Fengwei Yu, Shaoqing Lu, and Shi Gu. Learning in school: Multi-teacher knowledge inversion for data-free quantization. *arXiv preprint arXiv:2011.09899*, 2020.

[37] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[38] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2604–2613, 2019.

[39] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

[40] Liangchen Luo, Mark Sandler, Zi Lin, Andrey Zhmoginov, and Andrew Howard. Large-scale generative data-free distillation. *arXiv preprint arXiv:2012.05578*, 2020.

[41] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. *arXiv preprint arXiv:2006.07500*, 2020.

[42] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[43] Fabio Maria Carlucci, Paolo Russo, Tatiana Tommasi, and Barbara Caputo. Hallucinating agnostic images to generalize across domains. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[44] Paul Micaelli and Amos Storkey. Zero-shot knowledge transfer via adversarial belief matching. *arXiv preprint arXiv:1905.09768*, 2019.

[45] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv preprint arXiv:1711.05852*, 2017.

[46] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015.

[47] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[48] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, 2013.

[49] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap via style-agnostic networks. *arXiv e-prints*, 2019.

[50] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pages 4743–4751. PMLR, 2019.

[51] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[52] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[53] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[54] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668*, 2018.

[55] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[56] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. Multi-component image translation for deep domain generalization. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[57] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. Self-supervised knowledge distillation for few-shot learning. *arXiv preprint arXiv:2006.09785*, 2020.

[58] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[59] Soroosh Shahtalebi, Jean-Christophe Gagnon-Audet, Touraj Laleh, Mojtaba Faramarzi, Kartik Ahuja, and Irina Rish. Sand-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization. *arXiv preprint arXiv:2106.02266*, 2021.

[60] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018.

[61] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

[62] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[63] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

[64] Nathan Somavarapu, Chih-Yao Ma, and Zsolt Kira. Frustratingly simple domain generalization via image stylization. *arXiv preprint arXiv:2006.11207*, 2020.

[65] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, 2016.

[66] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[67] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[68] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 1999.

[69] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018*, 2018.

[70] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[71] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 2018.

[72] Yufei Wang, Haoliang Li, and Alex C Kot. Heterogeneous domain generalization via domain mixup. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[73] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2020.

[74] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[75] Zheng Xu, Yen-Chang Hsu, and Jiawei Huang. Training shallow and thin networks for acceleration via knowledge distillation with conditional adversarial networks. *arXiv preprint arXiv:1709.00513*, 2017.

[76] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.

[77] Hao-Wei Yeh, Baoyao Yang, Pong C Yuen, and Tatsuya Harada. Sofa: Source-data-free feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 474–483, 2021.

[78] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[79] Chris Yoon, Ghassan Hamarneh, and Rafeef Garbi. Generalizable feature learning in the presence of data bias and domain class imbalance with application to skin lesion classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2019.

[80] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[81] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[82] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[83] Xiangguo Zhang, Haotong Qin, Yifu Ding, Ruihao Gong, Qinghua Yan, Renshuai Tao, Yuhang Li, Fengwei Yu, and Xianglong Liu. Diversifying sample generation for accurate data-free quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15658–15667, 2021.

[84] Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, Yaxian Li, Zhichao Song, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source distilling domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12975–12983, 2020.

[85] Brady Zhou, Nimit Kalra, and Philipp Krähenbühl. Domain adaptation through task distillation. In *European Conference on Computer Vision*, pages 664–680. Springer, 2020.

[86] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *arXiv preprint arXiv:2103.02503*, 2021.

[87] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[88] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021.

A Related Work

Our method addresses the Data-Free Domain Generalization (DFDG) problem. To the best of our knowledge, we are the first to address this problem. In the following, we discuss approaches to related problem settings.

A.1 Domain Generalization

Domain Generalization (DG) approaches can be broadly classified into three categories. Domain alignment methods attempt to learn a domain-invariant representation of the data from the source domains by regularizing the learning objective. Variants of such a regularization include the minimization across the source domains of the maximum mean discrepancy criteria (MMD) [19, 32], the minimization of a distance metric between the domain-specific means [67] or covariance matrices [65], the minimization of a contrastive loss [47, 79, 41, 26], or the maximization of loss gradient alignment [61, 59]. Other works use adversarial training with a domain discriminator model [16, 34] for the same purpose. Another category of works leverages meta-learning techniques, e.g., the bi-level optimization scheme proposed in [12], to optimize for quick adaptation to different domains [31], or to learn how to regularize the output layer [4]. A combination of meta-learning and embedding space regularization is proposed in [10]. Another line of works augment the training data to tackle DG. On the one hand, some approaches perturb the source domain data by computing inter-domain examples [74, 76, 72] via Mixup [82], by randomizing the style of images [49], by computing adversarial examples [18] using a class classifier [63, 69, 55] or a domain classifier [60], or corrupting learned features to incentivize new feature discovery [14]. On the other hand, CNNs are trained to generate new images from the source domains [56, 64, 6] or from novel domains [43, 87]. Other works perturb intermediate representations of the data [23, 88, 14]. We refer to [86] for a more extensive overview of DG approaches.

Unlike standard DG approaches that require access to the source domain datasets, our method merges the domain-specific knowledge from models trained on the source domains into a single model resilient to domain shift, while preserving data privacy.

A.2 Knowledge Distillation

Knowledge distillation (KD) [22] was originally proposed to compress the knowledge of a large teacher network into a smaller student network. Several KD extensions and improvements [58, 81, 75, 2, 53] enabled its application to a variety of scenarios including quantization [45, 54], domain adaptation [84, 85], semantic segmentation [38], and few-shot learning [57, 8]. While these methods rely on the original data, Data-Free Knowledge Distillation (DFKD) methods were recently developed [39, 44, 50, 7]. Hereby, knowledge is transferred from one [44, 50, 7, 9, 78, 40, 83] or multiple [36] teacher(s) to the student model via the generation of synthetic data, either by optimizing random noise examples [50, 78, 83] or by training a generator network [44, 7, 9, 40]. Nevertheless, the aforementioned DFKD methods focus on scenarios without any domain shift, i.e. the student is evaluated on examples from the same data distribution used for training the teacher. In the DFDG problem setting we address, the student is trained from multiple teachers that are trained on different source domains in a way that enables generalization to data from unseen target domains. We propose a baseline that extends the usage of a recent DFKD method [83] to the DFDG setting, and compare it to our approach (Section 3).

A.3 Source-free domain adaptation

The recently addressed Source-Free Domain Adaptation problem [37, 33, 28] assumes access to one or multiple model(s) trained on the source domains, as well as data examples from a specific target domain. Proposed approaches to tackle it include the combination of generative models with a regularization loss [33], a feature alignment mechanism [77], or a weighting of the target domain samples by their similarity to the source domain [28]. SHOT [37] employs an information maximization loss along with a self-supervised pseudo-labeling, and is extended to the multi-source scenario via source model weighting [1]. BUFR [11] aligns the target domain feature distribution with the one from the source domain. Another line of works leverage Batch Normalization (BN) [25] layers by replacing the BN-statistics computed on the source domain with those computed on the

target domain [35], or by training the BN-parameters on the target domain via entropy minimization [70]. While these approaches rely on the availability of data from a known target domain, we address the DFDG scenario where the model is expected to generalize to *a priori unknown* target domain(s) without any modification or exposure to their data. We also note that some methods [28, 37, 11] modify the training procedure on the source domain, which would not be possible in cases where the data is not accessible anymore.

B More details about DEKAN

In the following, we introduce the first DEKAN stage in more detail. The second and third stages are described in Section 2.2. DEKAN's training procedure is described in Algorithm 1.

B.0.1 Intra-Domain Data-Free Knowledge Extraction

In this stage, we extract the domain-specific knowledge from the available teacher models T_i separately by generating domain-specific synthetic datasets D_g^i. For this, we apply [83], an improved version of the data-free knowledge distillation method DeepInversion (DI) [78] that enables the generation of more diverse images. Hereby, we use inceptionism-style [46] image synthesis, also called DeepDream, i.e., we initialize random noise images $\hat{x}$ and optimize them to be recognized as a sample from a pre-defined class by a trained model. This process is also referred to as Inversion [13, 78]. Following [78, 83], uniformly sample labels y and optimize the corresponding random images $\hat{x}$ by minimizing the domain-specific inversion loss L_{DS} given by

$$L_{DS} = L_C(T(\hat{x}), y) + \lambda_1 L_R(\hat{x}) + \lambda_2 L_M(\hat{x}), \tag{4}$$

where L_C denotes the classification loss, e.g., cross-entropy, L_R an image prior regularization, L_M a feature moment matching loss, and λ_1 and λ_2 weighting coefficients. L_R penalizes the l_2-norm and the total variation of the image to ensure the convergence to valid natural images [42, 52, 46, 78]. L_M, also called moment matching loss [40], optimizes the synthetic images so that their feature distributions captured by batch normalization (BN) layers match those of the real data used to train the teacher model. Formally,

$$L_M(\hat{x}) = \sum_l max(\|\mu_l(\hat{x}) - \hat{\mu}_l\|_2 - \delta_l, 0) + \sum_l max(\|\sigma_l^2(\hat{x}) - \hat{\sigma_l}^2\|_2 - \gamma_l, 0). \tag{5}$$

L_M minimizes the l_2-norm between the BN-statistics of the synthetic data, i.e., mean $\mu_l(\hat{x})$ and variance $\sigma_l^2(\hat{x})$, and those stored in the trained teacher model, $\hat{\mu}_l$ and $\hat{\sigma_l}^2$, at each BN layer l [78]. In order to increase the diversity of the generated images, we relax this optimization by allowing the BN-statistics computed on the synthetic images to deviate from those stored in the model within certain margins, as introduced in [83]. These deviation margins are defined by relaxation constants for mean and variance, denoted by δ_l and γ_l respectively. The latter are computed as the ϵ_{DS} percentile of the distribution of differences between the stored BN-statistics and those computed using random images, as proposed in [83]. We note that the higher the value of the hyperparameter ϵ_{DS}, the higher the relaxation.

We apply this data-free inversion step to each domain-specific model T_i separately, yielding domain-specific synthetic datasets D_g^i that are correctly classified by their respective model and match the distribution of the features extracted by it.

B.0.2 Algorithm

Algorithm 1 summarizes the 3 stages of the DEKAN's training procedure. We note that the updates of the syntehtic data and the student model parameters θ are performed using gradient-based optimization, specifically Adam [27] in our case. Explicit update rule formulas and iteration over the synthetic data batches are omitted for simplicity of notation.

Algorithm 1 Domain Entanglement via Knowledge Amalgamation from domain-specific Networks

Require: $T_{1..I}$: I Domain-specific teacher models
// First stage: Intra-Domain Knowledge Extraction
for $i \leftarrow 1$ to I **do**
  Initialize the domain-specific synthetic dataset D_g^i: Images $\hat{x} \sim \mathcal{N}(0, I)$ and arbitrary labels
  while not converged **do**
    Update D_g^i by minimizing the domains-specific inversion loss L_{DS} (Eq. 4) using T_i
  end while
end for
// Second stage: Cross-Domain Knowledge Fusion
for $i \leftarrow 1$ to I **do**
  for $j \leftarrow 1$ to I and $i \neq j$ **do**
    Initialize the cross-domain synthetic dataset D_g^{ij}: Images $\hat{x} \sim \mathcal{N}(0, I)$ and arbitrary labels
    while not converged **do**
      Update D_g^{ij} by minimizing the cross-domain inversion loss L_{CD}^{ij} (Eq. 1) using T_i, T_j and D_g^j
    end while
  end for
end for
// Third stage: Multi-Domain Knowledge Distillation
Initialize the student model S_θ randomly or from a pre-trained model
Concatenate the domain-specific and cross-domain synthetic datasets into one dataset D_g
while not converged **do**
  Randomly sample a mini-batch $B = \{\hat{x}, y\}$ from D_g
  Update θ by minimizing the knowledge distillation loss L_{KD} (Eq. 3) using B and $T_{1..I}$
end while
return Domain-generalized student model S_θ

Chapter 6

Summary of Contributions

This thesis included four contributions that addressed four different learning problems, which involve data from multiple sources and are relevant for real-world applications. Despite tackling four different problems with different assumptions and characteristics, the four contributions addressed the same overarching question. We investigated how to train deep learning models on multiple datasets to best capture knowledge that can be reused when faced with related tasks/datasets. We developed approaches that leverage and optimize knowledge transfer from the training datasets to the ones encountered at evaluation time.

In our first contribution (Chapter 2), we addressed the underexplored intersection of the well-studied one-class classification (OCC) and few-shot learning problems. On the one hand, most of the Anomaly Detection (AD) approaches developed in prior work require large datasets of normal examples to generalize. However, such datasets are not available in data-scarce application scenarios. On the other hand, the few-shot classification literature focuses on class-balanced classification scenarios, where examples are available from all classes. Yet, due to the extreme rarity of anomalous behavior, e.g., defective products in industrial manufacturing or the diagnosis of a rare disease in healthcare, data examples from the anomalous class are usually not available, and OCC techniques are employed to perform AD. Hence, we addressed the few-shot one-class classification (FS-OCC) problem, a practical setting where an application-specific anomaly detector is needed and only a few normal examples are available for training. Our contribution is fourfold. Firstly, we have empirically shown that classical OCC methods fail in the low-data regime. Secondly, we theoretically analyzed why the parameter initializations optimized by gradient-based meta-learning algorithms, e.g., MAML, are not tailored for OCC, and why second-order

derivatives are necessary to optimize for such initializations. Thirdly, we proposed an episode sampling technique that adapts any meta-learning algorithm that employs a bi-level optimization to the FS-OCC problem. Finally, we demonstrated the effectiveness of the proposed approach on eight datasets of images and time-series, including an industrial sensor readings dataset. Future works could investigate an unsupervised approach to FS-OCC which does not require the meta-training tasks to be labeled.

Our second contribution (Chapter 3) addressed the unexplored intersection of the continual learning and the anomaly detection problems. Continual learning investigates ways of training models that are able to learn several tasks incrementally, i.e., reducing the impact of the catastrophic forgetting phenomenon. Such models are essential for real-world applications where the data distribution changes frequently, e.g., quality control in industrial manufacturing, where the product portfolio is constantly evolving. While the vast majority of continual learning works focus on class-balanced classification, many real-world applications exhibit a high class-imbalance due to the rarity of some categories. Anomaly detection problems are usually framed as one-class classification problems (OCC), where only data from the normal class is available. To the best of our knowledge, we were the first to address the Continual Anomaly Detection problem, which considers practical use-cases where a central anomaly detector for multiple applications is needed and new applications become available gradually over time. This contribution is threefold. Firstly, we introduced the novel and praxis-relevant continual anomaly detection problem and discussed its challenges: catastrophic forgetting and overfitting to the normal class. Secondly, we proposed an effective and model-agnostic meta-learning approach to address CAD. Our method learns a learning strategy tailored for learning anomaly detection task-sequences with minimal forgetting. Finally, we empirically validated our approach on three datasets, where we significantly outperformed previous class-balanced continual learning and anomaly detection methods. While our experiments focused on tasks containing in-domain anomalies, it would be interesting to investigate the suitability of our approach to out-of-distribution detection applications where the model is evaluated on anomalies that belong to unseen domains.

In our third contribution (Chapter 4), we tackled the domain generalization problem. In real-world applications, distribution shifts between training and test data are commonly encountered. For instance, data distributions might differ from one hospital to another and from one production plant to another due to using different scanners and machines. Domain generalization works study learning scenarios, which involve different datasets

exhibiting domain shift, and where the target domain is unknown beforehand, e.g., the data collected by a scanner that will be acquired in the future. In this contribution, we proposed a domain generalization (DG) approach that incentivizes the model to capture as many features as possible. This is based on the assumption that a richer set of features improves the knowledge transfer to a wider variety of unseen domains. Our algorithm leverages methods from the explainable machine learning literature to identify the features captured by the model. Thereafter, these learned features are corrupted and the model is trained on the corrupted version of the data, hence enforcing new feature discovery. We evaluated our method on a DG testbed that fairly compares DG algorithms by including the same pre-processing pipeline and hyperparameter search. We found that our algorithm outperforms 18 DG approaches on three different DG benchmark datasets. Furthermore, our results have shown that the richer set of learned features also improves in-domain generalization, suggesting the suitability of our approach for applications beyond domain generalization to include scenarios without domain shift. In future works, it would be interesting to explore an unsupervised variant of our approach where the model is trained to reconstruct the corrupted features instead of predicting the correct class of the corrupted datapoint.

Our fourth and last contribution (Chapter 5) addressed the novel data-free domain generalization problem. While machine learning methods require data to learn, in many real-world scenarios, data access is not possible, e.g., due to data privacy, security, or safety concerns, or to avoid commercial disadvantage and/or reverse engineering. As a response, Data-Free Knowledge Distillation (DFKD) methods were proposed to tackle the scenario where the data owners are willing to share a model trained on their data instead of releasing the original dataset. However, most DFKD methods address domain-specific scenarios, while many real-world applications exhibit domain shift between training and test data. In this work, we addressed the unexplored intersection of domain generalization and data-free learning, which we defined as the Data-Free Domain Generalization (DFDG) problem. DFDG investigates the practical setting where a model that is robust to domain shift is needed and only models trained on the source domains are available. Moreover, we proposed Domain Entanglement via Knowledge Amalgamation from domain-specific Networks (DEKAN), an effective approach for this problem, as well as several baseline methods. Our algorithm extracts and merges the knowledge contained in the available domain-specific teacher model by generating domain-specific and cross-domain synthetic examples. The latter are optimized by maximizing the agreement of different domain-specific teachers

and minimizing a cross-domain feature distribution matching loss. The generated images are then used to transfer the knowledge to a student model via multi-teacher knowledge distillation. The student is tested on the target domain without any modification or prior exposure to their data. Finally, we evaluated DEKAN on two DG benchmark datasets and found that it achieved new state-of-the-art results on this challenging problem and reduced the gap between the best DFDG baseline and the upper-bound oracle method that uses the private source domain data. In future works, it would be interesting to study DFDG from a differential privacy perspective, e.g., by applying DEKAN to teacher models trained with differential privacy methods, which might prevent private information leakage to the generated synthetic data.

Bibliography

Charu C Aggarwal. Outlier analysis. In *Data mining*, pages 237–263. Springer, 2015.

Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *CVPR*, 2021.

Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *arXiv:1911.00804*, 2019.

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3375, 2017.

Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2:1–18, 2015.

Jerone Andrews, Thomas Tanay, Edward J Morton, and Lewis D Griffin. Transfer representation-learning for anomaly detection. JMLR, 2016.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.

Sercan O Arık and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *AAAI*, volume 35, pages 6679–6687, 2021.

Mihalj Bakator and Dragica Radosav. Deep learning and medical diagnosis: A review of literature. *Multimodal Technologies and Interaction*, 2(3):47, 2018.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *NeurIPS*, 2018.

Shawn Beaulieu, Lapo Frati, Thomas Miconi, Joel Lehman, Kenneth O Stanley, Jeff Clune, and Nick Cheney. Learning to continually learn. *arXiv preprint arXiv:2002.09571*, 2020.

Sarah Bechtle, Artem Molchanov, Yevgen Chebotar, Edward Grefenstette, Ludovic Righetti, Gaurav Sukhatme, and Franziska Meier. Meta learning via learned loss. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4161–4168. IEEE, 2021.

Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *NeurIPS*, 2011.

Francesco Cappio Borlino, Antonio D'Innocente, and Tatiana Tommasi. Rethinking domain generalization baselines. In *International Conference on Pattern Recognition (ICPR)*, 2021.

Junyi Chai and Anming Li. Deep learning in natural language processing: A state-of-the-art survey. In *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 1–6. IEEE, 2019.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

Arslan Chaudhry, Albert Gordo, Puneet K Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. *arXiv preprint arXiv:2002.08165*, 3, 2020.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *ICCV*, 2019.

Haokun Chen, Ahmed Frikha, Denis Krompass, and Volker Tresp. Fraug: Tackling federated learning with non-iid features via representation augmentation. *arXiv preprint arXiv:2205.14900*, 2022.

Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 90–98. SIAM, 2017.

Gabriella Contardo, Ludovic Denoyer, and Thierry Artières. A meta-learning approach to one-step active learning. *arXiv preprint arXiv:1706.08334*, 2017.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Zhun Deng, Frances Ding, Cynthia Dwork, Rachel Hong, Giovanni Parmigiani, Prasad Patil, and Pragya Sur. Representation via representations: Domain generalization via adversarially learned invariant representations. *arXiv:2006.11478*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *NeurIPS*, 2019.

Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 1999.

Ahmed Frikha, Haokun Chen, Denis Krompaß, Thomas Runkler, and Volker Tresp. Towards data-free domain generalization. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021a.

Ahmed Frikha, Denis Krompaß, Hans-Georg Köpken, and Volker Tresp. Few-shot one-class classification via meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7448–7456, 2021b.

Ahmed Frikha, Denis Krompaß, and Volker Tresp. Columbus: Automated discovery of new multi-level features for domain generalization via knowledge corruption. *arXiv:2109.04320*, 2021c.

Ahmed Frikha, Denis Krompaß, and Volker Tresp. Arcade: A rapid continual anomaly detector. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10449–10456, 2021d. doi: 10.1109/ICPR48806.2021.9412627.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

Pedro Garcia-Teodoro, Jesus Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1-2):18–28, 2009.

Robert Geirhos, Carlos R Medina Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. *arXiv:1808.08750*, 2018.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4367–4375, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014a.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv:1412.6572*, 2014b.

Benjamin Frederick Goodrich. Neuron clustering for mitigating catastrophic forgetting in supervised and reinforcement learning. *PhD diss., University of Tennessee, 2015*, 2015.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *JMLR*, 2012.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv:2007.01434*, 2020.

Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016.

Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *European conference on computer vision*, pages 124–141. Springer, 2020.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. *Advances in neural information processing systems*, 31, 2018.

Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 170–180. Springer, 2002.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer, 2001.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning, 2018.

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference*, 2020.

Mike Huisman, Jan N Van Rijn, and Aske Plaat. A survey of deep meta-learning. *Artificial Intelligence Review*, 54(6):4483–4541, 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Khurram Javed and Martha White. Meta-learning representations for continual learning. In *Advances in Neural Information Processing Systems*, 2019.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Shehroz S Khan and Michael G Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014.

Siavash Khodadadeh, Ladislau Boloni, and Mubarak Shah. Unsupervised meta-learning for few-shot image classification. *Advances in neural information processing systems*, 32, 2019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

Gregory R. Koch. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *CVPR*, 2020.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019.

Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in neural information processing systems*, 2017.

Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. *arXiv preprint arXiv:1801.05558*, 2018.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *IEEE ICCV*, 2017a.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence*, 2018a.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Sequential learning for domain generalization. In *European Conference on Computer Vision*, pages 603–619. Springer, 2020a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *IEEE CVPR*, 2018b.

Ke Li and Jitendra Malik. Learning to optimize neural nets. *arXiv preprint arXiv:1703.00441*, 2017.

Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020b.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021a.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, 2018c.

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv:1603.04779*, 2016.

Yuhang Li, Feng Zhu, Ruihao Gong, Mingzhu Shen, Xin Dong, Fengwei Yu, Shaoqing Lu, and Shi Gu. Mixmix: All you need for data-free compression are feature and data mixing. In *ICCV*, 2021b.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017b.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1013–1023, 2021a.

Yuang Liu, Wei Zhang, Jun Wang, and Jianyong Wang. Data-free knowledge transfer: A survey. *arXiv preprint arXiv:2112.15278*, 2021b.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*, 2017.

Ali Madani, Ramy Arnaout, Mohammad Mofrad, and Rima Arnaout. Fast and accurate view classification of echocardiograms using deep learning. *NPJ digital medicine*, 1(1): 1–8, 2018.

Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. *arXiv:2006.07500*, 2020.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

Fabio Maria Carlucci, Paolo Russo, Tatiana Tommasi, and Barbara Caputo. Hallucinating agnostic images to generalize across domains. In *IEEE/CVF ICCV Workshops*, 2019.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

Paul Micaelli and Amos Storkey. Zero-shot knowledge transfer via adversarial belief matching. *arXiv:1905.09768*, 2019.

Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *IEEE ICCV*, 2017.

Mary M Moya, Mark W Koch, and Larry D Hostetler. One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N*, 93, 1993.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

Vahid Nasir and Farrokh Sassani. A review on deep learning in machining and tool monitoring: methods, opportunities, and challenges. *The International Journal of Advanced Manufacturing Technology*, 115(9):2683–2709, 2021.

Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *ICML*, 2019.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 721–731, 2018.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. *Applied Soft Computing*, 93:106384, 2020.

Kunkun Pang, Mingzhi Dong, Yang Wu, and Timothy Hospedales. Meta-learning transferable active learning policies by deep reinforcement learning. *arXiv preprint arXiv:1806.04798*, 2018.

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv:2009.00329*, 2020.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Marcel Prastawa, Elizabeth Bullitt, Sean Ho, and Guido Gerig. A brain tumor segmentation framework based on outlier detection. *Medical image analysis*, 8(3):275–283, 2004.

Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5822–5830, 2018.

Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *IEEE/CVF CVPR*, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. Multi-component image translation for deep domain generalization. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.

Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. Correlation-aware adversarial domain adaptation and generalization. *Pattern Recognition*, 2020.

S Benson Edwin Raj and A Annie Portia. Analysis on credit card fraud detection methods. In *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, pages 152–156. IEEE, 2011.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR, 2019.

Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1577–1581. IEEE, 2017.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representations (ICLR) 2017*, 2016.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

Yi Ren, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Almost unsupervised text to speech and automatic speech recognition. In *International Conference on Machine Learning*, pages 5410–5419. PMLR, 2019.

James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1907–1916, 2021.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pages 4393–4402, 2018.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018a.

Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization, 2018b.

Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks

to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017.

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

Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *Advances in neural information processing systems*, pages 582–588, 2000.

Luke Scime and Jack Beuth. Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. *Additive Manufacturing*, 19:114–126, 2018.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

Soroosh Shahtalebi, Jean-Christophe Gagnon-Audet, Touraj Laleh, Mojtaba Faramarzi, Kartik Ahuja, and Irina Rish. Sand-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization. *arXiv:2106.02266*, 2021.

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv:1804.10745*, 2018.

Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *IEEE/CVF CVPR*, 2019.

Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv:2104.09937*, 2021.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv:1710.10571*, 2017.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

Nathan Somavarapu, Chih-Yao Ma, and Zsolt Kira. Frustratingly simple domain generalization via image stylization. *arXiv:2006.11207*, 2020.

Giacomo Spigler. Meta-learnt priors slow down catastrophic forgetting in neural networks. *arXiv preprint arXiv:1909.04170*, 2019.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI Global, 2010.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv:1412.3474*, 2014.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning, 2016.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *arXiv:1805.12018*, 2018.

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Multimodal model-agnostic meta-learning via task-aware modulation. *Advances in Neural Information Processing Systems*, 32, 2019.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 2018.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Yaqing Wang and Quanming Yao. Few-shot learning: A survey. *arXiv preprint arXiv:1904.05046*, 2019.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020a.

Yufei Wang, Haoliang Li, and Alex C Kot. Heterogeneous domain generalization via domain mixup. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020b.

Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2020.

Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection. *Procedings of the British Machine Vision Conference 2015*, 2015. doi: 10.5244/c.29.8. URL `http://dx.doi.org/10.5244/C.29.8`.

Ju Xu and Zhanxing Zhu. Reinforced continual learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *AAAI Conference on Artificial Intelligence*, 2020.

Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv:2001.00677*, 2020.

Chengxiang Yin, Jian Tang, Zhiyuan Xu, and Yanzhi Wang. Adversarial meta-learning. *arXiv preprint arXiv:1806.03316*, 2018.

Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *CVPR*, 2020.

Chris Yoon, Ghassan Hamarneh, and Rafeef Garbi. Generalizable feature learning in the presence of data bias and domain class imbalance with application to skin lesion classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2019.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *Proceedings of machine learning research*, 2017.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv:1710.09412*, 2017.

Mengmi Zhang, Tao Wang, Joo Hwee Lim, Gabriel Kreiman, and Jiashi Feng. Variational prototype replays for continual learning. *arXiv preprint arXiv:1905.09447*, 2019.

Xiangguo Zhang, Haotong Qin, Yifu Ding, Ruihao Gong, Qinghua Yan, Renshuai Tao, Yuhang Li, Fengwei Yu, and Xianglong Liu. Diversifying sample generation for accurate data-free quantization. In *CVPR*, 2021.

Zijie Zhang, Zeru Zhang, Yang Zhou, Yelong Shen, Ruoming Jin, and Dejing Dou. Adversarial attacks on deep graph matching. *Advances in Neural Information Processing Systems*, 33:20834–20851, 2020.

Andrey Zhmoginov, Mark Sandler, and Max Vladymyrov. Hypertransformer: Model generation for supervised and semi-supervised few-shot learning. *arXiv preprint arXiv:2201.04182*, 2022.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *AAAI Conference on Artificial Intelligence*, 2020a.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020b.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *arXiv:2103.02503*, 2021a.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv:2104.02008*, 2021b.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.