Mitigating the Challenges of Distribution Shift under Strong Computational Constraints

DPhil Thesis
Supervised by Dr. Adel Bibi and Prof. Philip H. S. Torr

Botos Csaba
Wolfson College
University of Oxford

Torr Vision Group
Department of Engineering Science
University of Oxford

February 2024

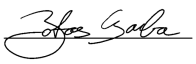
I dedicate this work to my greatest inspiration, Ficsu.

Declaration

I hereby declare that this thesis (and the work presented in it) is submitted to the Department of Engineering Science, University of Oxford, in fulfillment of the requirements for the degree of Doctor of Philosophy, and that this thesis is entirely my own work based on publications in collaboration with the co-authors where due acknowledgement is made.

April 2024

Botos Csaba



Acknowledgement

There is an anecdote how each cell in your body gets replaced by a new one roughly once in seven years. Well, now it seems every aspect of your personality gets replaced after every new chapter in your thesis. I would barely remember who I was and what we were all meant to do here if it wasn't for both the loving folks and the humbling adversaries, to whom I would like to dedicate this undeservingly short section.

First, I would like to thank my father, the original Botos Csaba, who endowed me with the skill of aggressively overestimating my talent by providing me with bottomless, unconditional love and support in every avenue of my interests. Genuinely, it must have been a moonshot to buy a kid in elementary school books about optimal transport, chaos theory and connectivism, but without hubris I would have never even dared to dream about the problems I work on, now on a daily basis. While shooting for the stars is all fun and exciting, one can seriously burn themselves in the process, and I wish I could express my sense of safety that I gained from my mother, Palkovics Borbála's loving trust. Her persistence and stubbornness in finding peace guided and inspired me to sail through the perpetual storm in the PhD teacup and it is but her words of kindness that I speak to people before they open their arms for a hug.

Being the son of the human equivalents of the Sun and the Moon left me with a genuinely difficult optimization problem: how far do I want to go and when do I call it a day? I got the answer in a heart-shaped box from the most brilliant engineer, an engineer of the human body, Dr. Zomborszki Emese. Without her support, I would not have had even the slightest chance of grasping where we came from, where we were *supposed to be* and where we *wanted to be*. Fate eyed us like a Pisces, things turned tar pit black, but here we are now: I have no complaint and am forever in debt to your priceless advice.

The pandemic. It happened. It happened, whether for the worse or the better is open to interpretation. Risking some controversy, I will admit, it came at the right time. It taught me the truly important values: relations and support. During my time, back in

Hungary I have been deeply humbled by the care of my friends and my family. When dark clouds appeared, Kiss-Baraksó Livi was always the first to spot and make me see the silver linings, and I am eternally in debt for her endless support in the hardest days. My closest relative, the untethered genius mind, currently inventing his own language, Gabi – thank you for all the walks and talks, the jams and gigs, and for having an open ear for the words that are hard to speak. My college room-mate, Hakkel Tomi, with whom we shared the same roof once again – Viktor, Marci, Csanna, Bálint, Dani thanks for cheering me on for rebuilding the wall after I tore out the chandelier. You are the best people to dust off the shoulders, even when life turns upside down. Thanks Zsiga, Berci and Ödön for the countless hours spent in the acrobatics center practicing sommersaults and literally being upside down.

After moving back to the *perfidious* Albion, on the verge of absurdity, I had to realize that I was surrounded by all the spotless minds, yet again. I would like to thank my supervisor, Philip Torr, for keeping me going, against all odds, against all events. Brilliant in the quality of a leader, pure in the quality of a human being: I feel grateful for the discussions about the big questions in life with Phil while feeding the ducks on the bank of Isis. I would also like to express how lucky I feel for the silver lining that came in the shape of a dear friend, Adel Bibi. He reviewed and endured many of my musings as a beginner researcher, called me out when I downplayed the earth shattering, seismic, seminal findings that the reader is going to learn about in the next few chapters, and muffled the enthusiasm when I was advertising our work like ”a bazar salesman”. Providing guidance and advice in every small and large steps along the road, I have met the best mentor one could wish for. Thank you, Adel!

I would like to express my gratitude towards all the great people I have had the privilege to exchange ideas and share meals with at Wolfson College. Madhu, Sam, Freddy and Adam always provide the most serious debates about the least serious topics. Tejo and Daattavya, the kindest of all the people I have met here in Oxford, have pushed me

through the countless challenges over the years. I am thankful towards the members of the Wolfson band, and for all the balls and concerts we performed at together. Through these musical experiences I have met brilliant talents such as Alessandro, Andreas, Toby, Eugene and many others. I would like to thank for all the trips around Europe and culinary experiences with Lizzie, Joost, Dim, Christine, Riz, Michael, Marija, Piotr and Tugrul. Finally, I would like to thank everyone in the Wolfson College Boatclub and the wonderful experiences we shared across races, from the Thames to Yangtze. Thank you for the hard volunteering work Yixieng, David, Mantas, Ben, Thomas and the excellent coaching, Rory, Phil and Philippa.

If one goes to where the magic happens, the office of Torr Vision Group, one will find a loving crowd of the finest thinkers, always having a banter. I would like to say thanks for the fun discussions we had over lunch breaks with Christian, Chen Lin, Runjia, Fabio, Ashkan and Alasdair, Francesco, Aleks, Jindong, Guangran, Shuyang, Ameya, Viveka, Harkirat, Jishnu, Jimmy, Namhoon, Anurag, Arnab, Nantas and Thalaiyasingam. The countless walks and talks with the kindest gentleman of all, Francisco really have a special place in my memories. You introduced me to my twin from a different mother, the legendary Alister Burt, for which I can hardly repay you in this lifetime. I want to thank Stuart Golodetz, Adnane Boukhayma, Xiaojuan Qi, Puneet Dokhania and Arslan Chaudhry for their helpful advice and exciting discussions. All of this would not have been possible without the kind reply of Jack Valmadre, the first person who replied from the group and referred me to Phil to review my application.

I am grateful for the confidence and enthusiasm my colleagues at Verizon gave me: without the support of Horváth András, Horváth István and Rekeczky Csaba, I would have not had the chance to become the researcher I am today. Furthermore, I would like to thank Paszternák András, for providing me with excellent opportunities to connect my research to larger communities through the sciene-education forums under his management: Csodák Palotája and Lauder Javne Iskola. I would like to mention how lucky I feel to have met

Matthias Müller, Motasem Alfarra, Alejandro Pardo, Shariq Bhat, Adam Loch and Diana Wofk at Intel Labs during my internship. It has been a true honor to strengthen our research group's connection with the King Abdullah University of Science and Technology. Their delegate of top notch researchers, such as Wenxuan, Yasir and Hasan brought some of the Saudi sunshine in their heart and really brightened up the days during their stay.

Here in Oxford, I have the true privilege to be in the epicentre of not only the best Machine Learning research but the biggest utopian cultural melting pot. Thanks to this sacred shrine of civilization, I have learned some of the worst German puns, the most affectionate Arab insults, ate the strangest American sweet potato sushis, studied the true teachings of 涛哥, shook hands with mayors of cities larger than my Country, and had a laugh with rabbies, imams and a Bishop. I have not only met people here, I have met nations. It is debatable whether the real value of my time being here is truly expressed in technological contributions in the lengthy chapters below, or in this few lines, condensed from all the friendship and good memories I have been gifted by the people over this course.

Thanks Závaczki Dani and Zakor Bia for welcoming me into your family. I just can't wait to show this work to my cheeky godson Koni, hoping that by the time he will have enough teeth to chew through this thesis, but enough literature review experience to refrain from doing so.

I wish I had the right words to explain how I am left in perpetual awe by my muse, the one and only Celia G. B. Hanna, a humble rising star, an excellent researcher of the kingdom of Neptune and the most stunning descendant of Queen Cleopatra herself. The weight on my shoulder, if there is any, feels like feathers on my wings when she sings and if at times, life feels like running from a rolling a boulder her smile makes me want to run another mile.

This work is dedicated for my dear family member, Ficsu, whose memory lives on forever in my heart. Enjoy the endless sky, see you in a few years!

Contents

List of Tables

Abstract

This thesis explores the use of unsupervised methods to address the challenges of distribution shift under strong computational constraints in deep learning vision tasks. Through a series of contributions, it demonstrates the effectiveness of unsupervised learning in enabling efficient and adaptable models suitable for resource-constrained environments. The Diversified Dynamic Routing (DivDR) method showcases the potential of unsupervised clustering for efficient image recognition, while the Domain Partitioning Network (DoPaNet) tackles mode collapse in generative adversarial learning. A systematic unification of pixel-level alignment, feature-level alignment, and pseudo-labeling is proposed for cross-domain object detection, highlighting the power of unsupervised adaptation techniques. The thesis also extends its investigation to online continual learning with label delay, proposing a novel approach to address the challenges of evolving data distributions and delayed feedback. The findings have significant implications beyond vision tasks, with the potential to impact a wide range of real-world applications. As machine learning continues to evolve and tackle increasingly complex challenges, the importance of unsupervised learning in enabling robust, efficient, and adaptable models will only grow. This thesis represents a significant step forward in the development of unsupervised methods, paving the way for future machine learning applications.

Introduction

> *"... As each generation developed and new entities were born and in turn reproduced, so complexity increased. Those old primordial and elemental principles were spun into life-forms of ever greater diversity, variety and richness. The beings that were born became endowed with nuanced and unique personalities and individuality. In computer language, it was as if life went from 2 bit to 4 bit to 8 bit to 16 bit to 32 bit to 64 bit and beyond. Each iteration represented millions and then billions of new permutations of size, form and what you might call resolution. [...] Creatures and gods that were ambiguous, inconsistent, unpredictable, intriguing and unknowable had arrived. [...] The fun began."*

— *Stephen Fry, Mythos, 2017*

The fun began

In the field of Machine Learning, we find ourselves at a juncture where the rapid growth of computational power, data availability, and algorithmic sophistication has given rise to a new generation of models capable of tackling increasingly complex and diverse tasks. Just like the detailed character traits and complex personalities emerging from subsequent generations of deities in the Greco-Roman theogony – brilliantly put by Stephen Fry – the field of machine learning has witnessed a progression from simple, constrained experimental

setups and specific models with narrow scope to more accurate representations of real-world problems, leading to models with general problem solving capabilities [151], able to navigate the ambiguities and inconsistencies of real-world data.

This thesis explores the challenges that arise in the most common Machine Learning applications, such as computational constraints, varying task difficulties, load balancing, delayed feedback and on-the-fly adaptation to new data. In our core contributions, we demonstrate how the common assumptions of data homogeneity and i.i.d. sampling are broken and how they impact the model performance. We find clear evidence, that without addressing the consequences of breaking these assumptions, the resulting models suffer severe degradation in performance. In mission-critical applications, such as healthcare, privacy and finance, it is hard to overestimate the importance of resolving such issues to avoid hurting the global progress of data-driven decision making. In this work, through a sequence of technical chapters that gradually increase in experimental complexity, we simulate more realistic machine learning application scenarios, to propose new unsupervised learning solutions that can enable efficient and adaptable models, suitable for deployment on consumer-grade hardware.

The central premise of this work is that unsupervised learning holds the key to unlocking the vast potential of deep learning in real-world settings, where the availability of labeled data is often limited and the distribution of data can change, or *shift* over time. By leveraging the abundance of unlabeled data and developing efficient unsupervised methods, we aim to create models that can learn meaningful representations, adapt to new environments, and generalize to unseen tasks.

1.1 Motivation

Deep learning has emerged as a transformative force across all scientific fields, revolutionizing the way we approach complex problems and extract insights from vast amounts of

data. From healthcare and finance to environmental studies and beyond, deep learning models have demonstrated remarkable capabilities in pattern recognition, prediction, and decision-making, enabling breakthroughs that were once thought impossible.

As deep learning finds its way into critical applications that directly impact our lives on all scales, ensuring high reliability becomes paramount. Whether it's diagnosing diseases, detecting financial fraud, or predicting natural disasters, the consequences of model failures can be severe and far-reaching. It is crucial that deep learning models consistently deliver accurate and trustworthy results, even in the face of evolving data landscapes and real-world challenges.

One of the most significant hurdles in deploying deep learning models in real-world applications is the problem of *distribution shift*. In practical settings, the data encountered during deployment often differs from the data used to train the model. This distribution shift can arise due to various factors, such as changes in data collection methods, temporal variations, or domain-specific nuances. The biggest concern regarding such changes in the data distribution is that it alters the behavior of the underlying model in unpredictable and often inexplicable ways. For example, when patient data arrives from a new hospital, it will likely have different statistics due to differences both in the medical recording devices and the population diversity of the new patients. Such difference in the input data *may* or *may not* impact the model that has already been deployed in our medical assistant product and without checking there is hardly any way to tell. Assuring that the models are prepared to handle such challenges is essential to ensure that deep learning models remain effective and reliable when applied to real-world tasks, across diverse domains and over extended periods.

Medical Sciences. In healthcare, deep learning models have been applied to medical image analysis, enabling early detection of diseases and assisting in clinical decision-making. For instance, convolutional neural networks have been used to detect skin cancer

from dermatoscopic images with accuracy rivaling that of trained dermatologists [212]. While with expert supervision such models can become helpful for aiding the diagnosis, thanks to the advent of large scale representation learning techniques, previously infeasible applications, such as reconstructing visual stimuli from fMRI scans in real time [16] and decoding speech from magneto encephalography (MEG) signals [48], have become possible. However, these applications often face challenges related to distribution shifts, as the data collected from different patients, devices, or environments may not always match the distribution of the training data. Due to the scarcity of data, a common practice among medical applications is unsupervised pre-training on the concatenated, publicly available datasets [189]. One factor that further complicates the medical applications is the delayed feedback, since some of the ground true labels for predictive tasks, such as the true causing factor behind the symptoms or the recovery time is inherently subject to the delay of the biological processes. While the labels for new patient data may be delayed, the recordings are immediately available, which opens a window of opportunity to use unsupervised adaptation techniques to improve the performance.

Security and Finance. In the field of finance, machine learning algorithms have found applications in fraud detection [12], risk assessment [125], and algorithmic trading [133]. Deep learning models have been used to detect credit card fraud in real-time, leveraging the ability of these models to learn complex patterns and anomalies from vast amounts of transactional data [148]. Machine learning has also been applied to predict stock market movements and optimize portfolio management. However, financial data is known to show strong distribution shifts, where the underlying patterns and relationships change over time due to evolving market conditions, consumer behavior, or economic factors [221]. Similarly to the delays due to the natural processes in healthcare applications, the lagging feedback is also present both in security and finance: a security breach remains unknown before the pre-existing triggering mechanisms go off [233], so does the profitability of an

investment into a startup before any performance metric shows an uptick [191]. Nevertheless, accessing the input data that shows even the slightest early signs of distribution shifts can significantly boost the predictive systems.

Environmental Studies and Sustainability. Environmental studies have also benefited greatly from the adoption of machine learning techniques. Deep learning models have been used for climate modeling, enabling more accurate predictions of future climate patterns and the potential impacts of climate change [122]. Machine learning has also been applied to ecosystem monitoring, such as using satellite imagery to track deforestation and habitat loss [225, 152]. In the domain of renewable energy, machine learning algorithms have been employed for wind and solar power forecasting, optimizing the integration of these intermittent energy sources into the power grid [235]. However, environmental data is often subject to distribution shifts due to seasonal variations, climate change, or land use modifications [150]. Unsupervised learning can help capture the underlying structure and dynamics of environmental systems from large amounts of unlabeled data, enabling models to generalize better across different conditions and adapt to changing distributions over time.

To address the challenge of distribution shift and improve the reliability of deep learning models in real-world applications, we can leverage the vast amounts of unsupervised data available in various domains. By developing unsupervised learning techniques that can extract meaningful representations and capture the underlying structure of the data, we can enable models to adapt to new distributions and maintain high performance even in the face of evolving data landscapes. However, it is important to recognize that there is no one-size-fits-all solution to this problem. The choice of the most suitable unsupervised learning paradigm depends on the specific characteristics of the environment and the variables at play. Factors such as the nature of the data, the extent of the distribution shift, the availability of computational resources, and the desired level of adaptability all

influence the selection of the most appropriate approach. By carefully considering these factors and tailoring the unsupervised learning techniques to the specific requirements of each application, we can effectively harness the power of unsupervised data to improve the robustness and reliability of deep learning models in real-world settings.

1.2 Scope and Approach

While every modality has its own challenges and merits, the paradigms that we are interested in this thesis are not limited to any specific domain. In this thesis, we chose image recognition problems as the main modality, simply because at the time of writing it provides the largest number of publicly accessible datasets with the highest number of different down-stream tasks. The abundance of visual data provides a unique opportunity to explore unsupervised learning methods at various scales. Furthermore, for educational purposes, vision tasks are, by far, the most responsive and interactive ways to develop good intuition about the changes in model behavior on unexpected inputs.

In fact, it is almost misleading how easy it is to generate a few test cases just using our web-camera that represents one or another type of distribution shift: it takes little to no time to illustrate the failure mode, however, to eliminate all confounding factors, one needs to take intro consideration an enormously large array of seemingly irrelevant environmental variables. Such variables can be the order of the datapoints for training, the device to capture the data, the diversity of the underlying subjects and the list goes on. For purely pragmatic reasons, main stream research often oversimplifies real-world problems to reduce environmental variables and improve reproducibility. However, over time, the purpose of this simplification fades, and the attention of the research community shifts towards achieving higher scores and better performance on established, and often outdated, benchmarks brilliantly highlighted by Prabhu *et al.* [171, 169] and Ghunaim *et al.* [69].

The approach taken in this thesis is to explore the use of unsupervised methods to bridge distribution discrepancies in deep learning vision tasks. By leveraging unsupervised learning, we aim to develop models that can learn meaningful representations from the vast amounts of unlabeled visual data available, reducing the reliance on costly and time-consuming manual annotations. The unsupervised methods explored in this thesis are designed to be efficient and suitable for deployment on consumer-grade hardware, democratizing machine learning and enabling its application in a wide range of real-world scenarios.

1.3 Why Unsupervised Methods are Essential

The success of deep learning in vision tasks has been largely driven by the availability of large-scale labeled datasets, such as ImageNet [49]. However, the reliance on labeled data is a significant limitation for the widespread adoption of deep learning in real-world applications. Obtaining large-scale labeled datasets for every possible vision task and domain is simply infeasible, as it requires a tremendous amount of human effort and expertise [167, 195]. In contrast, unsupervised learning methods offer a promising alternative by learning from unlabeled data, which is abundant and readily available.

Pre-training. One key paradigm is pre-training, which involves training a model on a large corpus of unlabeled data before fine-tuning it on a smaller labeled dataset for a specific task. Pre-training methods such as self-supervised learning and contrastive learning have shown promising results in learning useful visual representations without explicit labels. Unsupervised learning has already shown remarkable success in scaling large language models [174, 110, 204], vision-language models [139, 147], diffusion models [181, 94]. One of the biggest advantages of such pre-trained models is that they can be further trained, or *fine-tuned* to a specific use case with far less data and parameter updates that achieves higher accuracy and robustness to noise than a counterpart that is

solely trained on data collected from the specific-domain [243].

Model Sharing & Merging. At the time of writing, the leading trend in the ML community is to build on such pre-trained *foundational* models, which are predominantly contributions from non-academic research institutions. Depending on the terms under which the foundational model is licensed, the community either shares the fine-tuned weights on freely accessible hubs, such as the HuggingFace Hub [222], or publishes the model under a new license. More recently, it has been shown that without any further parameter updates, foundation models can be *merged*, i.e.,the hidden layers can be connected to outperform the individual foundation models on both generic and domain-specific tasks [1, 223].

Model Adaptation. Unsupervised learning also plays a crucial role in adaptation tasks, where the goal is to transfer knowledge from a source domain to a target domain with different data distributions. Unsupervised domain adaptation (UDA) techniques aim to align the feature distributions of the source and target domains without requiring labeled data in the target domain. Test-time adaptation (TTA) methods, on the other hand, adapt the model to the test data distribution during inference, improving the model's generalization ability.

Long Standing Challenges. However, applying unsupervised learning to vision tasks poses several challenges. Visual data is high-dimensional and unstructured, making it difficult to learn meaningful representations without explicit supervision [13]. Moreover, the evaluation of unsupervised learning methods is often more challenging than supervised learning, as there is no clear metric for measuring the quality of the learned representations [165]. Empirical evidence suggests [13, 45, 199] that the most reliable way to guarantee performance gains using unsupervised techniques is simply increasing the size of the training data by orders of magnitude and conducting extensive (and expensive) hyper-parameter search for each specific-application [96].

To address these challenges, this thesis focuses on developing efficient and effective unsupervised methods that can learn meaningful representations from unlabeled visual data. We explore various unsupervised learning techniques, such as generative adversarial networks [80], domain adversarial feature matching [63] and self-supervised learning [13], and propose novel methods that combine the strengths of these approaches. Moreover, we emphasize the importance of efficiency in unsupervised learning methods, as the widespread adoption of these methods in real-world applications requires models that can run efficiently on resource-constrained devices [69]. We propose lightweight architectures, and efficient mixture of experts methods that can reduce the computational and memory requirements of machine learning models, making them suitable for deployment on consumer-grade hardware.

1.4 Chapter Overviews

The central theme of this thesis, using unsupervised methods to bridge distribution discrepancies in deep learning vision tasks, is addressed through a sequence of contributions that gradually increase in experimental complexity. Each chapter introduces a novel unsupervised method and evaluates its effectiveness on a specific vision task, highlighting the benefits and limitations of the proposed approach.

Chapter 2 investigates the use of unsupervised clustering for efficient image recognition. The proposed method, Diversified Dynamic Routing (DivDR), trains several local experts on learnt subsets of the training dataset, addressing the challenge of relying on a universal representation for all samples. DivDR automates the subset assignment process, removing human priors and finding useful partitions that lead to more efficient models. Through extensive evaluations and comparisons on semantic segmentation, object detection, and instance segmentation tasks, DivDR demonstrates a better trade-off between accuracy and efficiency compared to existing methods.

Chapter 3 explores the use of unsupervised clustering for image generation. The proposed Domain Partitioning Network (DoPaNet) addresses the mode collapse problem in generative adversarial learning by employing multiple discriminators, each encouraging the generator to cover a different part of the target distribution. DoPaNet introduces a classifier to ensure that the learned modes do not overlap, enabling control over the generated modes and aligning with the thesis's goal of developing more adaptable models. Experiments on toy examples and real images demonstrate DoPaNet's superiority in covering the real distribution compared to competing methods.

Chapter 4 tackles the problem of cross-domain object detection, where the distribution of test data differs from that of the training data. The proposed unsupervised domain adaptation technique systematically unifies pixel-level alignment, feature-level alignment, and pseudo-labeling into a single training procedure. By addressing the challenge of distribution shifts between training and test data, this chapter demonstrates the effectiveness of unsupervised methods in adapting models to new environments and improving their generalization capabilities.

Chapter 5 addresses the challenge of online continual learning with label delay, a scenario in which both the training and test data distributions evolve over time, and labels become available only after a certain delay. This chapter builds upon the insights gained from the Torr Vision Group's prior research on continual learning, extending the experimental framework to better reflect real-world challenges and proposing a novel approach that is well-suited to the new benchmarks.

1.5 Manuscripts

The chapters described in the previous section are based on the following manuscripts:

Chapter 2: Botos Csaba, Adel Bibi, Yanwei Li, Philip Torr, Ser-Nam Lim. "Diversified Dynamic Routing for Vision Tasks". *European Conference on Computer Vision (ECCV)*

3rd Visual Inductive Priors for Data-Efficient Deep Learning Workshop, 2022.

Chapter 3: Botos Csaba, Adnane Boukhayma, Viveka Kulharia, András Horváth, Philip Torr. "Domain Partitioning Network".

Chapter 4: Botos Csaba, Xiaojuan Qi, Arslan Chaudhry, Puneet Dokania, Philip Torr. "Multilevel Knowledge Transfer for Cross-Domain Object Detection".

Chapter 5: Botos Csaba, Wenxuan Zhang, Matthias Müller, Ser-Nam Lim, Mohamed Elhoseiny, Philip Torr, Adel Bibi. "Label Delay in Continual Learning".

1.6 Thesis Overview

In this thesis, we have explored the use of unsupervised methods to bridge distribution discrepancies in deep learning vision tasks. Through a progression of contributions that address standard machine learning assumptions to increasingly complex and realistic settings, we have demonstrated the effectiveness of unsupervised learning in enabling efficient and adaptable models.

The proposed unsupervised methods have shown promising results in leveraging unlabeled visual data, reducing the reliance on costly and time-consuming manual annotations. By focusing on techniques that are computationally efficient and suitable for consumer-grade hardware, this thesis contributes to the democratization of machine learning, making it more accessible and applicable to a broader range of real-world problems.

The findings and techniques presented in this thesis have broader implications beyond the realm of vision tasks. The unsupervised methods developed here can potentially be extended to other domains, such as audio and sensor data, enabling efficient learning from unlabeled data in various application scenarios. Furthermore, the combination of unsupervised and supervised methods is an exciting avenue for future research, promising enhanced performance and generalization.

As machine learning continues to evolve and tackle increasingly complex real-world challenges, the importance of unsupervised learning will only grow. This thesis represents a significant step forward in the development of efficient and adaptable unsupervised methods, paving the way for future research and applications in this exciting field.

In conclusion, this thesis has explored the use of unsupervised methods to bridge distribution discrepancies in deep learning vision tasks, demonstrating the effectiveness and efficiency of these methods in increasingly complex and realistic settings. The contributions of this thesis have the potential to impact a wide range of real-world applications and pave the way for future research in unsupervised learning. As machine learning continues to evolve and tackle ever more challenging problems, the importance of unsupervised learning will only continue to grow, making the findings and techniques presented in this thesis all the more relevant and impactful.

Unsupervised Clustering for Efficient Image Recognition

Chapter Teaser

What is the importance of parameter sharing when solving complex vision tasks? Can we reduce the model inference time by learning subset specific features that are only computed when dealing with samples from the subset? Can we remove the human priors by automating the subset assignment process? How can we use the underlying mixture of expert systems to find particularly useful partitions? Are the partitions that benefit the model performance semantically meaningful?

Abstract

Deep learning models for vision tasks are trained on large datasets under the assumption that there exists a universal representation that can be used to make predictions for all samples. Whereas high complexity models are proven to be capable of learning such representations, a mixture of experts trained on specific subsets of the data can infer the labels more efficiently. However using mixture of experts poses two new problems,

namely (**i**) assigning the correct expert at inference time when a new unseen sample is presented. (**ii**) Finding the optimal partitioning of the training data, such that the experts rely the least on common features. In Dynamic Routing (DR) [129] a novel architecture is proposed where each layer is composed of a set of experts, however without addressing the two challenges we demonstrate that the model reverts to using the same subset of experts. In our method, Diversified Dynamic Routing (DivDR) the model is explicitly trained to solve the challenge of finding relevant partitioning of the data and assigning the correct experts in an unsupervised approach. We conduct several experiments on semantic segmentation on Cityscapes and object detection and instance segmentation on MS-COCO showing improved performance over several baselines.

2.1 Introduction

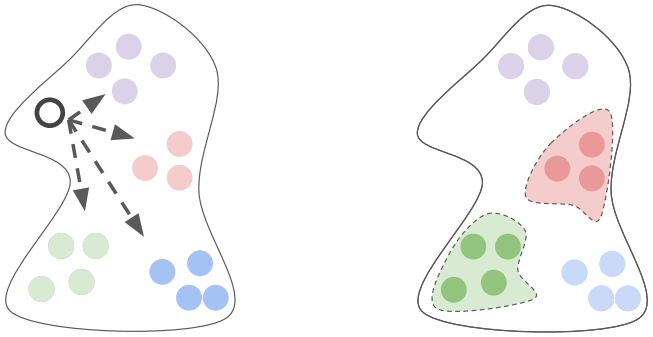
In recent years, deep learning models have made huge strides solving complex tasks in computer vision, e.g. segmentation [143, 30] and detection [71, 178], and reinforcement learning, e.g. playing atari games [158]. Despite this progress, the computational complexity of such models still poses a challenge for practical deployment that requires accurate real-time performance. This has incited a rich body of work tackling the accuracy complexity trade-off from various angles. For instance, a class of methods tackle this trade-off by developing more efficient architectures [201, 238], while others initially train larger models and then later distill them into smaller more efficient models [93, 229, 83]. Moreover, several works rely on sparse regularization approaches [214, 50, 186] during training or by performing a post-training pruning of model weights that contribute marginally to the final prediction. While listing all categories of methods tackling this trade-off is beyond the scope of this paper, to the best of our knowledge, they all share the assumption that predicting the correct label requires a universal set of features that works best for all samples. We argue that such an assumption is often broken even in well curated datasets. For example, in the task of segmentation, object sizes can widely vary across the dataset

requiring different computational effort to process. That is to say, large objects can be easily processed under lower resolutions while smaller objects require processing in high resolution to retain accuracy. This opens doors for class of methods that rely on *local experts*; efficient models trained directly on each subset separately leveraging the use of this local bias. However, prior art often ignore local biases in the training and validation datasets when tackling the accuracy-efficiency trade-off for two key reasons illustrated in Figure 2.1. (**i**) Even under the assumption that such local biases in the training data are known, during inference time, new unseen samples need to be assigned to the correct local subset so as to use the corresponding *local expert* for prediction (Figure 2.1 left). (**ii**) Such local biases in datasets are not known **apriori** and may require a prohibitively expensive inspection of the underlying dataset (Figure 2.1 right).

In this paper, we take an orthogonal direction to prior art on the accuracy-efficiency trade-off by addressing the two challenges in an unsupervised manner. In particular, we show that training *local experts* on learnt subsets sharing local biases can jointly outperform *global experts*, i.e. models that were trained over the entire dataset. We summarize our contributions in two folds.

1. We propose Diversified Dynamic Routing (DivDR); an unsupervised learning approach that trains several local experts on learnt subsets of the training dataset. At inference time, DivDR assigns the correct local expert for prediction to newly unseen samples.

2. We extensively evaluate DivDR and compare against several existing methods on semantic segmentation, object detection and instance segmentation on various datasets, i.e. Cityscapes [44] and MS-COCO [136]. We find that DivDR compared to existing methods better trades-off accuracy and efficiency. We complement our experiments with various ablations demonstrating robustness of DivDR to choices of hyperparameters.



Expert assignment
during inference

Finding meaningful subsets
during training

Figure 2.1: The figure depicts the two main challenges in learning local experts on subsets on subsets of the dataset with local biases. First, even when the subsets in the training dataset is presented where there is a local expert per subset, the challenge remains in assigning the local expert for new unseen samples (left Figure). The second challenge is that the local biases in the training data are not available during training time (right Figure).

2.2 Related Work

In prior literature model architectures were predominantly hand-designed, meaning that hyper-parameters such as the number and width of layers, size and stride of convolution kernels were predefined. In contrast, Neural Architecture Search [254, 138] revealed that searching over said hyper-parameter space is feasible provided enough data and compute power resulting in substantial improvement in model accuracy. Recently, a line of research [128, 137, 29, 201, 211] also proposed to constrain the search space to cost-efficient models that jointly optimize the accuracy and the computational complexity of the models. Concurrently, cost-efficient inference has been also in the focus of works on dynamic network architectures [160, 237, 218, 226], where the idea is to allow the model to choose different architectures based on the input through gating computational blocks during inference.

For example, Li et al. [129] proposed an end-to-end dynamic routing framework that generates routes within the architecture that vary per input sample. The search space

of [129], inspired by Auto-DeepLab [137], allows exploring spatial up and down-sampling between subsequent layers which distinguishes the work from prior dynamic routing methods. One common failure mode of dynamic models is mentioned in [160], where during the initial phase of the training only a specific set of modules are selected and trained, leading to a static model with reduced capacity. This issue is addressed by Mullapudi et *al.* [160] through clustering the training data in advance based on latent representations of a pretrained image classifier model, whereas [211] uses the Gumbel-Softmax reparameterization [109] to improve diversity of the dynamic routes. In this work, to mitigate this problem, we adopt the metric learning Magnet Loss [180] which acts as an improvement over metric learning methods that act on the instance level, e.g. Triplet Loss [219, 118], and Contrastive Learning methods [42, 86]. This is since it considers the complete distribution of the underlying data resulting in a more stable clustering. To adapt Magnet Loss to resolving the Dynamic Routing drawbacks, we use it as an unsupervised approach to increase the distance between the forward paths learned by the Dynamic Routing model this is as opposed to clustering the learned representations, i.e. learning clustered dynamic routes as opposed to clustered representations.

We review the recent advances on semantic segmentation and object detection which are utilized to validate our method in this work. For semantic segmentation, numerous works have been proposed to capture the larger receptive field [246, 30, 31, 32] or establish long-range pixel relation [247, 104, 192] based on Fully Convolutional Networks [143]. As mentioned above, with the development of neural network, Neural Architecture Search (NAS)-based approaches [29, 137, 162] and dynamic networks [129] are utilized to adjust network architecture according to the data while being jointly optimized to reduce the cost of inference. As for object detection, modern detectors can be roughly divided into one-stage or two-stage detectors. One-stage detectors usually make predictions based on the prior guesses, like anchors [176, 135] and object centers [205, 250]. Meanwhile, two-stage detectors predict boxes based on predefined proposals in a coarse-to-fine manner [73,

71, 178]. There are also several advances in Transformer-based approaches for image recognition tasks such as segmentation [248, 227] and object detection [21, 253], and while our method can be generalized to those architectures as well, it is beyond the scope of this paper.

2.3 DivDR: Diversified Dynamic Routing

We first start by introducing Dynamic Routing. Second, we formulate our objective of the iterative clustering of the dataset and the learning of experts per dataset cluster. At last, we propose a contrastive learning approach based on *magnet loss* [180] over the gate activation of the dynamic routing model to encourage the learning of different architectures over different dataset clusters.

2.3.1 Dynamic Routing Preliminaries

The Dynamic Routing (DR) [129] model for semantic segmentation consists of L sequential feed-forward layers in which dynamic *nodes* process and propagate the information. Each dynamic node has two parts: (**i**) the *cell* that performs a non-linear transformation to the input of the node; and (**ii**) the *gate* that decides which node receives the output of the cell operation in the subsequent layer. In particular, the gates in DR determine what resolution/scale of the activation to be used. That is to say, each gate determines whether the activation output of the cell is to be propagated at the same resolution, up-scaled, or down-scaled by a factor of 2 in the following layer. Observe that the gate activation determines the *architecture* for a given input since this determines a unique set of connections defining the architecture. The output of the final layer of the nodes are up-sampled and fused by 1×1 convolutions to match the original resolution of the input image. For an input-label pair (x, y) in a dataset $\mathcal{D}$ of N pairs, let the DR network parameterized by θ be given as $f_\theta : \mathcal{X} \to \mathcal{Y}$ where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Moreover, let $\mathcal{A}_{\tilde{\theta}} : \mathcal{X} \to [0, 1]^n$, where $\theta \supseteq \tilde{\theta}$, denote the gate activation map for a given input, i.e. the

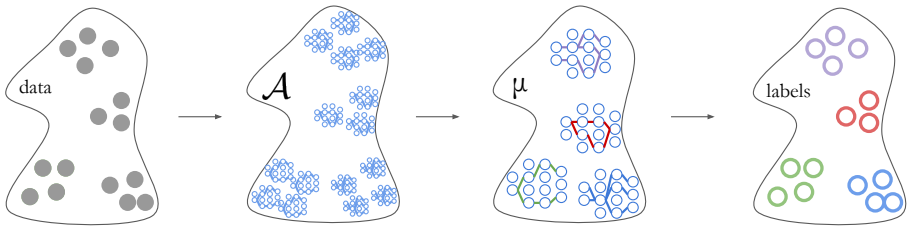


Figure 2.2: **Gate Activation cluster assignment.** To update the local experts, DivDR performs K-means clustering on the gate activations over the $\mathcal{A}(x_i)$ $\forall i$ in the training examples with fixed model parameters θ.

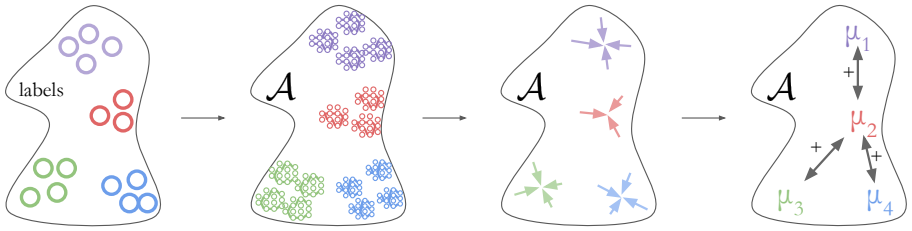


Figure 2.3: **Gate Activation Diversification.** We use the labels from the cluster assignment to reduce the *intra-cluster* variance and increase the *inter-cluster* variance by updating model parameters θ.

gates determining the architecture discussed earlier, then the training objective for DR networks under computational budget constraints have the following form:

$$\mathcal{L}_{DR} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{seg}\big(f_\theta(x_i), y_i\big) + \lambda \mathcal{L}_{cost}(\mathcal{A}_{\tilde{\theta}}(x_i)). \tag{2.1}$$

We will drop the subscript $\tilde{\theta}$ throughout to reduce text clutter. Note that $\mathcal{L}_{seg}$ and $\mathcal{L}_{cost}$ denote the segmentation and computational budget constraint respectively. Observe that when most of the gate activations are sparse, this incurs a more efficient network that may be at the expense of accuracy and hence the trade-off through the penalty λ.

2.3.2 Metric Learning in $\mathcal{A}$-space

Learning local experts can benefit performance both in terms of accuracy and computational cost. We propose an unsupervised approach to learning jointly the subset of the dataset and the soft assignment of the corresponding architectures. We use the DR framework for our approach.

We first assume that there are K clusters in the dataset for which we seek to learn an expert on each. Moreover, let $\{\mu_{\mathcal{A}_i}\}_{i=1}^K$, denote the cluster centers representing K different gate activations. Note that as per the previous discussion, each gate activation $\mu_{\mathcal{A}_i} \in [0,1]^n$ corresponds to a unique architecture. The set of cluster centers representing gate activations $\{\mu_{\mathcal{A}_i}\}_{i=1}^K$ can be viewed as a set of prototypical architectures for K different subsets in the datasets. Next, let $\mu(x)$ denote the nearest gate activation center to the gate activation $\mathcal{A}(x)$, i.e. $\mu(x) = \arg\min \|\mathcal{A}(x) - \mu_{\mathcal{A}_i}\|$. Now, we seek to solve for both the gate activation centers $\{\mu_{\mathcal{A}_i}\}_{i=1}^K$ and the parameters θ such that the gate activation centers are pushed away from one another. To that end, we propose the alternating between clustering and the minimization of a *magnet loss*[180] variant. In particular, for a given fixed set of activating gates centers $\{\mu_{\mathcal{A}_i}\}_{i=1}^K$, we consider the following loss function:

$$\mathcal{L}_{\text{clustering}}(\mathcal{A}(x_i)) = \Bigg\{ \alpha + \frac{1}{2\sigma^2}\|\mathcal{A}(x_i) - \mu(x_i)\| + \log\left(\sum_{k:\mu_{\mathcal{A}_k} \neq \mu(x_i)} e^{-\frac{1}{2\sigma^2}\|\mathcal{A}(x_i) - \mu_{\mathcal{A}_k}\|} \right) \Bigg\}_+ . \tag{2.2}$$

Note that $\{x\}_+ = \max(x, 0)$, $\sigma^2 = \frac{1}{N-1}\sum_i^N \|\mathcal{A}(x_i) - \mu(x_i)\|^2$, and that $\alpha \geq 0$. Observe that unlike in *magnet loss*, we seek to cluster the set of architectures by separating the gate activations. Note that the penultimate term pulls the architecture, closer to the most similar prototypical architecture while the last term pushes it away from all other architectures. Therefore, this loss incites the learning of K different architectures where each input x_i will be assigned to be predicted with one of the K learnt architectures. To that end, our overall *Diversified* DR loss is given as follows:

$$\mathcal{L}_{\text{DivDR}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{segm}(f_\theta(x_i), y_i) + \lambda_1 \mathcal{L}_{cost}(\mathcal{A}(x_i)) + \lambda_2 \mathcal{L}_{clustering}(\mathcal{A}(x_i)). \tag{2.3}$$

We then alternate between minimizing $\mathcal{L}_{\text{DivDR}}$ over the parameters θ and the updates of

the cluster centers $\{\mu_{\mathcal{A}_i}\}_{i=1}^{K}$. In particular, given θ, we update the gate activation centers by performing K-Means clustering [149] over the gate activations. That is to say, we fix θ and perform K-means clustering with K clusters over all the gate activations from the dataset $\mathcal{D}$, i.e. we cluster $\mathcal{A}(x_i)\ \forall i$ as shown in Figure 2.2. Moreover, alternating between optimizing $\mathcal{L}_{\text{DivDR}}$ and updating the gate activation cluster centers over the dataset $\mathcal{D}$, illustrated in Figure 2.3, results in a diversified set of architectures driven by the data that are more efficient, i.e. learning K local experts that are accurate and efficient.

2.4 Experiments

We show empirically that our proposed DivDR approach can outperform existing methods in better trading off accuracy and efficiency. We demonstrate this on several vision tasks, i.e. semantic segmentation, object detection, and instance segmentation. We start first by introducing the datasets used in all experiments along along with the implementation details. We then present the comparisons between DivDR and several other methods along with several ablations.

2.4.1 Datasets

We mainly prove the effectiveness of the proposed approach for semantic segmentation, object detection, and instance segmentation on two widely-adopted benchmarks, namely Cityscapes [44] and Microsoft COCO [136] dataset.

Cityscapes. The Cityscapes [44] dataset contains 19 classes in urban scenes, which is widely used for semantic segmentation. It is consist of 5000 fine annotations that can be divided into 2975, 500, and 1525 images for training, validation, and testing, respectively. In the work, we use the Cityscapes dataset to validate the proposed method on semantic segmentation.

COCO. Microsoft COCO [136] dataset is a well-known for object detection benchmarking

Table 2.1: Comparison with baselines on the Cityscapes [44] validation set. * Scores from [129] were reproduced using the official implementation. The evaluation settings are identical to [129]. We calculate the average FLOPs with 1024 × 2048 size input.

Method	**Backbone**	**mIoU**$_{val}$(%)	**GFLOPs**
BiSenet [238]	ResNet-18	74.8	98.3
DeepLabV3 [31]	ResNet-101-ASPP	78.5	1778.7
Semantic FPN [117]	ResNet-101-FPN	77.7	500.0
DeepLabV3+ [32]	Xception-71-ASPP	79.6	1551.1
PSPNet [246]	ResNet-101-PSP	79.7	2017.6
Auto-DeepLab [137]	Searched-F20-ASPP	79.7	333.3
Auto-DeepLab [137]	Searched-F48-ASPP	80.3	695.0
DR-A [129]*	Layer16	72.7±0.6	58.7±3.1
DR-B [129]*	Layer16	72.6±1.3	61.1±3.3
DR-C [129]*	Layer16	74.2±0.6	68.1±2.5
DR-Raw [129]*	Layer16	75.2±0.5	99.2±2.5
DivDR-A	Layer16	73.5±0.4	57.7±3.9
DivDR-Raw	Layer16	75.4±1.6	95.7±0.9

which contains 80 categories in common context. In particular, it includes 118k training images, 5k validation images, and 20k held-out testing images. To prove the performance generalization, we report the results on COCO's validation set for both object detection and instance segmentation tasks.

2.4.2 Implementation Details

In all training settings, we use SGD with a weight decay of 10^{-4} and momentum of 0.9 for both datasets. For semantic segmentation on Cityscapes, we use the exponential learning rate schedule with an initial rate of 0.05 and a power of 0.9. For fair comparison, we follow the setting in [129] and use a batch size 8 of random image crops of size 768 × 768 and train for $180K$ iterations. We use random flip augmentations where input images are scaled from 0.5 to 2 before cropping. For object detection on COCO we use an initial learning rate of 0.02 and re-scale the shorter edge to 800 pixels and train for 90K iterations. Following prior art, random flip is adopted without random scaling.

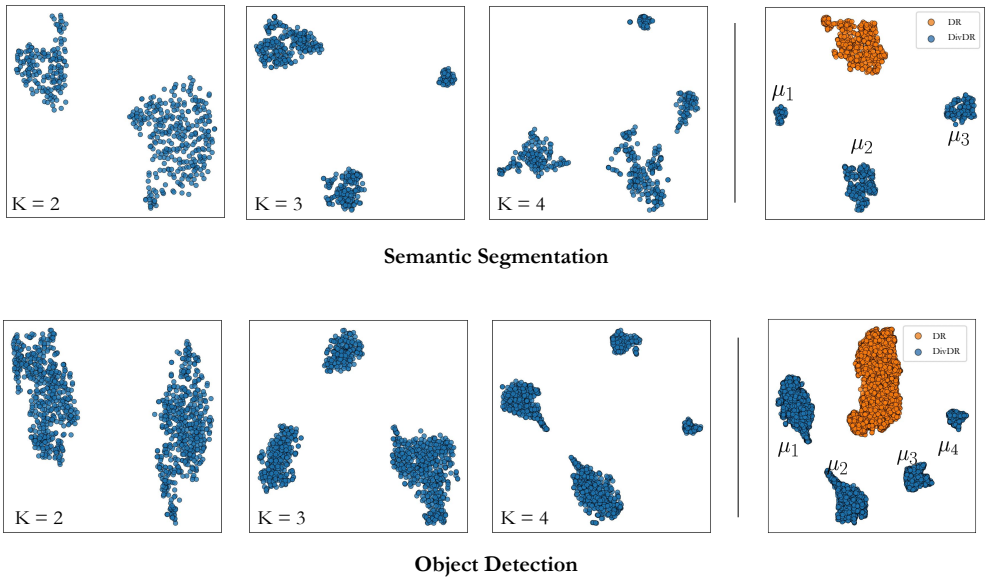


Figure 2.4: Visualizing the 183-dimensional $\mathcal{A}$-space of Dynamic Routing backbones trained for semantic segmentation on Cityscapes [44] (*top*) and 198-dimensional $\mathcal{A}$-space for object detection on COCO [136] (*bottom*) using t-SNE [210]. *Left:* varying number of *local experts*, $K = 2, 3, 4$. *Right:* joint t-SNE visualization of architectures of Dynamic Routing [129] (*orange*) and our approach (*blue*). It is clear that our method not only encourages diversity of the learned routes but also reduces variance in a specific cluster. Low *intra*-cluster variance is beneficial because it facilitates feature sharing between similar tasks

2.4.3 Semantic Segmentation

We show the benefits of our proposed DivDR of alternation between training with $\mathcal{L}_{\text{DivDR}}$ and computing the gate activations clusters through K-means on Cityscapes [44] for semantic segmentation. In particular, we compare two versions of our proposed unsupervised Dynamic Routing, namely with and without the computational cost constraint ($\lambda_1 = 0$ denoted as DivDR-Raw and $\lambda_1 = 0.8$ denoted as DivDR-A) against several variants of the original dynamic routing networks both constrained and unconstrained. All experiments are averaged over 3 seeds. As observed in Table 2.1, while both variants perform similarly in terms of accuracy (DR-Raw: 75.2%, DivDR: 75.4%), DivDR marginally improves the computational cost by 3.5 GFLOPs. On the other hand, when introducing cost efficiency

Table 2.2: Quantitative analysis of semantic segmentation on Cityscapes [44]. We report *Inter* and *Intra* cluster variance, that shows how far are the cluster centers are from each other in L_2 space and how close are the samples to the cluster centers respectively.

method	**mIoU**	**FLOPs**	**Inter**	**Intra**
DR-A	72.7	58.7	0.4	0.3
DivDR-A	72.0	49.9	0.6	0.2
DR-Raw	75.2	99.2	1.5	1.5
DivDR-Raw	75.7	98.3	1.2	0.5

constraint DivDR-A improves both the efficiency (58.7 GFLOPs to 57.7 GFLOPs) and accuracy (72.7% to 73.5%) as compared to DR-A. At last, we observe that comparing to other state-of-the-art, our unconstrained approach, performs similarly to BiSenet [238] with 74.8% accuracy while performing better in computational efficiency (98.3 GFLOPs vs. 95.7 GFLOPs).

Visualizing Gate Activations. We first start by visualizing the gate activations under different choices of the number of clusters K over the gate activation for DivDR-A. As observed from Figure 2.4, indeed our proposed $\mathcal{L}_{\text{DivDr}}$ results into clusters on local experts as shown by different gate activations $\mathcal{A}$ for $k \in \{2, 3, 4\}$. Moreover, we also observe that our proposed loss not only results in separated clusters of local experts, i.e. gate activations, but also with a small intra class distances. In particular, as shown in Table 2.2, our proposed DivDR indeed results in larger inter-cluster distances that are larger than the intra-cluster distnaces. The inter-cluster distances are computed as the average distance over all pair of cluster centers, i.e. $\{\mu_{\mathcal{A}_i}\}_{i=1}^{K}$ while the intra-cluster distances are the average distances over all pairs in every cluster. This indeed confirms that our proposed training approach results in K different architectures for a given dataset. Consequently, we can group the corresponding input images into K classes and visualize them to reveal common semantic features across the groups. For details see Fig 2.5. We find it interesting that despite we do not provide any direct supervision to the gates about the objects present on the images, the clustering learns to group semantically meaningful groups together.

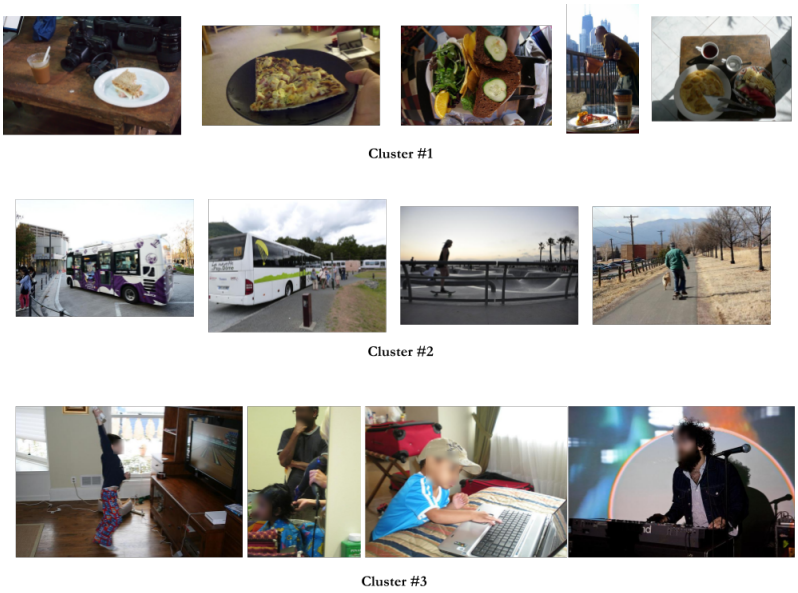


Figure 2.5: Visualization of images from the validation set of MS-COCO 2017 [136] challenge. In this training $K = 3$ and we visualize the top-5 images that fall closest to their respective cluster centers μ_i. Note that the dataset does not provide subset-level annotations, however our method uses different pathways to process images containing meals (*top row*), objects with wheels and outdoor scenes (*middle row*) and electronic devices (*bottom row*).

Ablating α and λ_2. Moreover, we also ablate the performance of α which is the separation margin in the hinge loss term of our proposed loss. Observe that larger values of α correspond to more enforced regularization on the separation between gate activation clusters. As shown in Figure 2.6 left, we observe that the mIOU accuracy and the FLOPs of our DivDR-A is only marginally affected by α indicating that a sufficient enough margin can be attained while maintaining accuracy and FLOPs trade-off performance.

2.4.4 Object Detection and Instance Segmentation

To further demonstrate the effectiveness on detection and instance segmentation, we validate the proposed method on the COCO datasets with Faster R-CNN [178] and Mask R-CNN [89] heads. As for the backbone, we extend the original dynamic routing networks with another 5-stage layer to keep consistent with that in FPN [134], bringing 17 layers in total. Similar to that in Sec. 2.4.3, no external supervision is provided to our proposed

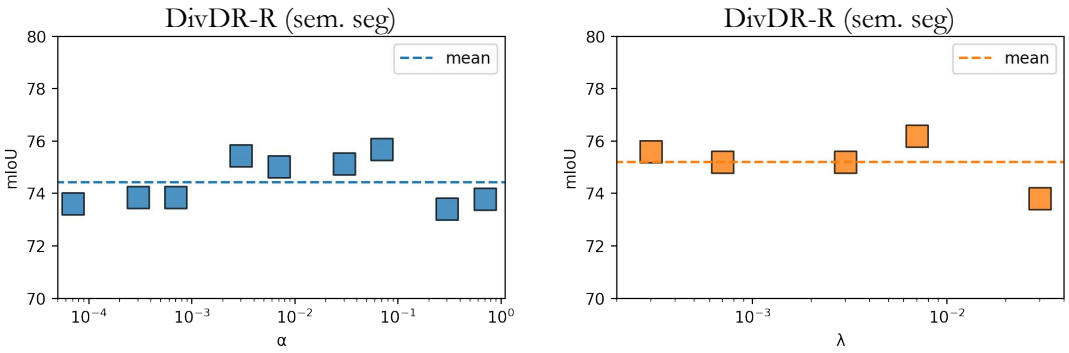


Figure 2.6: Ablation on the α (*left*) and λ_2 (*right*) parameter of the diversity loss term for Semantic Segmentation. The *mean* accuracy in case of the parameter sweep for λ_2 is higher since in each case the best performing α was used for the training. We can see that the method is stable regardless the choice of the parameters over various tasks.

Table 2.3: Quantitative comparison of Dynamic Routing [129] trained without the objective to diversify the paths and using various K for the clustering term. We omit $K = 1$ from our results as it reverts to forcing the model to use the same architecture, independent of the input image. Instead we report the baseline scores from [129] For comparison we report best Dynamic Routing [129] scores from 3 identical runs with different seeds.

(a) DivDR-A

K	mAP_{val}	GFLOPs	Inter	Intra
*	34.6	23.2	0.2	0.3
2	**35.1**	21.9	1.1	0.4
3	35.0	**19.2**	0.8	0.3
4	34.9	20.0	0.6	0.1

(b) DivDR-Raw

K	mAP_{val}	GFLOPs	Inter	Intra
*	37.8	38.2	0.5	0.7
2	36.5	**31.0**	0.6	0.5
3	37.4	32.6	1.2	0.5
4	**38.1**	32.8	0.7	0.2

DivDR during training. As presented in Tables 2.4 and 2.5, we conduct experiments with two different settings, namely without and with computational cost constraints. We illustrate the overall improvement over DR [129] across various hyper-parameters in Fig 2.8

Detection. Given no computational constraints, DivDR attains 38.1% mAP with 32.9 GFLOPs as opposed to 37.7% mAP for DR-R. While the average precision is similar, we observe a noticeable gain computational reduction of 5.3 GFLOPs. Compared with the ResNet-50-FPN for backbone, DivDR achieves similar performance but a small gain of 0.2% but with half of the GFLOPs (32.9 GFLOPs vs. 95.7 GFLOPs). When we introduce the computational regularization, the cost is reduced to 19.8 GFLOPs while

Table 2.4: Comparison with baselines on the COCO [136] **detection** validation set. * Scores from [129] were reproduced using the official implementation. The evaluation settings are identical to [178] with single scale. We calculate the average FLOPs with 800×800 size input

Method	**Backbone**	**mAP**$_{val}$	**GFLOPs**
Faster R-CNN [178]	ResNet-50-FPN	37.9	88.4
DR-A [129]*	Layer17	32.1±5.0	20.9±2.1
DR-B [129]*	Layer17	36.5±0.2	24.4±1.2
DR-C [129]*	Layer17	37.1±0.2	26.7±0.4
DR-R [129]*	Layer17	37.7±0.1	38.2±0.0
DivDR-A	Layer17	35.4±0.2	19.8±1.0
DivDR-R	Layer17	38.1±0.0	32.9±0.1

Table 2.5: Comparison with baselines on the COCO [136] **segmentation** validation set. * Scores from [129] were reproduced using the official implementation. The evaluation settings are identical to [178] with single scale. We calculate the average FLOPs with 800 × 800 size input

Method	**Backbone**	**mAP**$_{val}$	**GFLOPs**
Mask R-CNN [178]	ResNet-50-FPN	35.2	88.4
DR-A [129]*	Layer17	31.8±3.1	23.7±4.2
DR-B [129]*	Layer17	33.9±0.4	25.2±2.3
DR-C [129]*	Layer17	34.3±0.2	28.9±0.7
DR-R [129]*	Layer17	35.1±0.2	38.2±0.1
DivDR-A	Layer17	33.4±0.2	24.5±2.3
DivDR-R	Layer17	35.1±0.1	32.9±0.2

the performance is preserved with 35.4% mAP. Compared with that in DR-A, we observe that while Div-DR constraibntconstrainted enjoys a 1.1 lower GLOPS, it enjoys improved precision of 3.3% (35.4% mAP vs. 32.1% mAP) with a lower standard deviation.We believe that this is due to the local experts learnt for separate subsets of the data.

Instance Segmentation. As for the task of instance, as observed in Table 2.5, DivDR unconstrainted performs similarly to DR-R with 35.1% mAP. However, DivDR better trades-off the GLOPs with with a 32.9 GFLOPs in the unconstrained regime as opposed to 38.2 GLOPS. This is similar to the observations made in the detection experiments. Moreover, when computational constraints are introduced, DivDR enjoys a similar GLOPs

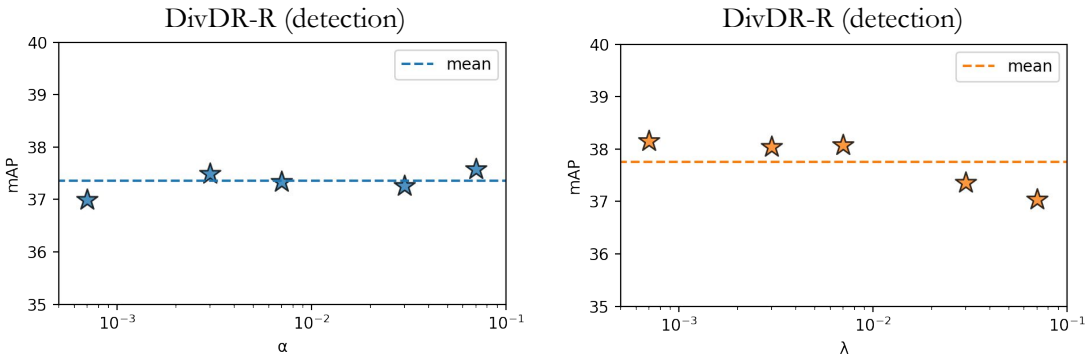


Figure 2.7: Ablation on the α (*left*) and λ_2 (*right*) parameter of the diversity loss term for Object Detection. We can see that the method is stable regardless the choice of the parameters over various tasks.

as DR-A but with an improved 1.6% precision (33.4% mAP vs. 31.8% mAP).

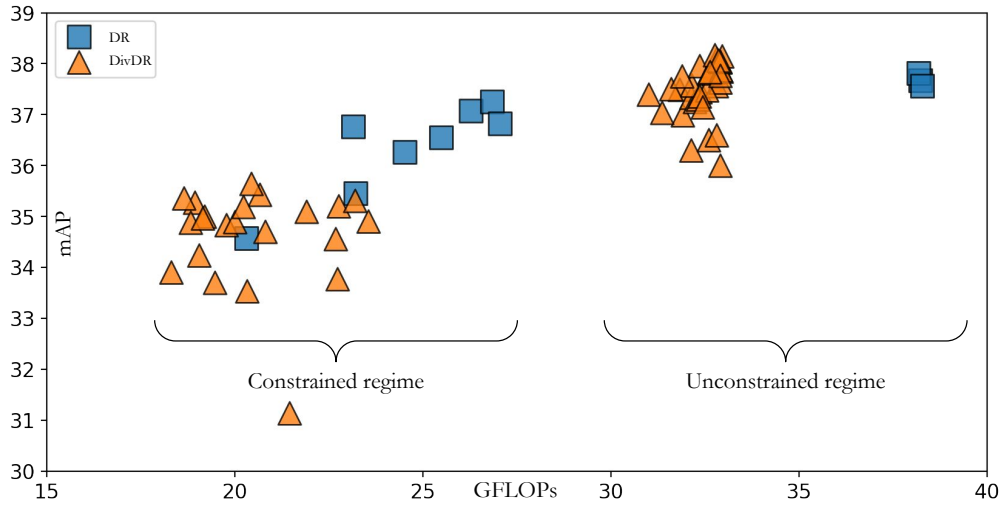


Figure 2.8: Evaluations of models trained on COCO [136] across different hyper-parameters

Ablating K. We compare the performance of our proposed DivDR under different choices of the number of clusters K over the gate activation for both unconstrained and constrained computational constraints, i.e. DivDR-A and DivDR-R respectively. We note that our proposed $\mathcal{L}_{\text{DivDr}}$ effectively clusters the gate activation cluster centers as shown in Figure 2.4. Moreover, we also observe that our proposed loss not only results in separated clusters of local experts, but also with a small intra-cluster distances as shown in Table 2.3. In particular, we observe that our proposed DivDR results in larger inter-cluster distances

that are larger than the intra-cluster distances (in contrast with DR [129]).

Ablating α and λ_2. As shown in Figure 2.7, we observe the choice of both α and λ_2 only marginally affect the performance of DivDR-A in terms of both mAP on the object detection task. However, we find that $\lambda_2 > 0.5$ starts to later affect the mAP for reduced computation.

2.5 Discussion and Future Work

In this paper we demonstrate the superiority of networks trained on a subset of the training set holding similar properties, which we refer to as *local experts*. We address the two main challenges of training and employing local experts in real life scenarios, where subset labels are not available during test nor training time. Followed by that, we propose a method, called Diversified Dynamic Routing that is capable of jointly learning local experts and subset labels without supervision. In a controlled study, where the subset labels are known, we showed that we can recover the original subset labels with 98.2% accuracy while maintaining the performance of a hypothetical *Oracle* model in terms of both accuracy and efficiency.

To analyse how well this improvement translates to real life problems we conducted extensive experiments on complex computer vision tasks such as segmenting street objects on images taken from the driver's perspective, as well as detecting common objects in both indoor and outdoor scenes. In each scenario we demonstrate that our method outperforms Dynamic Routing [129].

Even though this approach is powerful in a sense that it could improve on a strong baseline, we are aware that the clustering method still assumes subsets of *equal* and more importantly *sufficient* size. If the dataset is significantly imbalanced w.r.t. local biases the K-means approach might fail. One further limitation is that if the subsets are too

small for the *local experts* to learn generalizable representations our approach might also fail to generalize. Finally, since the search space of the architectures in this work is defined by Dynamic Routing [129] which is heavily focused on scale-varience. We believe that our work can be further generalized by analyzing and resolving the challenges mentioned above.

2.6 Acknowledgement

We thank Hengshuang Zhao for the fruitful discussions and feedback. This work is supported by the UKRI grant: Turing AI Fellowship EP/W002981/1 and EPSRC/MURI grant: EP/N019474/1. We would also like to thank the Royal Academy of Engineering. Botos Csaba was funded by Facebook Grant Number DFR05540.

2.7 Supplementary Material

2.7.1 Sensitivity to number of iterations between K-means update

In our early experiments we have found our method achieving satisfactory results if we kept the number of iterations between the K-means update low: ≤ 100. With lower frequency updates the diversity between the cluster centers was not sufficiently large, leading to the trivial solution, i.e. the model architecture learning to ignore the input image. In Deep Clustering [23] another technique is mentioned to avoid such trivial solutions, namely randomizing and manually altering the cluster centers in case they happen to be too close to each-other. We did not employ such techniques for our method.

On another note, we have found that while the cluster centers change significantly during the early phases of the training, the difference between two updates is less emphasized towards the end. This lead to a hypothesis that using an annealing policy on the frequency of the updates might be more practical as it could reduce the training time drastically, however such comparison is beyond the scope of this work.

In our experiments we use 50 iterations per K-means update everywhere.

2.7.2 Gathering gate activation values before or after non-linear layer

We have experimented with applying our method on the output of the final linear layer of each gate in our model. We have found that even though much higher variances can be achieved in terms of intra-cluster and inter-cluster diversity metrics, however most of these differences are marginalized by the final non-linear layer of the gates. In the most frequent case the model learned cluster centers that had negative values, which is entirely ignored by the ReLU-part of the non-linear function used by Dynamic Routing [129].

2.8 Conclusion

In this work we showed that training mixture of expert models by leveraging the biases of the underlying training dataset is beneficial both in terms of model accuracy and reduced inference time. The novelty in our method is that show how to cluster the training data into subsets in the absence of supervised labels by using the output of the gating mechanism instead the latent representation of the experts for the clustering algorithm. We show that using the recovered subsets not only reduces the inference time and improves the model accuracy, but results in semantically meaningful partitions of the dataset. While in the first two chapters we focused on models trained on i.i.d datasets, we continue our exploration of unsupervised methods in experimental settings where both labeled and unlabeled data is available for training, however only the unlabeled data is sampled from the distribution from which the test images are also sampled.

Unsupervised Clustering for Image Generation

Chapter Teaser

How to generate images when no corresponding captions or content descriptions are available? What are the reasons behind the instability of the adversarial training process? How to hedge our bets with multiple pairs of generator and discriminator models? How to assign the unlabeled images to the right model pair? Can we speed up the inference time by merging the generators into a single model? How to generalize the iterative min-max optimization to multiple discriminators? How to influence the output of the generated images?

Abstract

Standard adversarial training involves two agents, namely a generator and a discriminator, playing a mini-max game. However, even if the players converge to an equilibrium, the generator may only recover a part of the target data distribution, in a situation commonly referred to as *mode collapse*. In this work, we present the Domain Partitioning Network

(DoPaNet), a new approach to deal with mode collapse in generative adversarial learning. We employ multiple discriminators, each encouraging the generator to cover a different part of the target distribution. To ensure these parts do not overlap and collapse into the same mode, we add a classifier as a third agent in the game. The classifier decides which discriminator the generator is trained against for each sample. Through experiments on toy examples and real images, we show the merits of DoPaNet in covering the real distribution and its superiority with respect to the competing methods. Besides, we also show that we can control the modes from which samples are generated using DoPaNet.

3.1 Introduction

Generative Adversarial Networks [81] (GANs) consist of a deep generative model which is trained through a minimax game involving a competing generator and discriminator. The discriminator is tasked to differentiate real from fake samples, whereas the generator strives to maximize the mistakes of the discriminator. At convergence, the generator can sample from an estimate of the underlying real data distribution. The generated images, are observed to be of higher quality than models trained using maximum likelihood optimization. Consequently, GANs have demonstrated impressive results in various domains such as image generation [85], video generation [213], super-resolution [123], semi-supervised learning [51] and domain adaptation [251].

GANs are trained with the objective of reaching a Nash-equilibrium [153], which refers to the state where neither the discriminator nor the generator can further enhance their utilities unilaterally. However, the generator might miss some modes of the distribution even after reaching the equilibrium as it can simply fool the discriminator by generating from only few modes of the real distribution [79, 8, 28, 36, 184], and hence producing a limited diversity in samples. To address this problem, the literature explores two main approaches: Improving GAN learning to practically reach a better optimum [8, 156, 184,

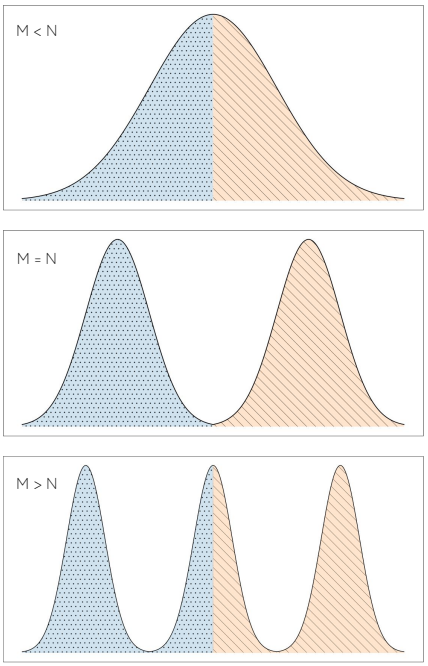


Figure 3.1: Illustration of the expected behaviour DoPaNet using two discriminators ($N = 2$), in case of a Uni-modal ($M = 1$, top), bi-modal ($M = 2$, middle) and tri-modal ($M = 3$, bottom) target distribution. The classifier ensures that the generated modes (in orange and blue) corresponding to two different discriminators do not overlap.

9, 85, 17], or explicitly forcing GANs to produce various modes by design [36, 68, 54, 28, 141]. We hereby follow the latter strategy and propose a new way of dealing with GAN mode collapse. By noticing that using a single discriminator often leads to the generator covering only a part of the data, we bring more discriminators to the game such that each incentivises the generator to cover an additional mode of the data distribution. For each discriminator to focus on a different target mode, we introduce a third player, a classifier Q that decides the discriminator to be trained using a given real sample. To ensure that these various target data modes do not collapse into the same mode, the classifier Q also decides the discriminator to train the generator for a given generated sample. We find that this strategy, illustrated in Figure 3.2, yields better coverage of the real data distribution at convergence and simultaneously improves the stability of the training as well.

We showcase our method on demonstrative toy problems and show that it outperforms competing methods in avoiding mode collapse. We show that the Q network is able to

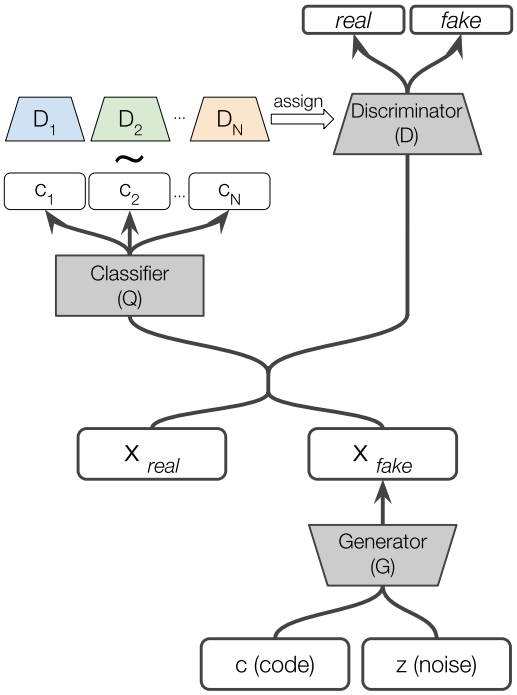


Figure 3.2: DoPaNet, our proposed framework. Here $\sim$ denotes a sampling operation from the categorical probability distribution $Q(x)$. c is a categorical code with one-hot encoding. Using the resulting category index $\sigma \sim Q(x)$ we select the corresponding discriminator $D_{i=\sigma}$ and connect it to the computation graph. From the perspective of the real sample x and the generated sample $\hat{x}$ the respective computation graph is fully-differentiable, and can be trained like the standard GAN [80].

distinguish different modes of the real data and therefore each discriminator works on a separate mode. This ensures that the generator can sample from a different mode for every input code vector. We also show DoPaNet's ability to generate good quality and diversified images covering various modes present in the datasets of real images.

We also provide theoretical analysis to show that at global optimum of the objective, the generator replicates the real distribution, categorized into different modes such that it can sample from any mode given the corresponding code vector c.

3.2 Related work

There is a rich literature on improving training stability and increasing sample diversity for GANs. We only focus on a selection of works that relate closely to ours. [10] introduces theoretical formulation stating the importance of multiple generators and discriminators in

order to completely model the data distribution. GMAN [54] proposes using multiple discriminators. They explore 3 settings where the generator can either be trained against the best discriminator, the averaged discriminators, or the weighted averaged discriminators. This helps training the network without modifying the minimax objective. Even though they use multiple discriminators, all of them are trained using all of the available real data, which does not explicitly help in avoiding mode collapse. We improve on this strategy by adding a classifier as a third component, with the task of choosing the discriminator for the given input sample during training, therefore each of the multiple discriminators specializes on a different part of the real data distribution. We also compare DoPaNet with GMAN [54] in our experiments (Section 3.4). Triple-GAN [126] incorporates a classifier in the adversarial training but it focuses on semi-supervised learning and therefore it needs some part of the real data to be labeled during training. It uses only one discriminator which is also conditioned on the sample labels. Contrarily our aim is to circumvent the mode collapse problem in the general case where the labels of the samples may not be available. InfoGAN [36] uses a Q network to maximize mutual information between the input code to the generator and its generated samples. It helps in disentangling several factors of variation, e.g. writing styles in case of digits, pose from lightning, etc. It is different from our approach as it uses the Q network as well to train the generator. Hence it is possible that the generator colludes with Q in disentangling the factors of variation, but simultaneously fooling the discriminator, while sampling from only few modes of the data. It can therefore still face the mode collapse problem which we show in the experiments (Section 3.4). Several works propose using multiple generators [11, 67, 141]. For instance, MAD-GAN [68] improves the learning by compelling the generators to produce diverse modes implicitly using the discriminator. This is achieved by requiring the discriminator to identify the generator that produced the fake samples along with recognizing fake samples from reals. The discriminator does not explicitly force each generator to capture a different mode, while in our case the generator is urged to capture distinct modes

by being trained with different discriminators. We also show DoPaNet's superiority over MAD-GAN in our experiments (Section 3.4).

3.3 Method

In this section we first briefly discuss the preliminaries (3.3.1): the general objective for training Generative Adversarial Nets and conditional sampling and training. Then we detail the objective of DoPaNet (3.3.2) and how we optimize it.

3.3.1 Preliminaries

Generative Adversarial Networks Generative adversarial networks can be considered as a game, where players in the form of neural networks are optimized against each other. Let p_d be the *real* data distribution and p_g be the distribution learnt by the generator G. Different tasks are assigned to the players: firstly, the generator G takes an input noise $z \sim p(z)$ and returns a sample $\hat{x} = G(z)$. The discriminator D takes an input x which can either be a real sample from the training set or a sample produced by the generator. The discriminator then outputs a conditional probability distribution over the source of the sample x. In practice D is a binary classifier that ideally outputs 1 if the sample is real and 0 if the sample is fake. Formally the following min-max objective is iteratively optimized:

$$\begin{aligned} \min_G \max_D V(D,G) := & \mathop{\mathbb{E}}_{x \sim p_d} \left[\log D\left(x\right)\right] \\ & + \mathop{\mathbb{E}}_{z \sim p_z} \left[\log\left(1 - D\left(G\left(z\right)\right)\right)\right] \end{aligned} \tag{3.1}$$

The parameters of D are updated to maximize the objective while the generator G is trained to minimize it.

Conditional generation We can condition the modeled distribution by making G take a code vector c as an additional input to produce a sample $\hat{x}_c = G(z, c)$, as it is done in InfoGAN [36] and other conditional variants [157]. In our case, we restrict the code vector c to have a one-hot encoding. Defining the conditional probability distribution as $Q(x) = p_{c|x}$, we obtain an objective function for the classifier Q, the general cross-entropy loss:

$$\min_{Q} L(Q, G) := \mathop{\mathbb{E}}_{z \sim p_z, c \sim p_c} [CE(c, Q(G(z, c)))] \tag{3.2}$$

where $CE(.,.)$ is the cross entropy function. The conditional variants of the standard GAN settings optimize both Objectives (3.1) and (3.2), where G may or may not be optimized over Objective (3.2). We do not use G to optimize the Objective (3.2).

3.3.2 Our approach: DoPaNet

DoPaNet consists of three main components: A conditional generator G, a classifier Q and a set of independent discriminators $\{D_i\}$. We use categorical code vectors $c \in \{0, 1\}^N$ with one-hot encoding where N is the number of discriminators used. We use the notation c_i to denote the one-hot code vector c with value at the i^{th} index as 1. As illustrated in Figure 3.2, G generates a sample $\hat{x}_c = G(z, c)$. Next we feed the sample to the classifier Q to get the categorical probability distribution. For each generated sample we draw $\hat{\sigma} \sim Q(\hat{x}_c)$, i.e. $\hat{\sigma} \in [1, ..., N]$ that decides the corresponding discriminator and $D_{\hat{\sigma}}$ that is going to process the generated sample. Formally, we define $D(\hat{x}) := D_{\hat{\sigma}}(\hat{x}_c)$. Similarly, for the real sample $x \sim p_d$, we draw $\sigma \sim Q(x)$ and define the discriminator $D(x) := D_\sigma(x)$ for the sample x. Thus, for every sample, the discriminator used is decided by the classifier Q. This yields a fully-differentiable computational graph, despite the fact that the sampling operation $\sigma \sim Q(x)$ is non-differentiable. In other words, once D is selected using predictions from Q, the training requires no further modifications to the standard GAN optimization algorithm, therefore it is compatible with all recent advanced variants of GANs. In our

Algorithm 1 DoPaNet training algorithm

1. Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from the noise prior $p_z(z)$.
2. Sample minibatch of m code samples $\{c^{(1)}, \ldots, c^{(m)}\}$ from the code prior $p_c(c)$.
3. Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from the data generating distribution $p_d(x)$.
4. Update Q by ascending its stochastic gradient:

$$\nabla_{\theta_q} \frac{1}{m} \sum_{i=1}^{m} c^{(i)} \cdot \log Q\left(G\left(z^{(i)}, c^{(i)}\right)\right)$$

5. Decide for every input which of the N discriminators to use by sampling from the likelihood distribution of Q:

$$\sigma(i) \sim Q\left(x^{(i)}\right) \qquad \hat{\sigma}(i) \sim Q\left(G\left(z^{(i)}, c^{(i)}\right)\right)$$

6. For all $n \in [1, \ldots, N]$, define the set of samples that are assigned to the n^{th} discriminator D_n as:

$$\mathcal{D}_n = \left\{x^{(i)} | \sigma(i) = n\right\} \qquad \hat{\mathcal{D}}_n = \left\{G\left(z^{(i)}, c^{(i)}\right) | \hat{\sigma}(i) = n\right\}$$

7. For all $n \in [1, \ldots, N]$, update the n^{th} discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_{d_n}} \left(\frac{1}{|\mathcal{D}_n|} \sum_{x \in \mathcal{D}_n} \log D_n(x) + \frac{1}{|\hat{\mathcal{D}}_n|} \sum_{\hat{x} \in \hat{\mathcal{D}}_n} \log\left(1 - D_n(\hat{x})\right) \right)$$

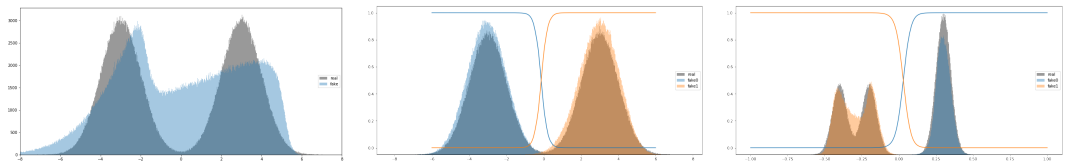
8. Repeat sampling of minibatches of noise and code samples as in steps (a) and (b).
9. Decide for every i^{th} input which of the N discriminators to use by sampling from likelihood distribution of Q as described previously.
10. Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D_{\hat{\sigma}(i)}\left(G\left(z^{(i)}, c^{(i)}\right)\right)\right)$$

experiments we define p_z as a standard normal distribution and p_c as a uniform categorical distribution unless otherwise stated.

Let us define the minimax objective for DoPaNet:

$$\begin{aligned} \min_G \max_{\{D_i\}_{i=1}^N} M(\{D_i\}_{i=1}^N, G) := & \mathop{\mathbb{E}}_{\substack{x \sim p_d \\ \sigma \sim Q(x)}} \left[\log D_\sigma(x)\right] \\ & + \mathop{\mathbb{E}}_{\substack{z \sim p_z, c \sim p_c \\ \hat{\sigma} \sim Q(G(z,c))}} \left[\log\left(1 - D_{\hat{\sigma}}(G(z,c))\right)\right] \end{aligned} \tag{3.3}$$



(a) Standard GAN (b) DoPaNet, 2 modes, 2 Disc. (c) DoPaNet, 3 modes, 2 Disc.

Figure 3.3: Theoretical analysis: The figures are plotted using 100000 samples and 1000 bins histogram where the grey area represents the real data distribution. *Fig. (a)*: Standard GAN - In special case, $N = 1$ DoPaNet is equivalent to the standard GAN. The blue area represents the model distribution. *Fig. (b, c)*: The orange and blue area represent the generations corresponding to c_1 and c_2. The number of discriminators is fixed at $N = 2$, while the number of modes is $M = 2, 3$ respectively. The orange and blue curves depict the predicted class probability of c_1 and c_2 respectively by the classifier Q. It can be seen that support of p_d^1 (i.e. $x < 0$) can be considered disjoint from the support of p_d^2 (i.e. $x > 0$) due to steep change in $p_{c|x}$ around $x = 0$. It also shows that the real distribution area corresponding to c_1 and c_2: ρ_{S_1} and ρ_{S_2} is almost equal in proportion and therefore equal to $1/2$.

We train DoPaNet by iteratively optimizing the following objective function (refer Algorithm 1):

$$\min_G \max_{\{D_i\}_{i=1}^N} M(\{D_i\}_{i=1}^N, G) + \min_Q L(Q, G) \tag{3.4}$$

3.3.3 Theoretical Analysis

The classifier Q is trained only using Objective 3.2, and is applied on the generated samples $\hat{x}$ as well as the real samples x to decide the discriminator to use. It is optimal when it is able to correctly classify the generated samples $\hat{x}$ into their corresponding c_i's. Empirically we observe that the classifier Q is easily able to reach its optimum, as can be observed in the Figure 3.3(b) and 3.3(c), as the blue and orange curves (samples predicted as c_1 and c_2 respectively) coincide with the samples forming blue and green area (samples generated using c_1 and c_2 respectively). Interestingly, we observe that the classifier Q is able to indirectly control the generator G through the discriminators as G groups its generations according to the code vectors c_i.

Here we provide formal theoretical formulation of our model with proof presented in

Appendix 3.7.

Lemma 3.3.1. *For optimal Q and fixed G, the optimal D_i, $\forall i \in [1, \ldots, N]$ is*

$$D_i^*(x) = \frac{\rho_{S_i} p_d^i(x)}{\rho_{S_i} p_d^i(x) + \frac{1}{N} p_g^i(x)} \tag{3.5}$$

where $S_i = \{x \in Supp(p_d) | Q(x) = c_i\}$, $\rho_{S_i} = \int_{x \in S_i} p_d(x) \mathrm{d}x$, p_d^i is a probability distribution such that $p_d^i(x) = \frac{p_d(x)}{\rho_{S_i}}$ and $Supp(p_d^i) = S_i$, and $p_g^i(x) = p_z(z)$ such that $G(z, c_i) = x$.

We can now reformulate the minimax game as

$$U(G) = \max_{\{D_i\}_{i=1}^N} M(\{D_i\}_{i=1}^N, G)$$

Theorem 3.3.2. *In case of N discriminators, the global minimum of $U(G)$ is achieved if and only if $p_g^i(x) = p_d^i(x)$, $\forall i \in [1, \ldots, N]$. When $\rho_{S_i} = 1/N$, the global minimum value of $U(G)$ is $-\log(4)$.*

Sampling from p_d^i is same as sampling from the i^{th} mode of the real distribution, the mode that covers the set of samples $S_i = \{x \in Supp(p_d) | Q(x) = c_i\}$. Please note that we can assume that each of $\{p_d^i\}_{i=1}^N$ has a disjoint support. Figure 3.3(b) and 3.3(c) empirically show that the assumption of disjoint support of the distributions p_d^1 and p_d^2, which is decided by the classifier Q, is valid.

So, in theory each $G_i(.) = G(., c_i)$ should converge to a different mode as the target dataset distribution p_d^i is itself different $\forall i \in [1, \ldots, N]$. Hence, empirically the number of modes covered should essentially be at least more diverse than the standard GAN model. This is also observed in all our experiments as well as when comparing the Figures 3.3(a) and 3.3(b).

Corollary 3.3.2.1. *At global minimum of $U(G)$, the generative model G replicates the real distribution p_d, categorized into different modes.*

Thus our model DoPaNet can learn the real data distribution while also controlling the diversity of the generations by sampling from a different real mode corresponding to each c_i, which we also verify experimentally in the next section.

3.4 Experiments

We demonstrate the performance of our method DoPaNet on a diverse set of tasks with increasing complexity, involving probability density estimation and image generation. To illustrate the functioning of DoPaNet, we first set up two low-dimensional experiments (Section 3.4.1) using Gaussian Mixture Models (GMMs) as the target probability density function: 1D GMM and 2D GMM. For the 1D Gaussian Mixture case, we compare DoPaNet's robustness against other approaches by reproducing the experiment setting detailed in [68] and we outperform all competing methods both qualitatively and quantitatively. We also show DoPaNet's performance using multiple discriminators and show how the training dynamics change according to the number of discriminators. We observe that increasing the number of discriminators improves the performance of the network until the point where the number of discriminators exceeds the number of underlying modes. Using the 2D circular GMM, we show that classifier Q is able to learn good partitioning of the distribution and therefore each discriminator acts on samples from a different mode unlike GMAN [54]. We show that DoPaNet is able to utilize the capacity of multiple discriminators and we can control the mode the generator samples from using the code c. Even in this case, DoPaNet performs better in capturing all the modes.

We finally demonstrate qualitative results on commonly investigated datasets: Stacked-MNIST, CIFAR-10 and CelebA in Section 3.4.2. DoPaNet is able to generate good quality diverse samples. In case of CIFAR-10, we also show that we can generate samples from every class given the class label y. The information about the network architectures and the implementation details are provided in Appendix 3.10.

3.4.1 Synthetic low dimensional distributions

In DoPaNet, the role of the classifier Q is to partition both the real and generated datapoints into different clusters or modes, and each discriminator is consequently only trained on a separate cluster. In order to fully understand how this helps the training, we experimented with two toy datasets obtained using mixture of Gaussian variants: a 1D GMM with 5 modes, as used in [68], and a 2D circular GMM with 3 and 8 modes on the unit circle.

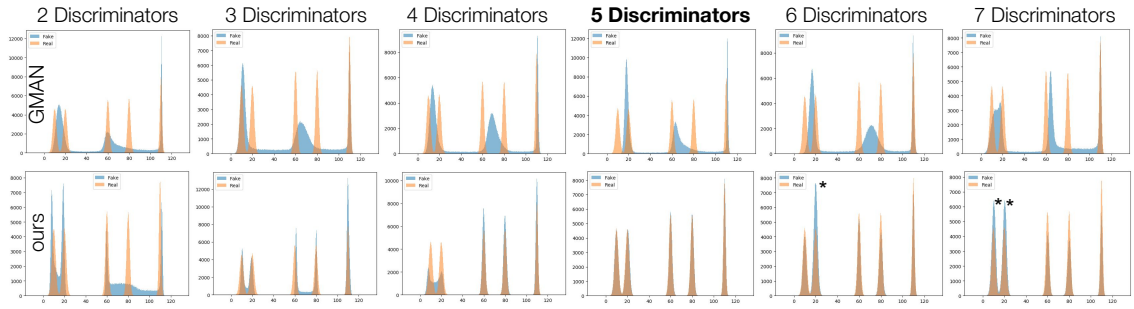


Figure 3.4: To study the behaviour of multi-discriminator settings with different number (N) of discriminators, we trained GMAN [54] and DoPaNet on a 1D data set with 5 modes. Both GMAN's (top) and DoPaNet's (bottom) results improved by adding discriminators while N is less or equal the number of modes. To justify this, we point at the case where $N = 5$: the models perform best since each class has to capture only a single mode. In the case of $N = 6$ and $N = 7$ DoPaNet decreased in performance only due to oversampling (marked with $*$ and $**$) of some classes.

Experiment setup and Evaluation details

First, we reproduced the 1D setting in [68] with 5 modes at $[10, 20, 60, 80, 110]$ and standard deviations $[3, 3, 2, 2, 1]$ respectively and we compare to the numbers reported in that paper in Table 3.1. We sampled $65,536$ data points each from the real distribution and the generator distribution. For each of these two distributions, we created a histogram using bin size of 0.1 with bins lying in the range of -10 to 130. We then obtained Chi-square distance as well as KL divergence between the generator distribution and the true data distribution using these two histograms. To compare against GMAN using different number of discriminators, we used $1,000,000$ samples (instead of $65,536$ above) and show

GAN Variants	Chi-square($\times 10^5$)	KL-Div
DCGAN*	0.90	0.322
WGAN*	1.32	0.614
BEGAN*	1.06	0.944
GoGAN*	2.52	0.652
Unrolled GAN*	3.98	1.321
Mode-Reg DCGAN*	1.02	0.927
InfoGAN*	0.83	0.210
MA-GAN*	1.39	0.526
MAD-GAN*	0.24	0.145
GMAN	1.44	0.69
DoPaNet	**0.03**	**0.02**

Table 3.1: 1D Gaussian Mixture Model experiment using best results from 3 runs for GAN variants that aims to solve mode collapse. Results for the GAN variants marked as $*$ were reproduced from [68].

the results in Table 3.2 and Figure 3.4.

We then introduce a 2D experiment setting with 2D Gaussian Mixture Model (GMM). It has multiple modes having covariance matrix of $0.01I$, where I is an identity matrix, and equally separated means lying on a unit circle (please refer to Figure 3.5 for the 3 mode case). For Table 3.3 we consider 8 modes and construct histograms using $1,000,000$ samples and bin size of 0.0028×0.0028 with bins lying in the range of $[-1.4, 1.4] \times [-1.4, 1.4]$.

For these experiments, we use uniform distribution $U(-1, 1)$ of dimension 64 for p_z and uniform categorical distribution for p_c to get the generations in both 1D and 2D experiments.

Observations

Comparing against other GAN variants In Table 3.1, we show that DoPaNet outperforms other GAN architectures on the 1D task by a large margin in terms of Chi-square distance and KL-Divergence. We believe that the success is due to the classifier Q's capability to learn to partition the underlying distribution easily. We also show in Table 3.3, that in the 2D task DoPaNet achieves better performance than GMAN [54] in terms of

both KL-Divergence an Chi-square.

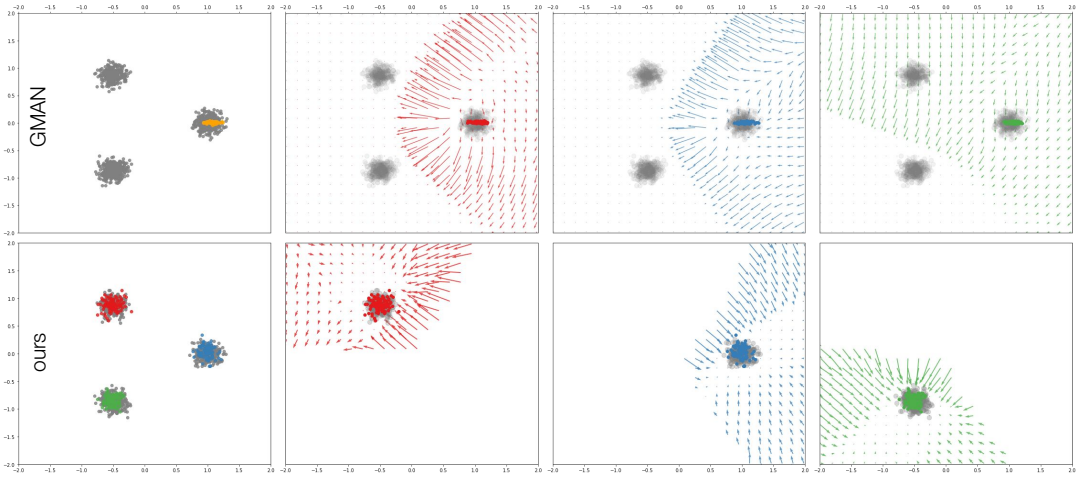


Figure 3.5: A circular 2D GMM with 3 modes on the unit circle with standard deviations of 0.01. We train GMAN and DoPaNet with $N = 3$ setting. In the first column we show the generated distribution for each method in colours (orange for GMAN, a single color since GMAN uses no conditioning on their samples, and red, green and blue in DoPaNet's case for samples generated using c_1, c_2 and c_3 respectively). In columns 2, 3 and 4 we plotted the gradient field for each of the equally separated data-points present in $[-2.0, 2.0] \times [-2.0, 2.0]$. Different colors show the gradient by different discriminators. For GMAN, each column shows the gradient field by the respective discriminator for each of the data-points. For DoPaNet, each of the data-points is first classified by Q and the respective discriminator is used to get the gradient field, which is shown in the corresponding column of 2-3. We see how classifier Q indirectly pushes different modes generated by G apart, more importantly the gradient field has the highest magnitude in the direction which separates the modes, while this phenomenon is not happening in the case of GMAN.

N	**Chi-square**($\times 10^7$)		**KL-Div**	
	GMAN	**DoPaNet**	GMAN	**DoPaNet**
2	5.00±6.80	**1.89±0.92**	1.74±0.63	**0.81±0.27**
3	2.96±2.88	**1.10±2.43**	1.50±0.57	**0.55±0.36**
4	3.41±2.73	**0.74±0.98**	1.48±0.42	**0.50±0.41**
5	4.62±3.92	**0.27±0.54**	1.55±0.30	**0.25±0.26**
6	3.94±3.22	**0.41±0.50**	1.56±0.22	**0.35±0.20**
7	2.84±1.51	**0.42±0.43**	1.45±0.38	**0.36±0.21**
8	2.80±1.55	**0.93±1.14**	1.36±0.43	**0.56±0.31**

Table 3.2: 1D Gaussian Mixture Model experiment using best results from 20 runs with different number of discriminators (N) as illustrated in Figure 3.4.

Benchmarking the number of discriminators We study the change in performance with regards to the number of discriminators used by both GMAN [54] and DoPaNet.

GAN Variants	Chi-square($\times 10^6$)	KL-Div
Standard GAN	3.883	2.860
GMAN	1.253	0.636
DoPaNet	**0.449**	**0.246**

Table 3.3: 2D Gaussian Mixture Model experiment with $M = N = 8$ (where N is the number of discriminators, M is the number of Gaussians used in the mixture model). For each experiment, we use a fixed set of 1,000,000 samples and do 5 runs for each algorithm and report the results using the best run. We took effort to make sure that the comparison was fair, and used the same set of parameters as in the 1D experiments.

The clustering mechanism with varying number of discriminators is illustrated on the 1D task in Figure 3.4. We see in this experiment that classes of the generated samples are first attracted towards larger clusters of the real data. By adding more discriminators, the quality of the reconstructed modes is refined. The refinement process starts first with the easiest separation, between the 2^{nd} and the 3^{rd} peaks, after that the 4^{th} and 5^{th} modes are distinguished by the classifier Q, and so on. We quantitatively see in Table 3.2 that increasing the number of discriminators improves the performance of both GMAN and DoPaNet up to a certain point where N (number of discriminators) matches the number of modes in the data. After this optimal point, increasing N yields a decreasing performance, because already captured modes are oversampled. In Figure 3.4 we have marked examples of oversampling in the last two columns with $*$ symbols. It is interesting to note that when the same experiment was carried out in MADGAN [68], which uses multiple generators, their performance peaked at $N_{Generators} = 4$ unlike GMAN and ours, both of which logically peaked at $N_{Discriminator} = 5$ considering that there are 5 visible modes. This shows a difficulty in tuning the hyper-parameter $N_{Generators}$ in [68] for different applications.

2D experiments In 2D experiments, for both GMAN [54] and DoPaNet we experiment with $N = M = 8$ (where N is the number of discriminators, M is the number of modes) for both quantitative (listed in Table 3.3) and qualitative results (see Appendix 3.8), and the $N = M = 3$ setting for qualitative results (illustrated in Figure 3.5). In all of the runs,

DoPaNet was able to capture, and classify all modes of the true distribution correctly, while GMAN [54] failed on both the $N = M = 3$ as well as the $N = M = 8$ setting.

In Figure 3.5 we show a circular 2D GMM with 3 modes on the unit circle which is used to train GMAN and DoPaNets. In the case of DoPaNet, it can also be observed (see column 1) that the generator generates from a different mode for a different c_i. We can also visually see that the classifier Q is indirectly able to control the conditioned samples $G(z, c)$ by routing them to the corresponding discriminators (see columns 2-3). It also illustrates that we are indeed able to utilize the capabilities of multiple D_i's as intended: different discriminators begin to specialize on different modes and therefore provide different gradients for the respective mode as well. Although being trained with the generated code vectors only, DoPaNet's classifier Q achieves fine partitioning of the original distribution. We suggest that our approach succeeds because each discriminator is fed different samples from the beginning. Q is initialized to assign each real sample to every discriminator with equal probability, but given that the generator samples different points for every code vector Q quickly learns the different modes that the samples from G_i are attracted towards (where G_i refers to the conditional distribution modeled by $G(z, c_i)$). Given that the updated Q is already providing different subsets of the input space to the different discriminators, the discriminators will provide different gradients for each corresponding code vector. Therefore G learns to separate the modes of the learnt distributions conditioned on c from each other. We argue that GMAN is not able to utilize multiple discriminators in this experiment setup and that most of the learning is done by just a few discriminators rather than their effective ensemble (see Appendix 3.8).

3.4.2 Image generation

After investigating the DoPaNet performance on low dimensional tasks, now we validate DoPaNet on real image generation tasks.

GAN Variants	KL Div
DCGAN* [173]	2.15
WGAN* [9]	1.02
BEGAN* [17]	1.89
GoGAN* [113]	2.89
Unrolled GAN* [156]	1.29
Mode-Reg DCGAN* [28]	1.79
InfoGAN* [36]	2.75
MA-GAN* [68]	3.4
MAD-GAN* [68]	0.91
GMAN [54]	2.17
DoPaNet (ours)	**0.13**

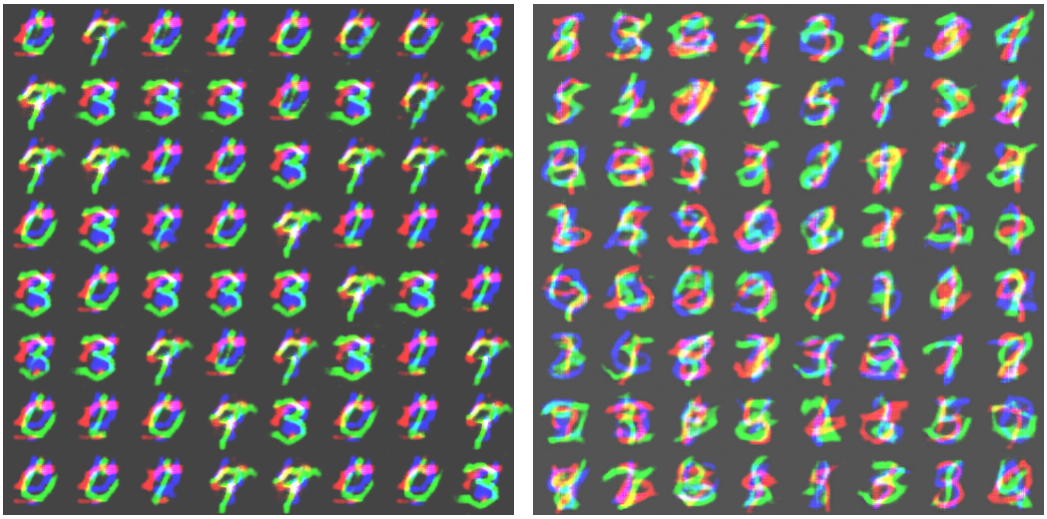
Table 3.4: Stacked-MNIST: we compare our method against several GAN variants. Through this experiment using a real dataset, we can show that DoPaNet is closer to the real distribution. Results for the GAN variants marked as $*$ were reproduced from [68].

Stacked-MNIST

We first investigate how well DoPaNet can reconstruct the real distribution of the data using the Stacked-MNIST dataset [194]. This dataset contains three channel color images, containing a randomly selected sample from the MNIST dataset in each channel. This results in ten possible modes on each channel so the number of all the possible modes in the dataset is 10^3. It was shown in [68] that various architectures recovered only a small portion of these modes. A qualitative image depicting the recovered modes using the traditional DCGAN [173] architecture and DoPaNet can be seen in Figure 3.6. We have also measured the Kullback-Leibler divergence between the real distribution and the generated distributions. We compare DoPaNet against the other GAN variants in Table 3.4.

Qualitative results on CIFAR-10 and celebA

To show the image generation capabilities of DoPaNet, we trained the multi discriminator setting on a lower and a higher complexity image generation task, CIFAR-10 and CelebA respectively. We compare our results qualitatively to the ones reported by GMAN [54] on



(a) (b)

Figure 3.6: Sample images from stacked MNIST. The samples on the left (a) were generated using the traditional architecture of DCGAN while the samples on the right (b) were generated using DoPaNet with 10 discriminators ($N = 10$). Visually, (a) appears to be more clear as only one color (green) is dominating over digits of the other two colors while in (b) more digits are covered per sample which refers to higher diversity in the modeled distribution, thus better mode coverage.

both tasks in Figure 3.9.

Figure 3.7: 32×32 CIFAR-10 samples drawn from DoPaNet trained with $N = 5$ discriminators for 700k iterations. The generations in each subfigure correspond to $y = 1, 2$ and 3 respectively. We call the attention to the various different details learnt for each class which can cause mode collapse in other GAN variants.

CIFAR-10 While learning the distribution of 32×32 colored images may sound easy, the main challenge is to learn geometric structures from low resolution and reproduce them in various colors, backgrounds, angles etc. Following [154], the generator takes ground truth label y as input as well along with the code c, and the discriminator outputs a 10 dimensional output of which only the y^{th} index is used for training D_σ as well as G, while Q is trained just using the code c. Thus, the code c helps it learn class invariant features.

We illustrate in Figure 3.7 that DoPaNet is capable of capturing these features such as different object orientations and colors depicted in various weather conditions. It is also able to recognize minute details like wheels, horse hair, ship textures, etc. We present more generations corresponding to each of the classes in Appendix 3.9.

CelebA We also show DoPaNet performance on large scale images such as 128×128 by training a residual network for 100k iterations on the celebA dataset. This dataset contains various modes like lighting, pose, gender, hair style, clothing, facial expressions which are challenging to capture for generative models. In Figure 3.8 we demonstrate that DoPaNet is capable of recovering the aforementioned visual features.

3.5 Discussion

We conclude that it is not necessary for a generator to have equal capacity adversary to converge, meaning that the standard GAN training procedure could be enhanced with multiple (and even weaker) discriminators specialized only in attracting the model distribution of the generator to their corresponding modes.

DoPaNet is proven experimentally to utilize the capability of multiple discriminators by partitioning the target distributions into several identifiable modes and making each discriminator work on a separate mode. Thus, it reduces the complexity of the modes to be learnt by each discriminator. We show qualitatively and quantitatively that DoPaNet is able to better cover the real distribution. We observe that the generator is also able to sample from different identifiable modes of the data distribution given the corresponding code vectors.

Figure 3.8: Random samples from DoPaNet trained at scale of 128×128 images on unaligned CelebA set for 100k iterations. As it can be seen, different faces appear in diverse poses with different background, and rarely occurring accessories such as orange sunglasses are learned by the model.

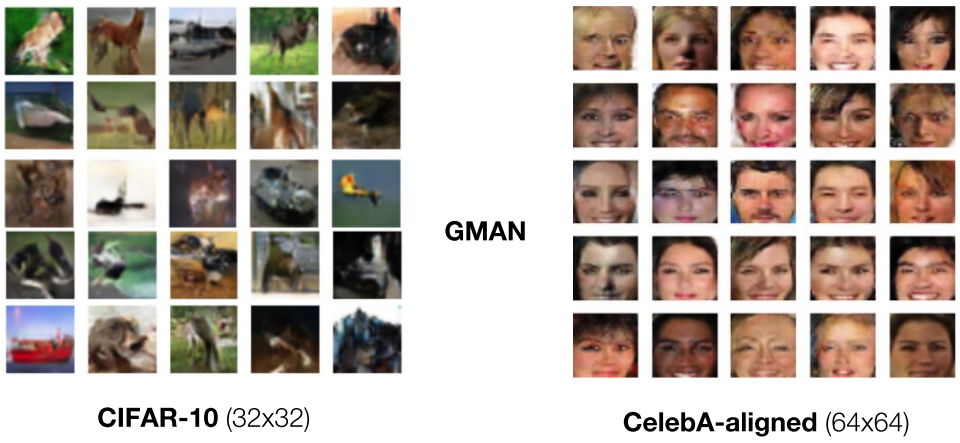


Figure 3.9: Random samples presented in [54] for image generation tasks such as CIFAR-10 and CelebA. Best results were achieved with GMAN-0 variant with $N = 5$ discriminators. For CelebA they cropped the images to exclude background.

3.6 Supplementary Material

Here we first give the theoretical formulation of our work DoPaNet to show that the modes captured should be different for each categorical code c_i and that the standard GAN can be considered as its lower bound on mode collapse. We then give some more experimental insights from the 2D task. We also give some more generations using CIFAR. Later we provide implementation details of the network architectures we used.

3.7 Theoretical formulation

Here we present the theoretical formulation for our proposed method DoPaNet.

Lemma 3.7.1. *For optimal Q and fixed G, the optimal D_i, $\forall i \in [1, \ldots, N]$ is*

$$D_i^*(x) = \frac{\rho_{S_i} p_d^i(x)}{\rho_{S_i} p_d^i(x) + \frac{1}{N} p_g^i(x)} \tag{3.6}$$

where $S_i = \{x \in Supp(p_d) | Q(x) = c_i\}$, $\rho_{S_i} = \int_{x \in S_i} p_d(x) \mathrm{d}x$, p_d^i is a probability distribution such that $p_d^i(x) = \frac{p_d(x)}{\rho_{S_i}}$ and $Supp(p_d^i) = S_i$, and $p_g^i(x) = p_z(z)$ such that $G(z, c_i) = x$.

Proof. Let us consider a case where we have $N = 2$ discriminators. The theoretical formulation for this case can be trivially extended to more number of discriminators. The objective being optimized by the generator and the discriminators is (Obj. 3.3):

$$\begin{aligned} \min_G \max_{D_1, D_2} M(\{D_1, D_2\}, G) := \Bigg[& \mathop{\mathbb{E}}_{\substack{x \sim p_d \\ \sigma \sim Q(x)}} \left[\log D_\sigma(x) \right] \\ & + \mathop{\mathbb{E}}_{\substack{z \sim p_z, c \sim p_c \\ \hat{\sigma} \sim Q(G(z,c))}} \left[\log \left(1 - D_{\hat{\sigma}} \left(G(z, c) \right) \right) \right] \Bigg] \end{aligned} \tag{3.7}$$

When the classifier Q has converged to its optimal form, the above Equation 3.7 can be

rewritten as:

$$\begin{aligned}\min_{G}\Bigg[\max_{D_1}\Bigg[&\mathop{\mathbb{E}}_{\substack{x\sim p_d\\ x\in S_1}}\left[\log D_1\left(x\right)\right]\\ &+p_c(c_1)\mathop{\mathbb{E}}_{z\sim p_z}\left[\log\left(1-D_1\left(G\left(z,c_1\right)\right)\right)\right]\Bigg]\\ +\max_{D_2}\Bigg[&\mathop{\mathbb{E}}_{\substack{x\sim p_d\\ x\in S_2}}\left[\log D_2\left(x\right)\right]\\ &+p_c(c_2)\mathop{\mathbb{E}}_{z\sim p_z}\left[\log\left(1-D_2\left(G\left(z,c_2\right)\right)\right)\right]\Bigg]\Bigg]\end{aligned} \tag{3.8}$$

where $x \in S_i$ if $Q(x) = c_i$. p_c is the categorical distribution and in our case equal probability is assigned to both the values c_1 and c_2. Here c_i is that code vector which leads the classifier Q to pass $G\left(z, c_i\right)$ to D_i for $i \in [1, 2]$. Please note that we can therefore consider $G\left(., c_1\right)$ as $G_1(.)$ and $G\left(., c_2\right)$ as $G_2(.)$, where G_1 and G_2 have shared weights except the bias weights in the initial layer. Bias weights in the initial layer are independently trained for G_1 and G_2.

The Objective 3.8 can be rewritten as:

$$\begin{aligned}\min_{G_1,G_2}\Bigg[\max_{D_1}\Bigg[&\rho_{S_1}\mathop{\mathbb{E}}_{x\sim p_d^1}\left[\log D_1\left(x\right)\right]\\ &+\frac{1}{2}\mathop{\mathbb{E}}_{z\sim p_z}\left[\log\left(1-D_1\left(G_1\left(z\right)\right)\right)\right]\Bigg]\\ +\max_{D_2}\Bigg[&\rho_{S_2}\mathop{\mathbb{E}}_{x\sim p_d^2}\left[\log D_2\left(x\right)\right]\\ &+\frac{1}{2}\mathop{\mathbb{E}}_{z\sim p_z}\left[\log\left(1-D_2\left(G_2\left(z\right)\right)\right)\right]\Bigg]\Bigg]\end{aligned} \tag{3.9}$$

where $\rho_{S_i} = \int_{x\in S_i} p_d(x)\mathrm{d}x$. p_d^i is a probability distribution such that $p_d^i(x) = \frac{p_d(x)}{\rho_{S_i}}$ and $Supp(p_d^i) = S_i$ where $S_i = \{x \in Supp(p_d)|Q(x) = c_i\}$ is the set of samples in the i^{th} mode of the real distribution. So, sampling from p_d^i is same as sampling from the i^{th} mode

of the real distribution p_d. Therefore, $Supp(p_d) = Supp(p_d^1) \cup Supp(p_d^2)$ and $Supp(p_d^1) \cap Supp(p_d^2) = \emptyset$.

For a fixed generator G, G_1 and G_2 are also fixed. For a given G_1 and G_2, the discriminator D_i tries to maximize the quantity (using Objective 3.9):

$$\begin{aligned} &\rho_{S_i} \int_x p_d^i(x) \log D_i(x) \mathrm{d}x \\ &\qquad + \frac{1}{2} \int_z p_z(z) \log(1 - D_i(G_i(z))) \mathrm{d}z \\ &= \int_x \rho_{S_i} p_d^i(x) \log D_i(x) + \frac{1}{2} p_g^i(x) \log(1 - D_i(x)) \mathrm{d}x \end{aligned} \tag{3.10}$$

where $p_g^i(x) = p_z(z)$ such that $G_i(z) = x$ for $i = 1, 2$. Therefore, for a fixed generator we get the optimal discriminator D_i as:

$$D_i^*(x) = \frac{\rho_{S_i} p_d^i(x)}{\rho_{S_i} p_d^i(x) + \frac{1}{2} p_g^i(x)} \tag{3.11}$$

In case of N discriminators, the optimal discriminator D_i can be similarly obtained as:

$$D_i^*(x) = \frac{\rho_{S_i} p_d^i(x)}{\rho_{S_i} p_d^i(x) + \frac{1}{N} p_g^i(x)} \tag{3.12}$$

□

Theorem 3.7.2. *In case of N discriminators, the global minimum of $U(G)$ is achieved if and only if $p_g^i(x) = p_d^i(x)$, $\forall i \in [1, \ldots, N]$. When $\rho_{S_i} = 1/N$, the global minimum value of $U(G)$ is $-\log(4)$.*

Proof. Given the optimal discriminators D_1^* and D_2^*, we can reformulate the Objective 3.9 as:

$$\min_{G_1,G_2}\Bigg[\rho_{S_1}\mathop{\mathbb{E}}_{x\sim p_d^1}\left[\log D_1^*\left(x\right)\right] + \frac{1}{2}\mathop{\mathbb{E}}_{z\sim p_z}\left[\log\left(1-D_1^*\left(G_1\left(z\right)\right)\right)\right] + \rho_{S_2}\mathop{\mathbb{E}}_{x\sim p_d^2}\left[\log D_2^*\left(x\right)\right] + \frac{1}{2}\mathop{\mathbb{E}}_{z\sim p_z}\left[\log\left(1-D_2^*\left(G_2\left(z\right)\right)\right)\right]\Bigg] \tag{3.13}$$

As noted earlier, bias weights in the initial layer of G_1 and G_2 are independently trained with all the other weights shared. As it empirically turns out, the shared weights help learn the similar features, which are essential in low-level image formation and should be similar even if G_1 and G_2 were trained independently. So, we can rather relax the restriction and consider G_1 and G_2 to be independent of each other. So, the objective 3.13 can be rewritten as:

$$\min_{G_1} W(G_1) + \min_{G_2} W(G_2) \tag{3.14}$$

where,

$$W(G_i) := \Bigg[\rho_{S_i}\mathop{\mathbb{E}}_{x\sim p_d^i}\left[\log D_i^*\left(x\right)\right] + \frac{1}{2}\mathop{\mathbb{E}}_{z\sim p_z}\left[\log\left(1-D_i^*\left(G_i\left(z\right)\right)\right)\right]\Bigg] \tag{3.15}$$

This is same as optimizing different G_i-D_i pairs on dataset distributions p_d^i decided by the classifier Q based on the target real distribution p_d. Figure 3.3(b) and 3.3(c) empirically show that the assumption of disjoint support of the distributions p_d^1 and p_d^2 is valid. The Equation 3.15 can be rewritten as:

$$
\begin{aligned}
W(G_i) := & \left[\rho_{S_i} \mathop{\mathbb{E}}_{x \sim p_d^i} \left[\log D_i^* (x) \right] \right. \\
& \left. + \frac{1}{2} \mathop{\mathbb{E}}_{x \sim p_g^i} \left[\log \left(1 - D_i^* (x) \right) \right] \right] \\
= & \left[\rho_{S_i} \mathop{\mathbb{E}}_{x \sim p_d^i} \left[\log \frac{\rho_{S_i} p_d^i(x)}{\rho_{S_i} p_d^i(x) + \frac{1}{N} p_g^i(x)} \right] \right. \\
& \left. + \frac{1}{2} \mathop{\mathbb{E}}_{x \sim p_g^i} \left[\log \left(\frac{\frac{1}{N} p_g^i(x)}{\rho_{S_i} p_d^i(x) + \frac{1}{N} p_g^i(x)} \right) \right] \right]
\end{aligned}
\tag{3.16}
$$

We can further reformulate Equation 3.16 as:

$$
\begin{aligned}
W(G_i) := \rho_{S_i} \Bigg[& - \log(c_1^i) + KL \left(p_d^i \middle\| \frac{p_d^i}{c_1^i} + \frac{p_g^i}{2\rho_{S_i} c_1^i} \right) \\
& + c_2^i \Bigg(- \log(c_3^i) \\
& + KL \left(\frac{p_g^i}{2\rho_{S_i} c_2^i} \middle\| \frac{p_d^i}{c_2^i c_3^i} + \frac{p_g^i}{2\rho_{S_i} c_2^i c_3^i} \right) \Bigg) \Bigg]
\end{aligned}
\tag{3.17}
$$

where KL is the Kullback-Leibler divergence, c_1^i, c_2^i and c_3^i are constants such that $2\rho_{S_i} c_2^i = 1$ ($\implies$ $c_2^i = 1/2\rho_{S_i}$) for $p_g^i / 2\rho_{S_i} c_2^i$ (the first distribution of second KL term) to be a probability distribution. The Kullback-Leibler divergence between two distributions is always non-negative and, zero iff the two distributions are equal. In above equation, the two KL terms are zero simultaneously when $(c_1^i - 1)(c_2^i - 1) = 1$ and the generator distribution is

$$p_g^i = 2(c_1^i - 1)\rho_{S_i} p_d^i$$

where $2(c_1^i - 1)\rho_{S_i} = 1$ ($\implies$ $c_1^i = 1 + 1/2\rho_{S_i}$) for p_g^i to be a probability distribution. Therefore, the global minimum of Eq. 3.17 is achieved iff $p_g^i = p_d^i$. The constants in

Eq. 3.17 are chosen such that:

$$c_1^i = 1 + \frac{1}{2\rho_{S_i}}, \qquad c_2^i = \frac{1}{2\rho_{S_i}}, \qquad c_3^i = 1 + 2\rho_{S_i} \tag{3.18}$$

Please note that when $\rho_{S_i} = 1/2$, the Eq. 3.17 can be reformulated as:

$$W(G_i) := -\log(2) + JSD\left(p_{\text{data}} \| p_g\right) \tag{3.19}$$

and the global minimum of $U(G)$ obtained is $-\log(4)$. This global minimum value is the same in general case for N discriminators when $\rho_{S_i} = 1/N$. □

Corollary 3.7.2.1. *At global minimum of $U(G)$, the generative model G replicates the real distribution p_d, categorized into different modes.*

Proof. As noted in the proof of Lemma 3.1, sampling from p_d^i is same as sampling from the i^{th} mode of the real distribution p_d. At global minimum of $U(G)$, we have $p_g^i = p_d^i$ so $G_i(.) = G(., c_i)$ is able to sample from the i^{th} mode of the real distribution. As the real distribution is categorized into N modes in total and each of $\{G(., c_i)\}_{i=1}^N$ can samples from the corresponding modes, so G can replicate the real distribution p_d, categorized into different modes. □

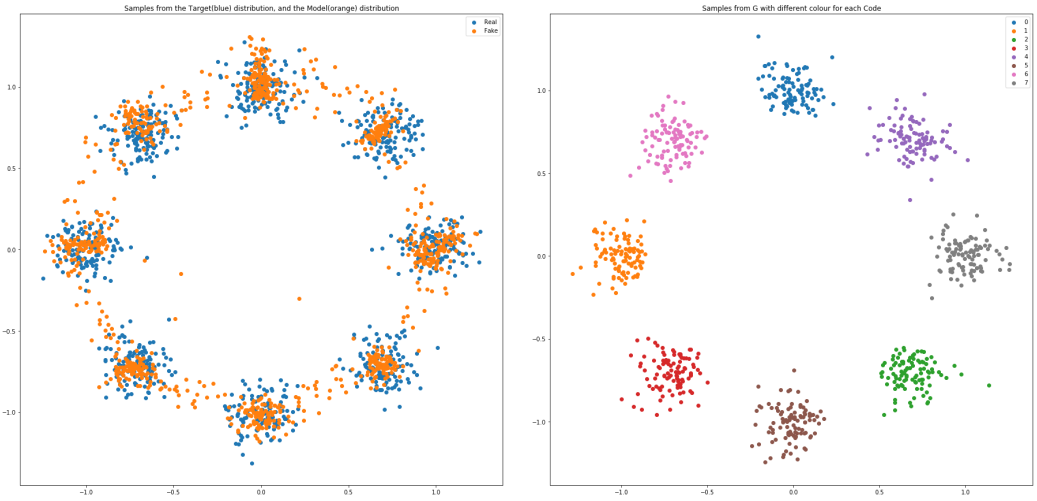
3.8 2D GMM

As discussed in the section 3.4.1, here we show qualitative results with $N = M = 8$ (where N is the number of discriminators, M is the number of modes) whose corresponding quantitative results are mentioned in the Table 3.3. We illustrate our findings in Figure 3.10 and Figure 3.11.

We argue that GMAN is not able to utilize multiple discriminators in this experiment

setup and that most of the learning is done by just a few discriminators rather than their effective ensemble (see Appendix 3.8).

In (4.1.2), under Error Analysis paragraph we claimed that GMAN fails to utilize multiple discriminators to their full potential. In Figure 3.5 we already have visual proof: the gradient field of the first two discriminators (top row, red and blue) are almost identical to each other, while the gradient of the third network (top row, green) is pointing towards a completely different mode (lower left) in its non-adjacent area, while around this distant mode the magnitude of the gradient is relatively small.

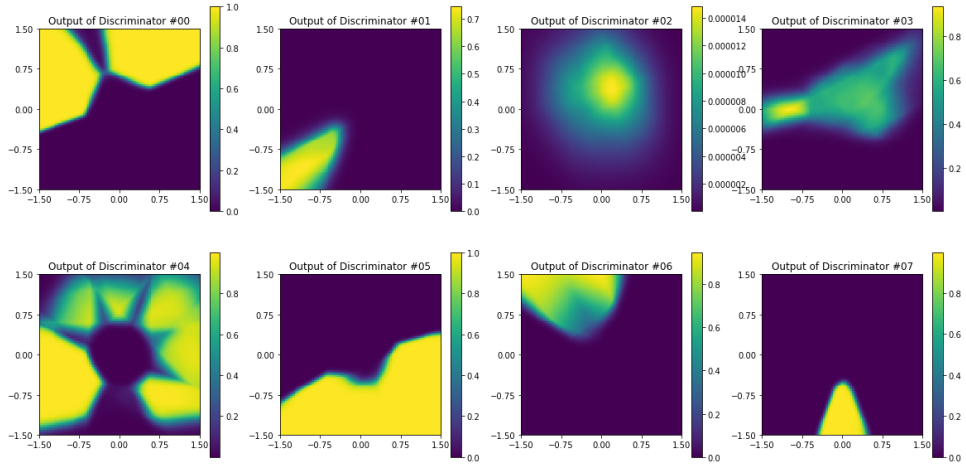


(a) (b)

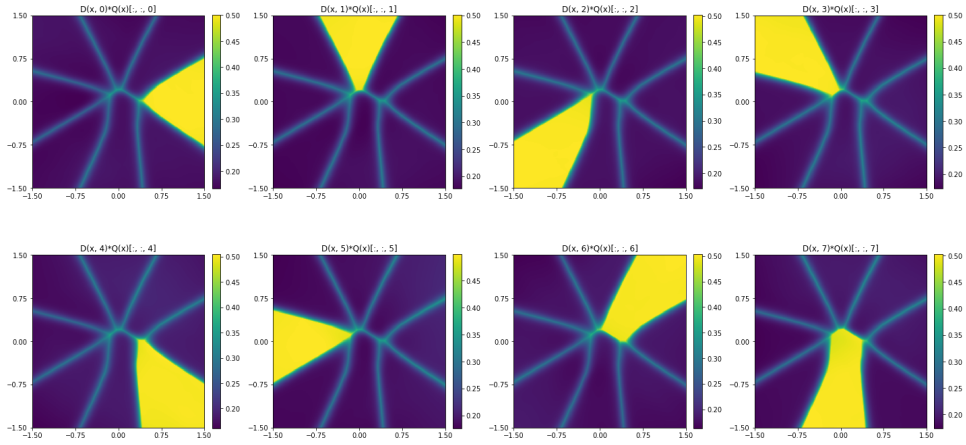
Figure 3.10: (a) GMAN: scatter plot of the data distribution (blue) and model distribution (orange). Although it is covering all the modes, there are a lot of false positives lying between the actual modes. (b) DoPaNet: scatter plot of the model distribution with different colors assigned for samples generated using different code. We sample c from uniform categorical distribution. Notice that the modes are clearly separated from each other as compared to GMAN.

3.9 CIFAR-10

As discussed in the Section 3.4.2, here we present some more results obtained for each of the 10 classes of CIFAR-10 in Figure 3.12 and Figure 3.13.



(a)



(b)

Figure 3.11: Heat map of discriminator scores $\in [0, 1]$ (0 signifies fake while 1 means real) when evaluated for every data-point pair lying in $[-1.5, 1.5]^2$ (corresponding to Figure 3.10). (a) GMAN: it is clear that discriminator #04 already covers the majority of the modes while the other discriminators give high scores for obviously fake samples (#03 and #05). (b) DoPaNet: here we multiplied the discriminator scores with the probability of each point being assigned to that discriminator (obtained from classifier Q). Although the capacity of each discriminator in DoPaNet is identical to the discriminators in the GMAN experiment, the DoPaNet framework reduces the complexity of each discriminator's task by making it work only on a different identifiable mode.

3.10 Implementation details

Here we present the way we structured our experiments and the details about the network architecture we used in the experiments.

3.10.1 Synthetic low dimensional distributions

First, we reproduced the 1D setting in [68] with 5 modes at $[10, 20, 60, 80, 110]$ and standard deviations $[3, 3, 2, 2, 1]$ respectively and we compare to the numbers reported in that paper in Table 3.1. Second, we compared DoPaNet directly to GMAN [54] qualitatively in Figure 3.5 and quantitatively in Table 3.3 using a circular 2D GMM distribution with 3 and 8 modes respecitvely, around the unit circle. To illustrate the advantage of DoPaNet over GMAN [54] we plotted the gradient field to visualize the benefit of using multiple discriminators. The gradient field of this setup can be seen in Figure 3.5. To get quantitative results, we estimated the probability density distribution using a histogram with 1400 bins over the real and generated samples and computed the Chi-square and KL-divergence between the two histograms.

Comparing against other GAN variants When comparing against other GAN variants, we run the 1D experiments using a fixed set of 200,000 samples from the real distribution and generate 65,536 elements from each model.

Since DoPaNet is directly designed to separate different modes, we outperform all the other methods as shown in Table 3.1.

In our case, we sample the code vectors for the generator from a categorical distribution with uniform probability. For the best results, we use 5 discriminators in both GMAN [54] and DoPaNet. For both, we train 3 instances and select the best score from each of them.

Benchmarking the number of discriminators In 1D for better non-parametric probability density estimation, we increased the number of generated samples from 65,536 to 1,000,000 samples as done in [68]. For more reliable results on the implied mechanism of both approaches, we run the training 20 times for each algorithm with number of discriminators $N = 2, \ldots, 8$, totaling 320 training. As in the previous experiment, we chose the

best results from each run.

2D experiments In 2D, for both variants we experiment with $M = N = 8$ (where N is the number of discriminators, M is the number of Gaussians we used in the mixture) for quantitative results, listed in Table 3.3 and $M = N = 3$ setting for qualitative results, illustrated in Figure 3.5. For each experiment we use a fixed set of 1,000,000 samples and take 5 run per each algorithm, then report the best run. We took effort to make sure that the comparison was fair, and used the same set of parameters as it was done in the 1D experiments.

3.10.2 Image generation

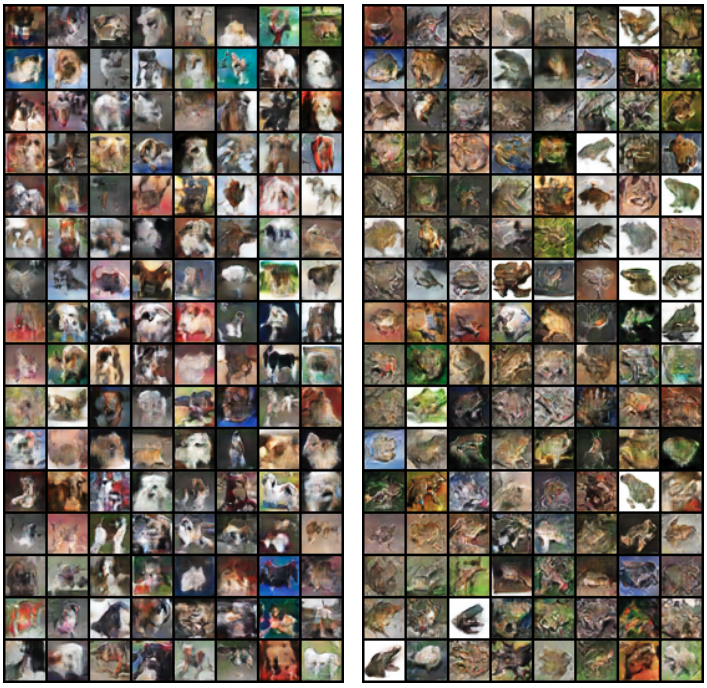
For both the generator and discriminator we use ResNet-architectures [90], with 18 layers each in the CIFAR-10 experiments, and 26 layers each in the CelebA experiments. As was done in [153] we multiply the output of the ResNet blocks with 0.1, use 256-dimensional unit Gaussian distribution. For categorical conditional image generation we use an embedding network that projects category indices to 256 dimensional label vector, normalized to the unit sphere. In the case of conditional image generation the classifier Q is trained on code vectors, so it is constrained to learn the original class labels. We embed the code vector similar to the ground truth labels in this setting for CIFAR-10. We use Leaky-RELU nonlinearities everywhere, without BatchNorm.

Following the considerations in [153] for optimizing parameters of Q, D, G we use the RMSProp with $\alpha = 0.99$, $\epsilon = 10^{-8}$, and initial learning rate of 10^{-3}. We use a batch size of 64, and train the algorithm for 700,000 and 400,000 iterations for CIFAR-10 and CelebA tasks respectively. Similar to work that provided state of the art results on image generation tasks [114, 153] for visualizing the generator's progress we use an exponential moving average of the parameters of G with decay 0.999.

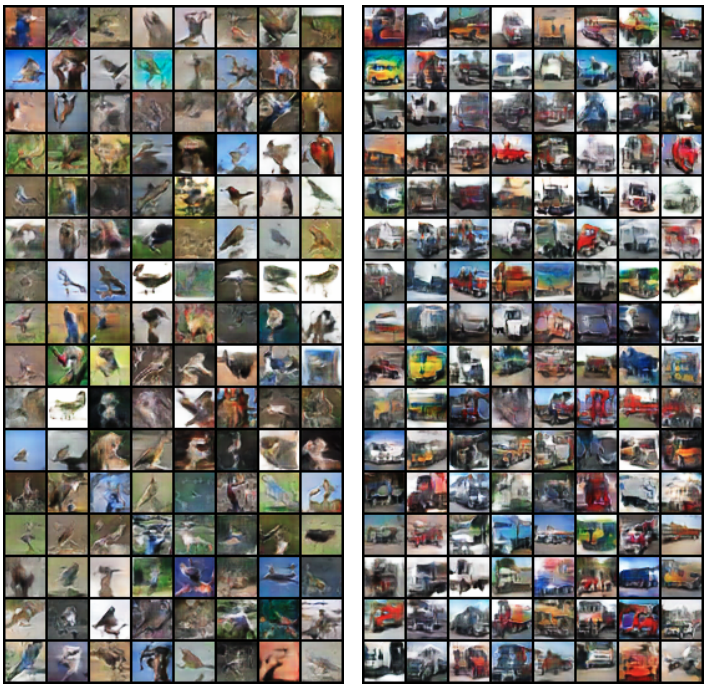
(a) Ship (b) Horse (c) Car

(d) Airplane (e) Cat (f) Deer

Figure 3.12: 32×32 CIFAR-10 samples drawn from DoPaNet trained with $N = 5$ discriminators for 700k iterations. The generations in each subfigure correspond to $y = 1, \ldots, 6$ respectively. We call the attention to the various different details learnt for each class which can cause mode collapse in other GAN variants.



(a) Dog (b) Frog



(c) Bird (d) Truck

Figure 3.13: 32×32 CIFAR-10 samples drawn from DoPaNet trained with $N = 5$ discriminators for 700k iterations. The generations in each subfigure correspond to $y = 7, 8, 9$ and 10 respectively. We call the attention to the various different details learnt for each class which can cause mode collapse in other GAN variants.

3.11 Conclusion

In this chapter we showed the importance underlying structure of the training data in absence of supervision. We proposed a new multi-agent training algorithm that iteratively clustered the training data and assigned each subset of images to its corresponding model-pair. By doing so, we could stabilize the adversarial training procedure and minimize the likelihood of the optimization process recovering a trivial solution, i.e. avoiding the mode-collapse of GANs. In the next chapter, we explore how descriptive vision tasks such as object detection, semantic and instance segmentation can benefit from the optimal partitioning of the underlying training datasets.

Unsupervised Domain Adaptation for Cross-Domain Object Detection

Chapter Teaser

How does it impact the model performance if the samples for evaluation are drawn from a different distribution than the training samples? How can we adapt the model when large quantities of unlabeled samples are available from the test distribution? What differences between distributions can these adaptation methods address? Is there a way to combine multiple adaptation technique in the same training procedure? Does the resulting adapted model have the same computational complexity as the original model?

4.1 Introduction

In the last decade, deep learning has transformed the field of computer vision, becoming the standard approach in the majority of tasks such as classification [228], segmentation [240] and detection [202]. In particular, the features learned by deep networks allow for accurate prediction on object detection tasks in a variety of fields such as medical imaging, astronomy, autonomous driving, *etc.* However, this success comes at a cost of re-

lying on substantial amount of labeled data which may not be available for many practical problems.

Another issue inhibiting the real-world applicability of deep learning based detection algorithms, is their sensitivity to domain shift: when the *target* distribution, from which the test images are sampled, is different than the training *source* distribution. Even the most efficient models suffer severely from this problem [39, 182, 19, 115]. For example, when the weather conditions during test time are different than training time, modern object detectors become unreliable [183, 39]. The naive solution is to collect new training data from the target distribution, and train a new model from scratch. In practice, the high cost of producing object detection labels in particular [167, 195] renders standard supervised detectors infeasible for general use.

To circumvent the limitations posed by insufficient annotation, Unsupervised Domain Adaptation (UDA) techniques combine the labeled source data with unlabeled samples from the target data distribution. Following Chen *et al.* [39], growing number of works [98, 252, 19, 182, 115] utilize UDA and show encouraging results, some even reaching the accuracy of *oracle* models fully trained with labeled target data (see Table 4.1).

In UDA the domain-specific information can be removed at three levels. 1) *Pixel-level alignment* [251, 102], where the low-level variations such as, color shifts, textures, light conditions etc. are removed between the source and target domains. 2) *Domain-invariant representation learning* [39, 91, 252], where the model is trained such that it remains invariant to domain-specific information. 3) *Pseudo-labeling* [19, 115], where a teacher-student network is used to generate additional training data. Existing works [98, 116, 182, 105] either improve one of these levels or combine two of them into a joint model. To the best of our knowledge, none of the existing works jointly study all the three levels for object detection. In this work, we study how each of these three levels progressively remove the domain shift between the source and target domains, and propose a joint

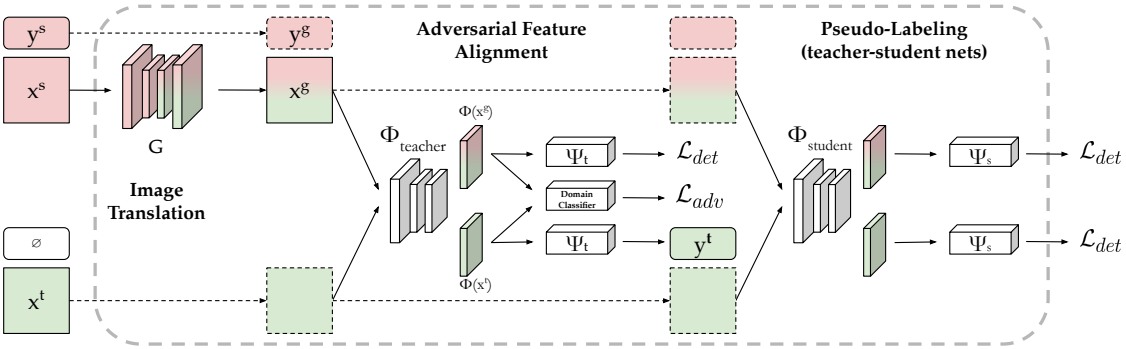


Figure 4.1: Multilevel Knowledge Transfer utilizing all three levels of domain adaptation: 1) pixel-level adaptation 2) adversarial feature alignment and 3) pseudo-labeling. First, the pre-trained image-to-image translation model G converts the labeled source image x^s into a synthetic image x^g that imitates the visual appearance of the images from the target distribution. Under the assumption that the size and location of the objects are left intact by the translation, we reuse (dashed line) the source labels $y^g = y^s$ for the translated image (see Figure 4.2). Second, a teacher network is trained with adversarial feature alignment where the feature extractor Φ_{teacher} learns a canonical representation for the synthetic and the target image. This is achieved by jointly optimizing the original detection loss $\mathcal{L}_{det}$ and adversarially maximizing the cost function of a shallow binary classifier that discriminates features w.r.t their domain. The latter objective ensures that information about the corresponding domain of the input is eliminated from the learned representation space, while the former is necessary to retain useful features required to carry out the detection task by the predictor head Ψ. Lastly, we use the pre-trained teacher network to generate pseudo-labels y^t to the corresponding target image x^t. The student model is trained without feature level alignment, on the mixture of the labeled synthetic images $(\mathcal{X}^g, \mathcal{Y}^g)$ and the pseudo-labeled target images $(\mathcal{X}^t, \mathcal{Y}^t)$.

model utilizing all three. Our method (see Figure 4.1) achieves high accuracy, reaching up to 98% accuracy of the fully-supervised target model, on benchmarks for cross-domain object detection.

Our contributions are as follows: i) We propose a systematic unification of image-to-image translation, feature alignment and pseudo-labeling into a single training procedure. ii) Perform ablation study to better understand how various adaptation techniques complement each other to improve the accuracy under various domain shift scenarios. iii) We evaluate our method on classic unsupervised cross-domain detection settings [39] using the official Faster R-CNN [179] implementation, Detectron2 [224] and achieve an absolute gain of up to +8.71% in accuracy over the current state of the art techniques (see Table 4.1).

4.2 Related work

4.2.1 Object detection

Recent state-of-the-art object detection methods can be categorized into two lines of work. Two-stage detectors that classify and refine region proposals [74, 72, 179, 89] and single-stage detectors [142, 176, 177, 47] that skip the region proposal stage and directly predict bounding boxes and category scores. While the latter focuses on decreasing inference time while maintaining accuracy, the former aims to achieve higher accuracy regardless of the speed. An extensive comparison of the two categories can be found in [101]. For fair comparison with recent methods our work considers Faster-RCNN [179] as baseline, following [39, 105, 19, 98, 252, 182, 116, 115, 91]. [179] employs a classic ImageNet [49] classifier architecture (e.g. VGG [190], ResNet [90], Inception [200]) as a backbone feature extractor. Using the backbone features a Region Proposal Network generates coarse instance predictions that are refined and categorized in the second stage by Region of Interest (ROI) heads. Furthermore, Faster-RCNN [179] can be easily extended for other tasks as well, such as keypoint detection, instance segmentation, 3D pose estimation, *etc.* Our choice of the baseline is also reinforced by the popularity of the reference implementation, Detectron2 by Wu *et al.* [224] which provides a flexible framework and validated evaluation tools, thus facilitates the integration of our method into future works on cross-domain detection. Although our proposal is evaluated using two-stage detectors, the approach can be easily implemented for single-shot detectors as well.

As pointed out by subsequent works [39, 105, 98, 252], classic detection benchmarks [57, 136, 65] only provide limited coverage of real-world challenges, yet top-notch detectors face severe difficulties when the train and test data distribution differs. The field that studies such domain-shift scenarios is often referred to as Domain Adaptation.

4.2.2 Domain Adaptation

In the family of Transfer Learning methods, Domain Adaptation (also known as Transductive Transfer Learning) addresses the different-data, same-task problems, while *e.g.* Inductive Transfer Learning focuses on same-data, different-task problems. A comprehensive overview of DA techniques for visual applications can be found in [46], while [216] lists deep neural net adaptation methods. Our study focuses on Unsupervised Domain Adaptation (UDA) that considers three models: 1) a baseline model trained on supervised source data without adaptation, 2) a model trained on supervised source data and adapted to unsupervised target data, 3) an oracle trained on supervised target data. In each case, the accuracy is measured on the target. The goal is to improve on the baseline accuracy by finding a way to utilize unlabeled target input samples.

While the vast majority of adaptation techniques is mainly focusing on image classification [166, 52, 53, 58, 62, 63, 66, 78, 82, 119, 127, 144, 145, 159, 196], recently semantic segmentation [95, 244, 38, 40, 206, 245, 97] and object detection [102, 39, 105, 19, 98, 252, 182, 116, 115, 91] has gained more traction. Conventional adaptation methods include Multiple Kernel Learning (MKL) [77], adaptive-MKL [53], domain transfer-MKL [52], geodesic flow kernel [78], deformable part based models [230, 231], asymmetric metric learning [119, 232], co-variance matrix lignment [216], alignment of second order statistics [196] and subspace alignment [175, 58]. In the advent of deep learning, adaptation techniques were refocused on achieving domain-invariant representation during training. In our work, we categorize these techniques into three classes based on which point of the data-flow do they take effect on, namely: image, feature and output level adaptation methods.

Image level.

Formerly known as reweighting algorithms [188, 100], image-level adaptation refers to techniques that transform the labeled source samples to appear as target samples while

retaining their semantic content such that the source labels will be still meaningful after the transformation. Such transformation can be learned in an unsupervised fashion using models inspired by pix2pix [108] and Coupled-GAN [141], such as CycleGAN [251], UNIT [140] and MUNIT [103]. Furthermore, there are image translation algorithms specifically tailored for cross-domain adaptation, AugGAN [102] and SPLAT [208].

Feature level.

The goal of feature-level adaptation is to encourage the feature extractor to preserve discriminative features and to discourage learning domain-specific representations. In [144, 145, 197, 22] the mean embedding of the source and target distributions are matched using the Maximum Mean Discrepancy (MMD) metric. In another line of work [62, 63, 209] employs adversarial training from [80] to approximate the $\mathcal{H}$-divergence [39] by training a domain classifier network on the learned representations and trains the feature extractor to maximize the error of the classifier.

Output level.

Compared to the image level adaptation where one approximates the target *image* distribution, methods that approximate the target *label* distribution are considered output level adaptation methods. In our work we consider a trivial solution for doing so, which is the standard teacher-student setup [185] consisting of two stages: training a teacher network on the source dataset and reuse it to generate pseudo-labels on the target images, then training a student model on the union of labeled and pseudo-labeled data. Subsequent works following this setup are more commonly studied in the field of semi-supervised learning [236, 121, 203, 105, 228] as well as in Knowledge Distillation [93] and Pseudo-Labeling [124, 107, 187]. Sun *et al.* [198] suggests a major overlap between the field of domain adaptation and self-supervised learning. Similarly to image and feature level adaptation, [206, 207] approaches output level adaptation with adversarial training.

Most of these studies restrict their experiments to image classification or semantic segmentation, however, the findings of [115, 105] show encouraging results in cross-domain object detection.

4.2.3 Cross-Domain Object Detection

Approaches prior to [39] include deformable part-based models [230, 231] and subspace alignment methods [58, 175]. These solutions were limited to specific cases while more general real-world challenges remained to be unsolved.

Chen *et al.* [39] suggested three realistic experimental settings in the context of autonomous driving: synthetic-to-real transfer, changing weather conditions and different recording devices. The source and target datasets are respectively [111, 44], [44, 183] and [65, 44]. Works after [39] follow these settings to measure the efficiency of their proposed adaptation procedure.

DA-Faster-RCNN [39] adopts adversarial feature alignment [62] for two-stage object detectors: a domain discriminator is trained on the final stage of the backbone and another discriminator on the box predictor head while the feature extractors are trained to confuse the domain classifiers.

Multi-Adversarial Faster-RCNN (MAF) [91] extends [39] to align multiple stages of the backbone feature extractor.

Pseudo-Labeling for object detection is studied in [105], which also employs CycleGAN [251] to perform image-level adaptation as a preprocessing step.

MTOR [19] adopts the Mean Teacher paradigm [203] for object detection by integrating Object Relations into the consistency cost between student and teacher models.

Selective Cross-Domain Alignment (SCDA) [252] and Strong-Weak Distribution alignment (SWDA) [182] argues that previous approaches were focusing on ensuring strict alignment

on the global representation, despite it can be harmful to tasks heavily relying on spatial information. SCDA [252] breaks down the problem of cross-domain object detection to two sub-problems: *"where to look"* and *"how to align"*, while SWDA [182] implements a different strategy for soft alignment of low-level local features and strict alignment of high-level global features. To further improve results, [182] uses image-level adaptation in addition to feature alignment in some of their experiments.

Inspired by [82], Progressive Domain Adaptation [98] suggests combining image and feature level adaptation in two stages: first adapting a model between source to the intermediate domain constructed by image translation, followed by adaptation between intermediate and target domain.

Diversify and Match (DM) [116] proposes an alternative approach to integrate image and feature level adaptation by training multiple translation models with different objective functions to map the source domain into various distinctive intermediate domains and trains the detector using Multi-domain-invariant Representation Learning (MRL), a generalization of [63].

Robust R-CNN [115] studies how can one deal effectively with noisy labels to better utilize pseudo-labels generated by the teacher network. Their approach augments the teacher-student setup with an external image classifier trained on the source data using hard labels and on the pseudo-labeled target using soft labels. The classifier is then used to filter the teacher's predictions and provide soft-labels for training the student model on the target data with the refined pseudo-labels.

The core limitation of approaches that perform only image [102], feature [39, 252, 91] or output [19, 115] level adaptation is their narrow focus on improving existing solutions in one particular direction, not considering the merits of other paradigms. On the other hand, [98, 182, 116] successfully combine image-and-feature level or image-and-output level adaptation techniques [105], indicating that adaptation methods of different levels com-

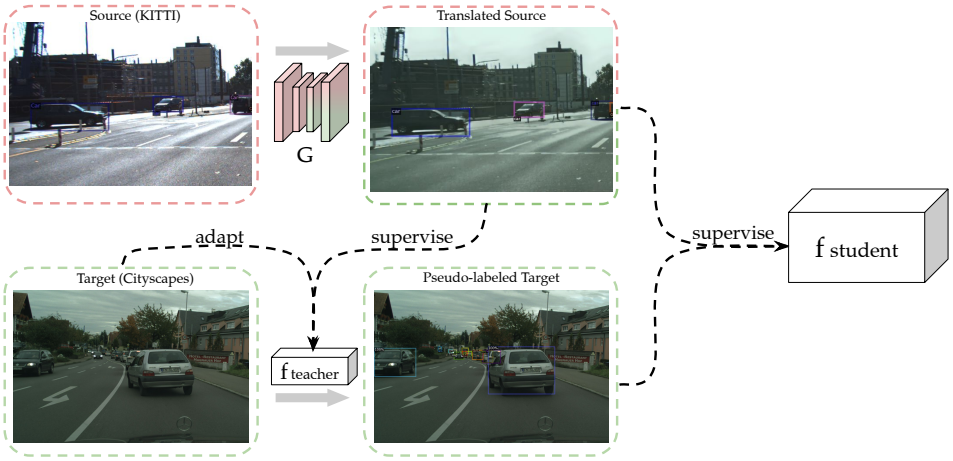


Figure 4.2: Illustration of our method on the *cross-camera* domain adaptation scenario. The labeled source data (Cityscpaes [44]) is transformed by a pre-trained G to make it appear as if it was a sample from the target domain (KITTI [65]). We enforce G to learn low level visual transformations, thus only textural differences are adopted, whereas the geometric properties of the underlying objects remain intact, therefore the original source labels can be reused. The translated source data is then used to provide supervised training signal to the teacher model $f_{teacher}$, meanwhile the unlabeled target data is used in the domain adversarial adaptation of $f_{teacher}$. Pseudo-labels are provided for the target data by $f_{teacher}$, which in turn is used in combination with the translated source data to train the final model, $f_{student}$. For full details of the process see Figure 4.1.

plement each other. From this point of view, one question arises naturally: why not utilize all three levels of adaptation? In this paper, we propose a method to integrate a simple technique from each level and demonstrate its efficiency, superior to more sophisticated methods with a narrower focus.

4.3 Multilevel Knowledge Transfer

We now begin the exposition of our method. The goal in supervised object detection is to learn a function $f_\theta = \Psi \circ \Phi : \mathcal{X}^t \to \mathcal{Y}^t$, parametrised by θ (a neural network in our case), mapping an input image to the bounding box coordinates and labels of all classes present in the dataset of interest (defined as target data in this work). In many real-world setups it is difficult to collect a labeled dataset for this task. Unsupervised Domain Adaptation (UDA) attempts to use data from another domain, referred to as source domain $(\mathcal{X}^s, \mathcal{Y}^s)$,

where labeled data is readily available and combines it with unlabeled target data to improve final performance on the target domain test set. One way to achieve this in UDA is to gradually remove the domain shift between the source and the target domain, so that both labeled $(\mathcal{X}^s, \mathcal{Y}^s)$ and unlabeled $\mathcal{X}^t$ can be leveraged for the better overall performance on the target domain.

Our proposed model, Multilevel Knowledge Transfer (see Figure 4.1) can be summarized in three steps: 1) *Translate* the supervised source images into the target domain. 2) Train a model on the translated images using the source labels with additional feature alignment constraint between the translated source and the target domain. 3) Generate pseudo-labels for the target images from the model obtained in the previous step to train a new model on the combined dataset of translated source dataset and the pseudo-labeled target dataset. Below we describe each step in detail.

4.3.1 Image-to-Image translation

In image level adaptation we remove low-level domain specific variations of the input space, such as: color shifts, textures, light conditions, reflections etc. For this, we transform the labeled source images $\mathcal{X}^s$ to a synthetic dataset $\mathcal{X}^g$ that approximates the distribution of the target data $\mathcal{X}^t$ by removing the domain shift between the two distributions. We do so by training an image-to-image translation model, Multimodal Unsupervised Image Translation (MUNIT) [103], that requires unlabeled input images from both source and target domains to learn the translation $G : \mathcal{X}^s \rightarrow \mathcal{X}^g$. One notable advantage of MUNIT is its ability to transform a single source image to multiple different synthetic images with identical content but different appearances. Therefore, in addition to reducing the domain shift between the two distributions, MUNIT captures diversity in the target domain. This effectively increases the size of the training set by many folds. We assume that MUNIT only modifies the appearance, not the geometry, of the original object. The bounding-box class, size and location does not change, therefore, source labels can be reused without

modification for the translated images: $\mathcal{Y}^g = \mathcal{Y}^s$. Prior work [98, 182, 116] uses CycleGAN [251] for the same purpose. However, CycleGAN does not guarantee multi-modality resulting in reduced robustness of the model as show in Section 4.4.7.

4.3.2 Feature level adaptation

With image-to-image translation covering the dataset aspect of training, we now turn our attention towards the model itself. To increase robustness, we aim to train the model such that learned features ignore domain-specific information and only encode domain-invariant discriminative information. For this, we train a model, referred to as the *teacher*, that *aligns* the features between the target ($\mathcal{X}^t$) and the translated source images ($\mathcal{X}^g$) (obtained in the previous step). We use adversarial training for this alignment. More specifically, the teacher model f_t consists of a feature extractor Φ_t and a predictor Ψ_t, making the predictions $f_t(x) = \Psi_t \circ \Phi_t(x)$. Similarly, we define the student model $f_s(x) = \Psi_s \circ \Phi_s(x)$. In addition to Ψ_t, we train a *Domain Classifier* (D) on top of Φ_t that outputs 1 when the input is in the target domain and 0 otherwise by minimizing the following loss: [1]

$$\begin{aligned} \mathcal{L}_{adv} = \; & \mathbb{E}_{x \sim p(x^g)} \left[\log D(\Phi_t(x)) \right] + \\ & \mathbb{E}_{x \sim p(x^t)} \left[\log \left(1 - D(\Phi_t(x)) \right) \right]. \end{aligned} \tag{4.1}$$

Note that maximizing $\mathcal{L}_{adv}$ over the parameters of Φ eliminates domain-specific information in the learned feature space, as it maximizes the error of D. Using GRL [63], we simultaneously minimize $\mathcal{L}$ over the parameters of D and maximize it over the parameters

[1] Please note, that in our notation t as a superscript means *target* (such as $\mathcal{X}^t$), whereas as a subscript it refers to the *teacher* model (e.g. f_t)

of Φ. Finally, the overall training objective $\mathcal{L}_{teacher}$ is given by:

$$\mathcal{L}_{teacher} = \sum_{(\mathcal{X}^g, \mathcal{Y}^g)} \mathcal{L}_{det}(f_t(x), y) - \mathcal{L}_{adv} \tag{4.2}$$

where $\mathcal{L}_{det}$ is the standard detection loss [179].

4.3.3 Teacher-student training

While the domain-invariant model trained in the previous step can be used on the target test set, one can further refine the model by generating *pseudo-labels* $\mathcal{Y}^t$ for the target images and then training a student model on the joint dataset $(\mathcal{X}^g \cup \mathcal{X}^t, \mathcal{Y}^g \cup \mathcal{Y}^t)$. We obtain pseudo-labels by applying the teacher model f_t, explained in the previous step, on unlabeled target images, and hard-thresholding the outputs at 0.5 *objectness* score. Experimentally, we found hard-thresholding to be more effective than soft-labeling (see Sec. 4.4.6 for details). Note, contrary to prior work, such as Noisy Labeling [115] that uses filtering based on a separate network (43M additional parameters trained for 300k iterations), our method is simple and efficient. In addition, in contrast to [105], we do not iterate over multiple teacher-student trainings. Once the pseudo-labels from the teacher network are available, the training over the joint dataset $\mathcal{D} = (\{\mathcal{X}^g \cup \mathcal{X}^t\}, \{\mathcal{Y}^g \cup \mathcal{Y}^t\})$ proceed as per the following objective:

$$\mathcal{L}_{student} = \sum_{x,y \in \mathcal{D}} \mathcal{L}_{det}(f_s(x), y) \tag{4.3}$$

An important design choice is that we omit feature level adaptation for the student training. The reason behind excluding adversarial feature level adaptation is that we have found empirically the supervised training on pseudo-labeled target images sufficient and more stable w.r.t. convergence rate and target accuracy.

4.4 Experiments

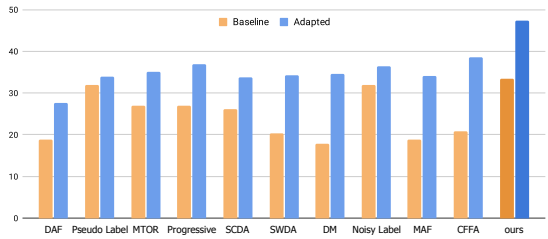


Figure 4.3: Cityscapes→Foggy Cityscapes mAP_{50} scores. The bars on the left are the baseline scores reported in prior methods. Apart from illustrating the superiority of our approach, we use this figure to illustrate the sensitivity of the mAP_{50} metric w.r.t. the underlying implementation.

4.4.1 Implementation Details

We adopt ResNet-50 [90] as our backbone network and follow [39] to set the hyperparameters including training iterations, learning rate policy, batch size and number of proposals in a batch. Our implementation is based on Faster-RCNN [179] in Detectron2 [224]. For every experiment we resize the input images such that their shorter side has a length of 600 pixels. Unless stated otherwise, no data augmentation is used. Mini-batch of size 2, consisting of one source image with corresponding label and one target image without label, is used for feature level adaptation [39], and mini-batch of size 1 consisting of a single labeled source image is used when feature level adaptation is not used. The rest of the hyper-parameters are adapted from [39]. The backbone network is initialized with parameters pre-trained on ImageNet [49] and the detection model is trained for 50k iterations using SGD with momentum of 0.9, learning rate of 0.001 and weight decay of 0.0005. The learning rate is then reduced to 0.0001 and the network is further trained for 20k iterations.

The scores are reported on the validation set of the target domain, after training the model for 70k iterations. We report the mean average precision (mAP) with a threshold of 0.5 using the generalized COCO evaluation tool in [224]. To better evaluate the efficiency

of various adaptation techniques, we also propose a new metric that represents to what extent the performance-gap (between source-supervised *baseline* and target-supervised *oracle* models) has been covered by the *adapted* model:

$$coverage = \frac{\text{mAP}(adapted) - \text{mAP}(baseline)}{\text{mAP}(oracle) - \text{mAP}(baseline)}$$

4.4.2 Datasets

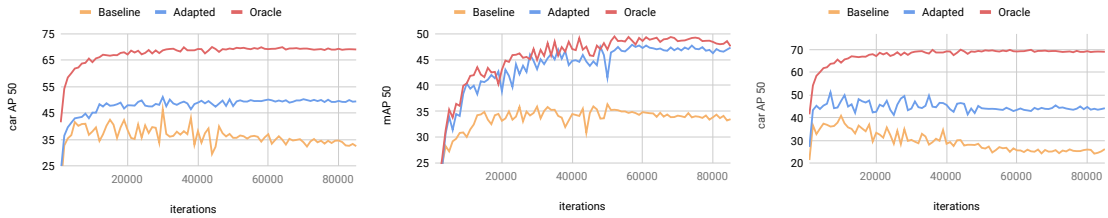
Cityscapes [44] is a dataset for autonomous driving experiments in urban settings where images were captured with an on-board device. SIM 10K [111] has 10,000 synthetic images captured from the video game *Grand Theft Auto V*. Foggy Cityscapes [183] is an artificial dataset to measure model performance in adverse weather conditions by adding synthetic fog to the real images in Cityscapes [44], which contains three different scenarios with decreasing level of objects visibility. Finally, KITTI [65] provides a dataset for autonomous driving in diverse real-world traffic scenarios ranging from rural areas to inner-city environments. Please refer to the Sec. 4.4.5 for further details on the datasets.

We follow the cross-domain object detection setups introduced by Chen *et al.* [39]. Namely, we evaluate our method in **sim2real** (SIM 10K [111] → Cityscapes [44]), **changing weather** (Cityscapes [44] → Foggy Cityscapes [183]) and **cross-camera** adaptation (KITTI [65] → Cityscapes [44]) settings.

4.4.3 Comparison with State-Of-The-Art

Transfer from Simulation to Real World (sim2real).

In this setting we evaluate how well can knowledge from synthetic data be utilized in a real world setting. The importance of such experimental setting is relevant in scenarios where annotating real data with humans is expensive compared to the cost of 1) generating diverse labeled synthetic images from the simulator and 2) acquiring new input images



(a) SIM 10K→Cityscapes (b) Cityscapes→Foggy (c) KITTI→Cityscapes

Figure 4.4: AP_{50} evaluated on the validation split of the target dataset during the training every 1000 iterations for a total 85k iterations. Baseline is a model trained on source only, while the oracle is trained on target only.

Method	*sim2real*		*adverse weather*		*cross-camera*	
baseline [179] (ours)	34.91	-	33.48	-	25.11	-
DA-Faster-RCNN [39]	38.97	+8.85	27.60	+8.8	38.50	+8.30
Pseudo-Labeling [105]	39.05	+7.97	33.90	+2.0	40.23	+9.13
MTOR [19]	46.60	+7.20	35.10	+8.2	-	-
PDA [98]	-	-	36.90	+10.0	43.90	+5.70
SCDA [252]	43.02	+9.06	33.80	+7.6	42.50	+5.10
SWDA [182]	41.50	+6.90	34.30	+14.0	-	-
DM [116]	-	-	34.60	+16.7	-	-
Noisy Labeling [115]	42.56	+11.48	36.45	+4.5	42.98	+11.88
MAF [91]	41.10	+11.00	34.00	+15.2	41.00	+10.80
CFFA [249]	43.80	+8.80	38.60	**+17.8**	-	-
ours	**50.22**	**+15.31**	**47.31**	+13.8	**44.25**	**+19.14**
oracle (ours)	69.03	+34.12	47.56	+14.1	69.03	+34.12

Table 4.1: Quantitative analysis of the adaptation efficiency measured in AP_{50} and relative improvement w.r.t. the reported baseline. Underlined scores mark higher relative improvements than the *oracle* (a model trained entirely on labeled target data), which is due to diminishing returns.

from the target domain. For training we use the entire SIM 10K [111] dataset as the source domain $(\mathcal{X}^s, \mathcal{Y}^s)$, while we use the training split of Cityscapes [44] as the target domain unlabeled images $\mathcal{X}^t$. We evaluate the final model performance on the validation split of Cityscapes. Since only the car instances exists in SIM 10k, we report the *car* AP_{50}.

Results are shown in the *sim2real* column of Table 4.1, where we compare our performance against previous methods. As the reported baseline performance (Faster-RCNN [179] trained on source) in several existing works differ significantly, we report the total accuracy and the relative improvement with regards to **the baseline accuracy reported in each paper** for fair comparison. Our method outperforms the previous state-of-the-art

Experiment	Method	Baseline	Adapted	Oracle	Coverage
sim2real	MTOR [19]	39.40	46.60	58.60	37.50%
	ours	34.91	50.22	69.03	**44.87%** (+7.37%)
adverse weather	CFFA [249]	20.80	38.60	43.30	79.11%
	ours	33.48	47.31	47.56	**98.22%** (+19.11%)
cross-camera	PDA [98]	38.20	43.90	55.80	32.39%
	ours	25.11	44.25	69.03	**43.58%** (+11.19%)

Table 4.2: Supervised Accuracy Coverage comparison to state-of-the-art methods in various domain adaptation scenarios

approach MTOR [19] by **+3.6%**, and their reported relative improvement (**7.2%**) with regards to their baseline is much smaller than ours (**15.31%**). From the perspective of relative improvement our method is also superior to other adaptation techniques: [115] reports a relative improvement of +11.48% while our method has **+15.31%** relative improvement. The high relative improvement manifests that our accuracy improvements stems from the proposed multilevel knowledge translation approach instead of a stronger baseline model. In Figure 4.4a we show that our technique *covers* more than **44%** of the performance gap, meanwhile the baseline begins to over-fit the source domain after 50k iterations.

Adaptation to Adverse Weather (changing weather). In this setting, we measure how changing weather scenarios impact the accuracy of a model trained on a dataset under good weather conditions. We use labeled images from the training set of Cityscapes [44] as source $(\mathcal{X}^s, \mathcal{Y}^s)$, and adapt our model to unlabeled images from the training split of Foggy Cityscapes [183] $(\mathcal{X}^t)$. We evaluate AP_{50} for all 8 categories in Foggy Cityscapes since it has compatible categories with Cityscapes. The model trained after 70k iterations is utilized for evaluation.

In the *adverse weather* column of Table 4.1, we show that our method outperforms all prior work in terms of mAP_{50}. In terms of mAP_{50}, the previous state-of-the-art – CFFA [249] – has achieved 36.9% mAP, while our results show a significant **+8.71%** improvement. In terms of relative improvement w.r.t. reported baseline, we again outperform previous works with +13.8% except MAF [91] and CFFA [249] that used a weaker baseline compared to ours. We argue that improvement upon a strong baseline is more convincing. To

provide further evidence for the difficulty of improving on stronger baselines we have highlighted in the last row of Table 4.8 that even the oracle's relative improvement could not exceed MAF [91]. In Figure 4.3, we demonstrate that our baseline performance exceeds DAF [39] and it is on-par with the accuracy of MAF [91] after adaptation. To illustrate the effectiveness of our approach, we plot the validation accuracy in Figure 4.4b, which shows the *coverage* of the performance gap between the baseline and the oracle is above **98%**. A detailed class-specific comparison of AP scores can be found in Sec. 4.4.9.

Cross-camera adaptation (cross-camera). Finally, we show results under a setting where both the source and the target dataset is real, however the input images were captured in different lighting conditions in various environments with different cameras. We use the entire KITTI [65] train set as the source domain $(\mathcal{X}^s, \mathcal{Y}^s)$, and the unlabeled images of the train set of Cityscapes [44] as the target domain $(\mathcal{X}^t)$. Following prior studies we report *car* AP_{50} scores on the validation set of Cityscapes after 70k iterations.

Our results on cross-camera adaptation are summarized in the *cross-camera* column of Table 4.1. We again outperform all compared approaches and achieve the highest relative improvement **+19.14%**. In Figure 4.4c we show that Multilevel Knowledge Transfer *covers* more than **43%** of the performance gap, which manifests the effectiveness of the approach in closing the performance gap caused by domain shifts.

4.4.4 Ablation studies and analysis

In the following, we will discuss the effectiveness of different componenents in our model and how they collaborate to achieve the state-of-the-art performance. To determine which domain adaptation component plays the most significant role in improving the target domain accuracy in each experimental setting, we have trained different detection models as below. First we trained the baseline model using the labeled source data only. Next, we train models with different combinations of the studied three levels of domain adaptation

(image, feature, output). The detailed results and comparisons are shown in Table 4.3.

IMG	FEA	OUT	*sim2real*	*adverse weather*	*cross-camera*
			34.91	34.16	25.11
		✓	36.78	37.79	25.58
	✓		43.55	43.83	42.00
	✓	✓	48.81	43.71	42.95
✓			47.73	44.33	40.23
✓		✓	50.00	46.09	41.81
✓	✓		43.14	**47.37**	43.74
✓	✓	✓	**50.22**	**47.31**	**44.25**
	oracle		69.03	47.56	69.03

Table 4.3: Target accuracy (mAP_{50}) reported after 70k training iterations under experimental settings described in 4.4.3.

FID	*sim2real*	*adverse weather*	*cross-camera*
source ↔ target	102.86	40.49	81.01
translated ↔ target	61.88	36.94	75.06

Table 4.4: FID [92] between training image distributions.

It is important to notice that the contribution of each level of adaptation differs in different experimental settings and exhibits inconsistent behaviours. We attribute these inconsistencies to the different domain shifts imposed by different settings. In Table 4.4, we report the Fréchet Inception Distances (FID) [92] between the input distributions in each experimental setting. The first row lists the distances between the distribution of original source and target images. In the second row, we list the distances between the distribution of translated source images and the target images.

Single component analysis. Image level adpatation boosts the performance more than feature level adaptation when the domain discrepancy between source and target is large – *sim2real* (+12.82% vs. +8.64%) and they yield similar performance improvement when the domain dicrepancy is small – *changing weather* (+10.17% vs. +9.67%) and *cross-camera* (+15.12% vs. +16.89%). In constrast, the output level domain adaptation alone doesn't benefit the performance a lot influenced by the quality of the teacher network.

Component combination analysis. We observe combining two components does not always bring performance improvement in comparison to the single component baseline. In the *sim2real* scenario, the performance drops to 43.14% (similar to feature-level only)

when image-level and feature-level adaptation are adopted together. This is potentially caused by imperfect translation results from image to image translation which is the most difficult in *sim2real* scenario. Output-level adaptation consistently improves the performance when combined with other level of domain adaptation approaches in all settings. Overall, our proposed framework incorporating all three levels of adaptation achieves the best performance.

4.4.5 Dataset statistics

Cityscapes

Cityscapes [44] is a dataset for autonomous driving experiments in urban settings where images were captured with an on-board device. It contains a train and validation split with 2,975 and 500 images, respectively. The training set contains 52088 instances, namely *person* (17,910), *rider* (1,778), *car* (26,957), *truck* (484), *bus* (380), *train* (168), *motorcycle* (737), *bicycle* (3,674). Since the dataset main purpose is to provide a benchmark for semantic and instance segmentation we cannot directly use its labels nor the corresponding evaluation tool-kit for measuring object-detection, therefore we generate tight bounding boxes for instances following prior approaches and use the COCO [136] AP evaluation tool implemented in Detectron2 [224]. Under settings where Cityscapes is used as *source* we use the training split. Under settings where Cityscapes is *target* we use the training split, but without labels. For evaluation we always use the validation set.

SIM 10K.

SIM 10K [111] has 10,000 synthetic images captured from the video game *Grand Theft Auto V*. Bounding boxes of 58,701 instances are provided in PASCAL VOC [57] format of *car* (57,776), *person* (4), *motorbike* (921) categories. Previous papers report using 58,701 car instances, however only 57,776 instances of the provided instances belong to the *car* category. Also, according to the standard Faster-RCNN [179] settings, we do not use

images without instances, which effectively reduces the size of the training set to 9975 images. During training we use the whole dataset and do not use an arbitrary validation split for hyperparameter tuning.

Foggy Cityscapes.

Foggy Cityscapes [183] is an artificial dataset created to measure performance of models in adverse weather conditions by adding synthetic fog to the real images in Cityscapes [44]. This makes compatible to compare category AP scores of models trained on Cityscapes. The dataset contains three different scenarios with decreasing level of visibility of objects. We use all three levels of visibility both for training and validation, effectively multiplying the available number of images and annotations by 3. Under settings where Foggy Cityscapes is *target* we use the training split without labels and for evaluation we use the validation set.

KITTI.

KITTI [65] provides a dataset for autonomous driving in diverse real-world traffic scenarios ranging from rural areas to inner-city environments. The dataset contains 7481 images with 51,865 instances in total of 9 categories, namely *car* (28,742), *van* (2,914), *truck* (1,094), *pedestrian* (4,487), *person sitting* (222), *cyclist* (1,627), *tram* (511), *miscallenous* (973), *don't care* (11,295). For the sake of simplicity we discard *don't care* regions from both training evaluation. Both under settings where KITTI is used as *source* and *target* we use the whole dataset and do not use an arbitrary split for acquiring a validation set to tune hyperparameters.

4.4.6 Hard vs. Soft Labels

We compare the effect of treating teacher predictions as hard-labels versus soft-labels (as introduced in [93]) and conclude that treating y^p as hard-labels is favorable in terms

Temperature	student source mAP	student target mAP
0.5	42.66	40.14
1.0	45.01	43.19
1.1	45.21	42.82
2.0	48.49	45.02
5.0	48.93	45.99
10.0	49.00	45.99
20.0	50.51	47.20
hard-label	**51.23**	**47.30**

Table 4.5: Quantitative comparison of treating pseudo-labels $\mathcal{Y}^p$ as ground truth. For the hard-label experiment $\alpha = 1$, otherwise $\alpha = \frac{1}{2}$.

of target accuracy. In our implementation only the classification head's cross-entropy objective function is modified as follows:

$$\sum_i \alpha p_i \log q_i + (1-\alpha)\hat{p}_i \log q_i$$

where p_i is the hard-label and $\hat{p}_i$ is the soft-label with temperature parameter T, defined by:

$$\hat{p}_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

We observe that in settings where $\alpha = 0$, the training of the student model becomes unstable resulting in close to 0.0 mAP scores. Experimental results using *sim2real* adaptation settings are shown in Table 4.5. We show that by increasing the temperature T along fixed α parameter the effect of soft-labels becomes less dominant during in the training and the accuracy converges to the hard-label $\alpha = 1$ setting.

4.4.7 Importance of multi-modal image translation

image-to-image model	*sim2real*	*adverse weather*	*cross-camera*
CycleGAN	46.82	**44.71**	34.67
MUNIT	**47.43**	44.33	**40.23**

Table 4.6: Qualitative comparison of target accuracy achieved by different image translation models.

In Table 4.6, we provide experimental results on the importance of covering the multi-modality present in the target dataset while performing image to image translation. Cy-

teacher threshold	number of pseudo-labels	student source mAP	student target mAP
0.0	296k	50.10	34.27
0.1	219k	50.20	36.81
0.2	160k	49.92	36.8
0.3	132k	50.28	37.64
0.4	115k	50.40	**38.01**
0.5	103k	**50.43**	37.98
0.6	93k	49.06	37.14
0.7	85k	49.62	36.34
0.8	77k	**50.43**	36.93
0.9	68k	49.58	35.18

Table 4.7: Qualitative comparison of different cut-off parameters for selecting pseudo-labels.

cleGAN [251] allows only a deterministic translation, therefore it provides a *one-to-one* mapping from source to target images. On the other hand, MUNIT [103] allows us to sample different style vectors from a normal distribution to control the appearance of the result of the image translation, providing a *one-to-many* mapping between source and target images. We observe that in 2 out of 3 domain adaptation scenarios using multi-modal image translation improves the target accuracy. In the *adverse weather* scenario is an example of diminishing returns, since converting *Cityscpes* [44] to the synthetic *Foggy Cityscapes* [183] target domain is less challenging than the other two settings (see Table 4.4), therefore CycleGAN [251] performance is on par with MUNIT [103].

4.4.8 Pseudo-Label performance sensitivity to threshold hyper-parameter

Applying the teacher network to target images yields a set of predictions which we reuse as pseudo-labels for training the student network. However, it is not trivial whether we should treat every prediction as a correct label or not. In our experiments we used a predetermined threshold applied on the per-instance *objectness-scores* (see Faster-RCNN [179] for details) to select which predictions to keep. In Table 4.7, we show experimental results on the sensitivity of the pseudo-labeling step with regards to the target accuracy measured on the *adverse weather* adaptation scenario.

4.4.9 Class specific comparison for the Adverse Weather experiment

Since the target dataset, Foggy Cityscapes [183] is artificially generated from the source dataset, Cityscapes [44] the categories are retained. The class specific comparison can be

found in Table 4.8.

Method	$person_{50}$	$rider_{50}$	car_{50}	$truck_{50}$	bus_{50}	$train_{50}$	$mcycle_{50}$	$bicycle_{50}$	mAP_{50}	
Faster-RCNN [179]	36.9	44.0	47.9	18.3	34.5	21.2	25.3	39.8	33.48	-
DA-Faster-RCNN [39]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.60	+8.8
Pseudo-Labeling [105]	31.9	39.9	48.0	25.1	39.9	27.2	25.0	34.1	33.90	+2.0
MTOR [19]	30.6	41.4	44.0	21.9	38.6	40.6	28.3	35.6	35.10	+8.2
PDA [98]	36.0	45.5	54.4	24.3	44.1	25.8	29.1	35.9	36.90	+10.0
SCDA [252]	33.5	38.0	48.5	26.5	39.0	23.3	28.0	33.6	33.80	+7.6
SWDA [182]	36.2	35.3	43.5	30.0	29.9	42.3	32.6	24.5	34.3	+14.0
DM [116]	30.8	40.5	44.3	27.2	38.4	34.5	28.4	32.2	34.60	+16.7
Noisy Labeling [115]	35.1	42.2	49.2	30.1	45.3	27.0	26.9	36.0	36.45	+4.5
MAF [91]	28.2	39.5	43.9	23.8	39.9	33.3	29.2	33.9	34.00	+15.2
CFFA [249]	34.0	40.9	52.1	30.8	43.2	29.9	34.7	37.4	38.6	**+17.8**
ours	**43.5**	**52.0**	**63.2**	**34.7**	**52.7**	**45.8**	**37.1**	**49.4**	**47.31**	+13.8
oracle	44.4	51.5	66.0	38.0	55.0	37.8	38.6	49.2	47.56	+14.1

Table 4.8: Cityscapes → Foggy Cityscapes. The unsupervised domain adaptation performance is on par with the network trained on the target (covers 98% of the performance gap). As highlighted, CFFA [249] achieves the highest respective improvement, however their baseline provides larger room for improvement (see Figure 4.3).

4.4.10 Fair Comparison

Detector backbone and optimization. Prior works implement the object detector, Faster-RCNN [179] using different backbones. VGG16 is used in [39, 252, 91, 98, 116], Inception V2 in [115], and [19] uses 50/152-layer ResNet [90] for different settings.

Evaluation. The choice of evaluation tools is missing from existing works. This is problematic because the mean Average Precision (mAP) metric is sensitive to the underlying implementation. Using the pre-trained networks from Detectron2 [224] for VOC 2007 and the default VOC evaluation tool the reported mAP and mAP50 performance is 51.9 and 80.3, while evaluating the same network on the same dataset with the COCO evaluation tool the corresponding scores are 46.9 and 76.2, resulting in a difference of 5.0% and 4.1% between the two measurements. This "relative improvement" from choosing an appropriate evaluation tool would be on-par with the improvement of [105, 115, 252] (+1.98%, +4.53%, +5.1%).

4.5 Conclusion

In this work we have addressed the problem of cross-domain object detection. After conducting extensive studies on previous approaches, we categorized different components into 3 classes: image, feature and output level domain adaptation. Our proposed method

is the first to successfully employ all levels of domain adaptation. Our method consists of 3 stages: 1) translate labeled source images to make them appear similar to target images 2) train a teacher network on translated images while aligning its features to the target domain 3) use the teacher model to generate pseudo-labels, then train a student model on the translated source images and the pseudo-labeled target images. Compared to recent state-of-the art works, our method uses simple adaptation techniques at each stage and still outperforms complex algorithms on every commonly used benchmark.

4.6 Acknowledgements

4.7 Conclusion

We conduct a comprehensive analysis of various UDA techniques, including pixel-level adaptation, feature alignment, and teacher-student approaches. Moreover, we propose a novel joint training method that combines these paradigms, resulting in an object detection model with comparable complexity to its supervised counterpart while significantly outperforming other UDA methods. To further our understanding of utilizing unsupervised data that is sampled from a different distribution than the supervised data, we explore the intersection between UDA and Online Continual Learning. In the next chapter we introduce a new experimental setup where both the supervised and the unsupervised data distribution is continuously evolving over time.

Label Delay

in Online Continual Learning

Chapter Teaser

How do we model a continuously changing data distribution? How can we extend this framework to model the delay caused by the annotation process? How does the delay impact the performance of the continually updated model? To what extent can we improve the performance in different label delay scenarios by simply using more compute? What are the techniques we can use to utilize the data before their corresponding label becomes available? How do these technique scale with more compute in comparison with the naive model?

5.1 Abstract

A critical yet often overlooked aspect in online continual learning is the label delay, where new data may not be labeled due to slow and costly annotation processes. We introduce a new continual learning framework with explicit modeling of the label delay between data and label streams over time steps. In each step, the framework reveals both unlabeled data from the current time step t and labels delayed with d steps, from the time step

$t-d$. In our extensive experiments amounting to 25000 GPU hours, we show that merely increasing the computational resources is insufficient to tackle this challenge. Our findings highlight significant performance declines when solely relying on labeled data when the label delay becomes significant. More surprisingly, state-of-the-art Self-Supervised Learning and Test-Time Adaptation techniques that utilize the newer, unlabeled data, fail to surpass the performance of a naïve method that simply trains on the delayed supervised stream. To this end, we propose a simple, robust method, called Importance Weighted Memory Sampling that can effectively bridge the accuracy gap caused by label delay by prioritising memory samples that resemble the most to the newest unlabeled samples. We show experimentally that our method is the least affected by the label delay factor, and successfully recovers the accuracy of the non-delayed counterpart.

5.2 Introduction

Machine learning models have become the de facto standard for a wide range of applications, including social media [161], finance [33], and healthcare [75]. However, these models usually struggle when the distribution from which the data is sampled is constantly changing over time, which is common in real-world scenarios. This challenge continues to be an active area of research known as Continual Learning (CL). However, most prior art in CL examines this problem with a presumption of the immediate availability of labels once the data is collected. This assumption rarely holds in real-world scenarios.

Consider the task of monitoring recovery trends in patients after surgeries. Doctors gather health data from numerous post-operative patients regularly. However, this data does not immediately indicate broader recovery trends or potential common complications. To make informed determinations, several weeks of extensive checks and tests across multiple patients are needed. Only after these checks are completed, the gathered data can be labeled as indicating broader “recovery” or “complication” trends. However, by the time

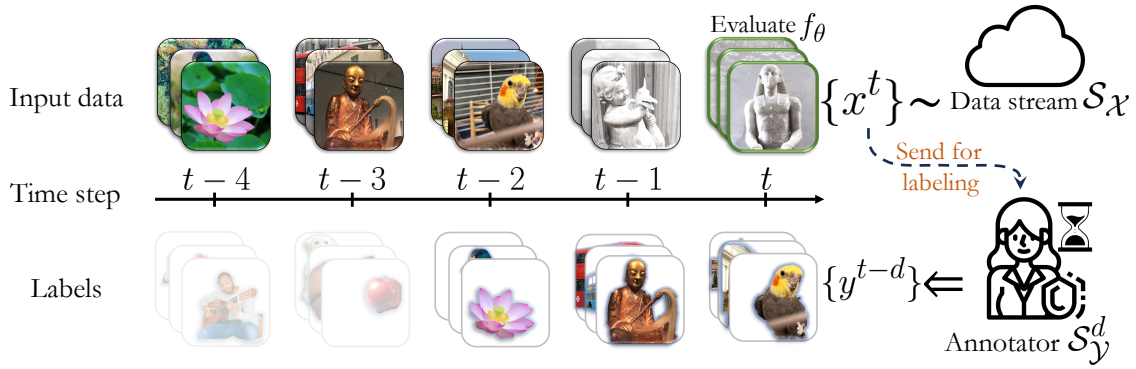


Figure 5.1: **Illustration of label delay.** This figure shows a typical Continual Learning (CL) setup with label delay due to annotation. At every time step t, the data stream $\mathcal{S}_\mathcal{X}$ reveals a batch of unlabeled data $\{x^t\}$, on which the model f_θ is evaluated (highlighted with green borders). The data is then sent to the annotator $\mathcal{S}_\mathcal{Y}$ who takes d time steps to provide the corresponding labels. Consequently, at time step t the batch of labels $\{y^{t-d}\}$ corresponding to the input data from d time steps before becomes available. The CL model can be trained using the **delayed labeled data** (shown in color) and the **newest unlabeled data** (shown in grayscale). In this example, the stream reveals three samples at each time step and the annotation delay is $d = 2$.

the data is gathered, assessed, labeled, and a model is trained, new patient data might follow trends that do not exist in the training data yet. This leads to a repeating cycle: collecting data from various patients, assessing the trends, labeling the data, training the model, and then deploying it on new patients. The longer this cycle takes, the more likely it is going to affect the model's reliability, a challenge we refer to as **label delay**.

In this paper, we propose a CL setting that explicitly accounts for the delay between the arrival of new data and the corresponding labels, illustrated by Figure 5.1. In our proposed setting, the model is trained continually over discrete time steps with a label delay of d steps. At each step, two batches of data are revealed to the model: unlabeled new samples from the current time step t, and the labels of the samples revealed at the step $t-d$. First, we show the naïve approach where the model is only trained with the labeled data while ignoring all unlabeled data. While this forms a strong baseline, its performance suffers significantly from increasing the delay d. We find that simply increasing the number of parameter updates per time step does not resolve the problem. Hence, we examine a number of popular approaches which incorporate the unlabeled data to improve this

naïve baseline. We investigate semi-supervised learning, self-supervised learning and test-time adaptation approaches which are motivated for slightly different but largely similar settings. Surprisingly, out of 12 different methods considered, none could outperform the naïve baseline given the same computational budget. Motivated by our extensive empirical analysis of prior art in this new setting, we propose a simple and efficient method that outperforms every other approach across large-scale datasets; in some scenarios it even closes the accuracy gap caused by the label delay. Our contributions are threefold:

- We propose a new formal Continual Learning setting that factors label delay between the arrival of new data and the corresponding labels due to the latency of the annotation process.

- We conduct extensive experiments ($\sim 25,000$ GPU hours) on various Online Continual Learning datasets, such as CLOC [20], CGLM [170], FMoW [43] and Yearbook [70]. Following recent prior art on Budgeted Continual Learning [169, 69], we compare the best performing Self-Supervised Learning [13], Semi-Supervised Learning [76] and Test Time Adaptation [131] methods and find that none of them outperforms the naïve baseline that simply ignores the label delay and trains a model on the delayed labeled stream.

- We propose **I**mportance **W**eighted **M**emory **S**ampling to rehearse past labeled data most similar to the most recent unlabeled data, bridging the gap in performance. IWMS outperforms the naïve method significantly and improves over Semi-Supervised, Self-Supervised Learning and Test-Time Adaptation methods across diverse delay and computational budget scenarios with a negligible increase in computational complexity. We further present an in-depth analysis of the proposed method.

5.3 Related Work

Label Delay in Online Learning. While the problem of delayed feedback has been studied in the online learning literature [220, 155], the scope is limited to problems of spam detection and other synthetically generated, low-complexity data [99, 76] and often view input images as "side info" [112]. Additionally, methods and error bounds proposed in [120, 172, 168, 64] are more focused on expert selection rather than representation learning, most of which cannot generalize to unstructured, large-scale image classification datasets. Furthermore, most of the prior art does not differentiate between the past and future unlabeled data. In our proposal, all unlabeled data is newer than the last labeled data due to delayed annotation, as illustrated in Figure 5.17. For a more in-depth analysis of prior art on online learning, please see our expanded literature review in the Supplementary Material 5.11.12.

Continual Learning. Early work on continual learning primarily revolved around task-based continual learning [18, 3], while recent work focuses on the task-free continual learning setting [5, 6, 20]. This scenario poses a challenge for models to adapt as explicit task boundaries are absent, and data distributions evolve over time. GDumb[171] and BudgetCL[169] demonstrate that minimalistic methods can outperform most offline and online continual learning approaches. RealtimeOCL [69] shows that Experience Replay [27] is the most effective method, outperforming more popular continual learning methods, such as ACE [18], LwF [130], RWalk [26], PoLRS [20], MIR [4] and GSS [7], when methods are normalized by their computational complexities. RealtimeOCL also considers *delay*, however, their delay arises from model complexity; in their *fast-stream* scenario, the stream releases input-label pairs quicker than models can update, causing models to be trained on an older batch of samples. In essence, labels are still instantly available in RealtimeOCL, while our work examines delay attributed to the non-instantaneous arrival of labels. RapidOCL [87] highlighted the exploitation of label-correlation in online

continual learning, with a focus on measuring online accuracy through future samples. In contrast, our framework allows the models to leverage the more recent, unlabeled data for adaptation.

Semi-Supervised Learning. While the labels arrive delayed, our setting allows the models to use new unlabeled data immediately. Possible directions to leverage the most recent unlabeled data entails Pseudo-Labeling (or often referred to as their broader category, Semi-Supervised Learning) methods [76] and Self-Supervised Semi-Supervised Learning (S4L) methods [242]. Pseudo-labeling techniques predict the labels of the samples before their true label becomes available to estimate the current state of the joint distribution of input and output pairs. This in turn allows the model to fit its parameters on the estimated data distribution. On the other hand, S4L integrates self-supervised learning, such as predicting the rotation of an image or the relative location of image patches, with the semi-supervised learning framework. We replace the early self-supervised tasks of S4L [242] with more recent objectives from Balestriero [13]. While a growing line of work adapts S4L to continual learning to make use of unlabeled data in continual learning settings, such as CaSSLe [59] in task-agnostic settings and SCALE [239] in task-free settings, most previous work did not perform a comprehensive examination of PL and S4L under a strict computational budget.

Test-Time Adaptation. Besides semi-supervised learning, TTA methods are also designed to adapt models with unlabeled data, sampled from the similar distribution as the evaluation samples. Entropy regularization methods like SHOT [132] and TENT [215] update the feature extractor or learnable parameters of the batch-normalisation layers [106] to minimize the entropy of the predictions. SAR [164] incorporates an active sampling scheme to filter samples with noisy gradients. More recent works consider Test Time Adaptation in online setting [2] or Continual Learning setting [217]. In our experiments, we fine-tuning the model with ER [27] across time steps and adapting a copy of the model

Algorithm 2 Single OCL time step with Label Delay

1: The Stream $\mathcal{S}_{\mathcal{X}}$ reveals a batch of images $\{x_i^t\}_{i=1}^n \sim \mathcal{D}_t$;
2: The model f_{θ_t} makes predictions $\{\hat{y}_i^t\}_{i=1}^n$ for the new revealed batch $\{x_i^t\}_{i=1}^n$;
3: The Annotator $\mathcal{S}_{\mathcal{Y}}^d$ reveals labels $\{y_i^{t-d}\}_{i=1}^n$;
4: The model f_{θ_t} is evaluated by comparing the predictions $\{\hat{y}_i^t\}_{i=1}^n$ and true labels $\{y_i^t\}_{i=1}^n$, where the true labels are only for testing;
5: The model f_{θ_t} is updated to $f_{\theta_{t+1}}$ using labeled data $\cup_{\tau=1}^{t-d}\{(x_i^\tau, y_i^\tau)\}_{i=1}^n$ and unlabeled data $\cup_{\tau=t-d}^{t}\{x_i^\tau\}_{i=1}^n$ under a computational budget $\mathcal{C}$.

with TTA to the most recent input samples at each time step.

5.4 Problem Formulation

We follow the conventional online continual learning problem definition proposed by Cai [20]. In such a setting, we seek to learn a model $f_\theta : \mathcal{X} \to \mathcal{Y}$ on a stream $\mathcal{S}$ where for each time step $t \in \{1, 2, \dots\}$ the stream $\mathcal{S}$ reveals data from a time-varying distribution $\mathcal{D}_t$ sequentially in batches of size n. At every time step, f_θ is required to predict the labels of the coming batch $\{x_i^t\}_{i=1}^n$ first. Followed by this, the corresponding labels $\{y_i^t\}_{i=1}^n$ are immediately revealed by the stream. Finally, the model is updated using the most recent training data $\{(x_i^t, y_i^t)\}_{i=1}^n$.

This setting, however, assumes that the annotation process is instantaneous, i.e.,the time it takes to provide the ground truth for the input samples is negligible. In practice, this assumption rarely holds. It is often the case that the rate at which data is revealed from the stream $\mathcal{S}$ is faster than the rate at which labels for the unlabeled data can be collected, as opposed to it being instantaneously revealed. To account for this delay in accumulating the labels, we propose a setting that accommodates this lag in label availability while still allowing for the model to be updated with the most recent unlabeled data. We modify the previous setting in which labels of the data revealed at time step t will only be revealed after d time steps in the future.

At every time step t, the Annotator $\mathcal{S}_{\mathcal{Y}}^d$ reveals the labels for the samples from d time steps before, i.e.,, $\{(x_i^{t-d}, y_i^{t-d})\}_{i=1}^n$, while the data stream $\mathcal{S}_{\mathcal{X}}$ reveals data from the the current

time step, i.e.,, $\{x_i^t\}_{i=1}^n$. Recent prior art [171, 169, 69] introduces more reasonable and realistic comparisons between continual learning methods by imposing a computational complexity constraint on the methods. Similarly to [171, 169, 69], in our experiments the models are given a fixed computational budget $\mathcal{C}$ to update the model parameters from θ_t to θ_{t+1} for every time step t. To that end, our new proposed setting can be formalized per time step t, alternatively to the classical OCL setting, as described in Algorithm 2.

Note that this means at each time step t, the stream reveals a batch of *non-corresponding* images $\{x_i^t\}_{i=1}^n$ and labels $\{y_i^{t-d}\}_{i=1}^n$, as illustrated in Figure 5.1. With the label delay of d time steps, the images themselves revealed from time step $t-d$ to time step t can be used for training, despite that labels are not available.

A *naïve* way to solve this problem is to discard the unlabeled images and only train on labeled data $\cup_{\tau=1}^{t-d}\{(x_i^\tau, y_i^\tau)\}_{i=1}^n$, However, it worth noting that the model is still evaluated on the most recent samples from $\mathcal{S}_\mathcal{X}$. Thus, training on the labeled training data leads to the model at least being d steps delayed. Since in our setting the distribution from which the training and evaluation samples are drawn from is not stationary, this discrepancy severely hinders the performance, as discussed in detail in Section 5.6.

Furthermore, we shall show in Section 5.7 that the existing paradigms, such as Test-Time Adaptation and Semi-Supervised Learning, struggle to effectively utilise newer, unlabeled data to bridge the aforementioned discrepancy. Our observations indicate that the primary failure is from the excessive computational demands of processing unlabeled data. To that end, we propose Importance Weighted Memory Sampling that prioritises performing gradient steps on labeled samples that resemble the most recent unlabeled samples.

5.5 IWMS: Importance Weighted Memory Sampling

To mitigate the challenges posed by label delay in online continual learning, we introduce a novel method named **Importance Weighted Memory Sampling (IWMS)**. Recog-

Algorithm 3 Importance Weighted Memory Sampling

1: At time step t, for each unsupervised batch of size n, $\{x_i^t\}_{i=1}^n$, the model f_θ computes predictions $\{\tilde{y}_i^t\}_{i=1}^n$;
2: For every predicted label $\tilde{y}_i^t$, select labeled samples from the memory buffer $\{(x_j^M, y_j^M)\}$ where $y_j^M = \tilde{y}_i^t$;
3: Compute pairwise cosine feature similarities $K_{i,j} = \cos\left(h(x_i^t), h(x_i^M)\right)$ between each unlabeled sample x_i^t and selected memory samples x_j^M;
4: Select the most relevant supervised samples (x_k^M, y_k^M) by sampling $k \in \{1 \dots |M|\}$ from a multinomial distribution with parameters $K_{i,:}$;
5: Update the model f_θ using the selected supervised samples, aiming to match the distribution of the unlabeled data.

nizing the limitation of traditional approaches that either discard unlabeled data or utilize it in computationally expensive ways, IWMS aims to bridge the gap between the current distribution of unlabeled data and the historical distribution of labeled data. Instead of directly adapting the model to fit the newest distribution with unlabeled data, which is inefficient due to the lack of corresponding labels, IWMS cleverly adjusts the sampling process from a memory buffer. This method ensures that the distribution of selected samples closely matches the distribution of the most recent unlabeled batch. This nuanced selection strategy allows the continual learning model to effectively adapt to the most recent data trends, despite the delay in label availability, by leveraging the rich information embedded in the memory buffer.

As discussed in Section 5.6, using the most recent labeled samples for training leads to over-fitting the model to an outdated distribution. Thus, we replace the newest supervised data by a batch which we sample from the memory buffer, such that the distribution of the selected samples matches the newest unlabeled data distribution. The sampling process is detailed in Algorithm 3. It consists of two stages: first, at each time step t, for every unsupervised sample x_i^t in the batch of size n, we compute the prediction $\tilde{y}_i^t$, and select every labeled sample from the memory buffer (x_j^M, y_j^M) such that the true label of the selected samples matches the predicted label $y_j^M = \tilde{y}_i^t$. In the second stage, we compute the pairwise cosine feature similarities $\mathbf{K}_{i,j}$ between the unlabeled sample x_i^t and the selected memory samples x_j^M by $\mathbf{K}_{i,j} = \cos\left(h(x_i^t), h(x_j^M)\right)$, where h represents the learned feature extractor part of f_θ, directly before the final classification layer. Finally, we select the most relevant supervised samples (x_k^M, y_k^M) by sampling $k \in \{1 \dots |M|\}$

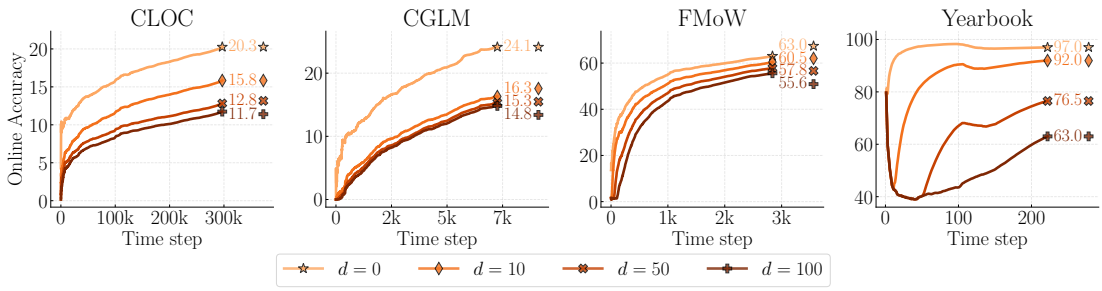


Figure 5.2: **Effects of Varying Label Delay.** The performance of a *Naïve* Online Continual Learner model gradually degrades with increasing values of delay d.

from a multinomial distribution with parameters $\mathbf{K}_{:,j}$. Thus, we rehearse samples from the memory which (1) share the same true labels as the predicted labels of the unlabeled samples, (2) have high feature similarity with the unlabeled samples.

To avoid re-computing the feature representation $h(x^M)$ for each sample in the memory buffer after every parameter update, we store the corresponding features of the input data computed for the predictions during the evaluation (Step 4 in Algorithm 2). This technique greatly reduces the computational cost of our method, but comes at the price of using outdated features. Such trade-off is studied in detail by contemporary Self-Supervised Literature [88, 35, 25] observing no significant impact on performance. We ablate the alternative option of selecting samples based only on their similarity in the Supplementary Material 5.11.11.

5.6 The Cost of Ignoring Label Delay

To better understand how label delay influences the performance of a model, we begin with the **Naïve** approach, i.e.,ignoring the most recent data points until their label becomes available and exclusively training on outdated labeled samples. More specifically, we are interested in measuring the performance degradation under various label delay d and computational budget $\mathcal{C}$ scenarios. To this end, we conduct experiments over 4 datasets, in 4 computational budget and 3 label delay settings. We analyse the results under normalised computational budget (Section 5.6.2) and demonstrate that the accuracy drop

can be only partially recovered by increasing the computational budget (Section 5.11.5).

5.6.1 Experimental Setup

Datasets. We conduct our experiments on four large-scale online continual learning datasets, Continual Localization (CLOC) [20], Continual Google Landmarks (CGLM) [170], Functional Map of the World (FMoW) [43], and Yearbook [70]. The last two are adapted from the Wild-Time challenge [234]. More statistics of the benchmarks are in Supplementary. We follow the same *training* and *validation* set split of CLOC as in [20] and o CGLM as in [170] and the official released splits for FMoW [43] and Yearbook [70].

Architecture and Optimization. Similarly to prior work [69, 169], we use ResNet18 [90] for backbone architecture. Furthermore, in our experiments, the stream reveals a mini-batch, with the size of $n = 128$ for CLOC, FMoW, Yearbook and $n = 64$ for CGLM. We use SGD with the learning rate of 0.005, momentum of 0.9, and weight decay of 10^{-5}. We apply random cropping and resizing to the images, such that the resulting input has a resolution of 224×224.

Baseline Method In our experiments, we refer to the *Naïve* method as the one naively training one labeled data soly. We apply the best continual learning mechanism as mentioned by [69] , Experience Replay (ER) [27], to eliminate the need to compare with other continual learning methods . The memory buffer size is consistently 2^{19} samples throughout our experiments unless stated otherwise. The First-In-First-Out mechanism [27, 20] to update the buffer. We report the Online Accuracy [20] at each time step in Step 4 of Algorithm 2 under label delay d. In our quantitative comparative analysis, for simplicity, we use the final Online Accuracy scores, denoted by Acc_d.

Computational Budget and Label Delay. Normalising the computational budget is necessary for fair comparison across CL methods, thus, we define $\mathcal{C} = 1$ as the number of FLOPs required to make one backward pass with a ResNet18 [90], similarly to

BudgetCL[169] and RealtimeOCL[69]. When performing experiments with a larger computational budget, we take integer multiplies of $\mathcal{C}$ to apply $\mathcal{C}$ parameter update steps per stream time steps. The proposed label delay factor d represents the amount of time steps the labels are delayed with. Note that, for $\mathcal{C} = 1, d = 0$, our experimental setting is identical to prior art[20, 69].

5.6.2 Observations

In Figure 5.2, we analyze how varying the label delay $d \in \{0, 10, 50, 100\}$ impacts the performance of Naïve on four different datasets, CLOC [20], CGLM [169], FMoW [43] and Yearbook [70]. The label delay impacts the online accuracy differently across all scenarios, thus, below we provide our observations case-by-case.

On **CLOC**, the non-delayed ($d = 0$) Naïve achieves $\text{Acc}_0 = 20.2\%$, whereas the heavily delayed counterpart ($d = 100$) suffers significantly from the label delay, achieving only $\text{Acc}_{100} = 11.7\%$. Interestingly, label delay influences the accuracy in a monotonous, but non-linear fashion, as half of the accuracy drop is caused by a very small amount of delay: $\text{Acc}_{10} - \text{Acc}_0 = -4.4\%$. In contrast, the accuracy degradation slows down for larger delays,i.e.,the accuracy gap between two larger delay scenarios ($d = 50 \rightarrow 100$) is rather marginal $\text{Acc}_{100} - \text{Acc}_{50} = -1.1\%$. We provide further evidence on the monotonous and smooth properties of the impact of label delay with smaller increments of d in the Supplementary Material 5.11.3.

For **CGLM** the accuracy gap landscape looks different: the majority of the accuracy decrease occurs by the smallest delay $d = 0 \rightarrow 10$, resulting in a $\text{Acc}_{10} - \text{Acc}_0 = -7.9\%$ drop. Subsequent increases ($d = 10 \rightarrow 50$ and $d = 50 \rightarrow 100$) impact the performance to a significantly smaller extent: $\text{Acc}_{50} - \text{Acc}_{10} = -1\%$ and $\text{Acc}_{100} - \text{Acc}_{50} = -0.5\%$.

In the case of **FMoW**, where the distribution shift is less imminent (i.e.,the data distribution varies less over time), the difference between the delayed and the non-delayed

counterparts should be small. This is the case for the satellite image data in the FMoW dataset, where the accuracy drops are $-2.8\%, -2\%, -1.9\%$ for $d = 0 \rightarrow 10 \rightarrow 50 \rightarrow 100$, respectively.

The **Yearbook**'s binary classification experiments highlight an important characteristic: if there is a significant event that massively changes the data distribution, such as the change of men's appearance in the 70's [70] the non-delayed Naïve ($d = 0$) suffers a small drop in Online Accuracy (at the middle of the time horizon $t = 130$), but quickly recovers as more data starts to appear. In contrast, under small and moderate delay ($d = 10, 50$), the decline is more emphasised and the recovery is delayed (at $t = 120, 180$, respectively). Interestingly, under large delay ($d = 100$), the decline is not apparent anymore, however the Acc_{100} stagnates until the relevant data arrives ($t = 200$). Notice, that such temporal shift is only visible across the profile of the curves, because the amount of delay is comparable in size to the time horizon, i.e., $\frac{d_{\max}}{t_{\max}} = \frac{1}{3}$, whereas this fraction is close to 0 in our other experiments. Alongside with more detailed investigation, we provide visual examples of the dataset to support our claims in the Supplementary Material 5.11.4.

5.6.3 Section Conclusion

Over- or Under-fitting. While in the experiments we report results under a single computational budget $\mathcal{C}$ per dataset, it is reasonable to suspect that the results might look different under smaller or larger budget. To this end, we ablate the effect of $\mathcal{C}$ over various delay scenarios, on multiple datasets in the Supplementary Material 5.11.5.

Common patterns. We argue that the consistent, monotonic accuracy degradation, present in all of our experiments, is due to the non-stationary property of the data distribution that creates a distribution shift. Our hypothesis is supported by the findings of Yao [234]. A complementary argument is presented by Hammoud [87], stating that the underlying datasets have high temporal correlations across the labels, i.e.,images of the

same categories arrive in bursts, allowing an online learning model to easily over-fit the label distribution even without using the input images.

Motivation for Delay Specific Solutions. As our experiments suggest so far, label delay is indeed an extremely elusive problem, not only because it inevitably results in an accuracy drop, but because the severity of the drop itself is hard to estimate a-priori. We showed that the accuracy gap always increases monotonically with increasing delay, nevertheless the increase of the gap can be gradual or sudden depending on the dataset and the computation budget. This motivates our efforts of designing special techniques to address the challenges of label delay. In the next set of experiments, we augment the Naïve training by utilizing the input images *before* their corresponding labels become available.

5.7 Utilising Data Prior to Label Arrival

In our proposed label delay experimental setting, we showed the larger the delay the more challenging it is for Naïve, a method that relies only on older labeled data, to effectively classify new samples. This is due to a larger gap in distribution between the samples used for training and for evaluation. This begs the question of whether the new unlabeled data can be used for training to improve over Naïve, as it is much more similar to the data that the model is evaluated on.

We propose four different paradigms for utilizing the unlabeled data, namely, Importance Weighted Memory Sampling (**IWMS**), Semi-Supervised Learning via Pseudo-Labeling (**PL**), Self-Supervised Semi-Supervised Learning (**S4L**) and Test-Time Adaptation (**TTA**). We integrate several methods of each family into our setting and evaluate them under various delays and computational budgets. In particular, we adapt each paradigm individually by augmenting the parameter update (Step 5 of Algorithm 2) of Naïve, described in detail in the following subsections. Furthermore, to quantify the how much of the accuracy gap ($G_d = \text{Acc}_d^{\text{Naïve}} - \text{Acc}_0^{\text{Naïve}}$) is recovered, we use the formula $R_d^* = \frac{\text{Acc}_d^* - \text{Acc}_d^{\text{Naïve}}}{|G_d|}$, namely the

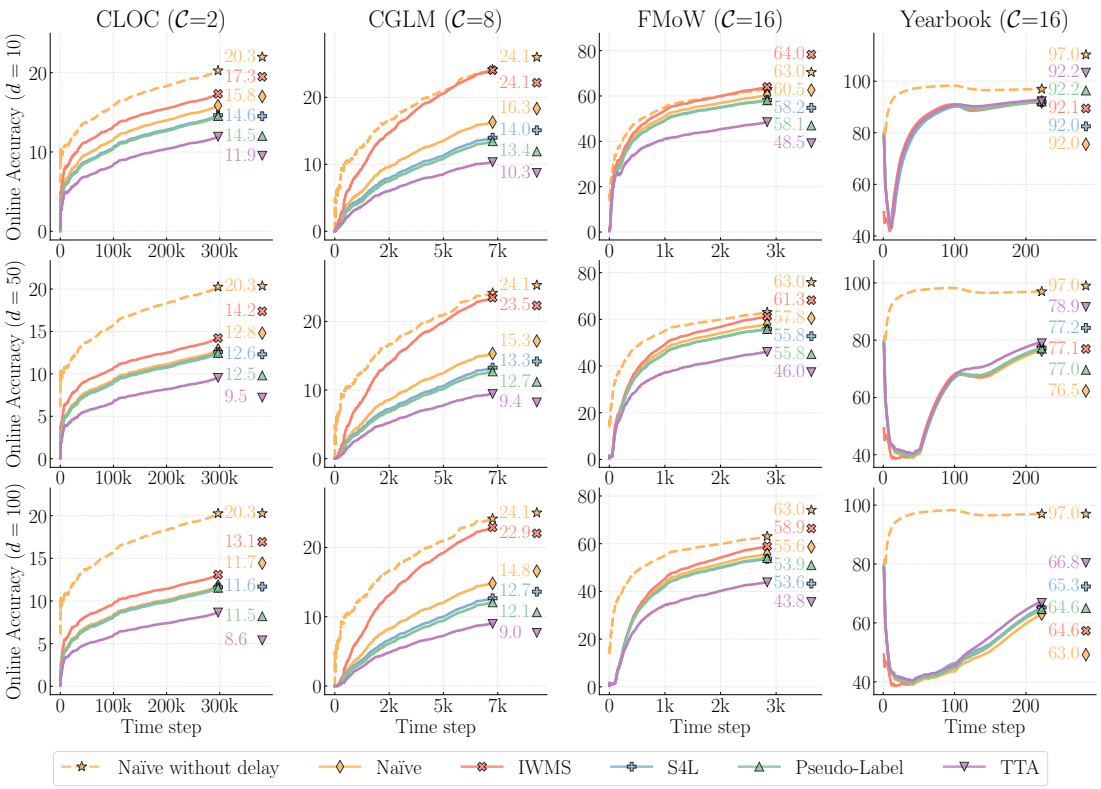


Figure 5.3: **Comparison of various unsupervised methods.** The accuracy gap caused by the label delay between the *Naïve without delay* and its delayed counterpart *Naïve*. Our proposed method, *IWMS*, consistently outperforms all categories under all delay settings on three out of four datasets.

improvement of the method divided by the extent of the accuracy gap for a given delay factor d.

5.7.1 Experiment Setup

Importance Weighted Memory Sampling (IWMS). The only additional cost of IWMS compared to Naïve is the cost of evaluating the similarity scores, which is still less than 1% of the inference cost for 100K samples, and can be evaluated in parallel, therefore we consider it negligible. Since our method simply replaces the newest supervised samples with the most similar samples from the replay buffer, we do not require any additional backward passes to compute the auxiliary objective. Therefore, the computational budget of our method is identical to the Naïve baseline, i.e.,, $\mathcal{C}_{\text{IMWS}} = 1$.

Self-Supervised Semi-Supervised Learning. For integrating S4L methods, we adopt the most effective approach through iterative optimization of both supervised and unsupervised losses. We report the best results across the three main families of contrastive losses, i.e.,, Deep Metric Learning Family (MoCo [88], SimCLR [34],and NNCLR [55]), Self-Distillation (BYOL [84] and SimSIAM [37], and DINO [25]), and Canonical Correlation Analysis (VICReg [15], BarlowTwins [241], SWAV [24], and W-MSE [56]).

For fair comparison, we normalise the computational complexity of the compared methods. According to [169, 69], Naïve augmented with Self-Supervised Learning at each time step takes two backward passes, since they augment each input images to two views, thus $\mathcal{C}_{\text{S4L}} = 2$. We provide further explanation of our S4L adaptation in the Supplementary Material 5.11.2.

Pseudo-Labeling. To make use of the newer unlabeled samples, we adopt the most common Semi-Supervised Learning technique [76]: Pseudo-Labeling (PL). To predict the labels of the samples before their true label becomes available we use a surrogate model g_ϕ. After assigning the predicted labels $\{\tilde{y}_i^t\}$ to each input data $\{x_i^t\}$ at time step t for $i = 1..n$, the main model f_θ is updated over the union of old, labeled memory samples and new *pseudo-labeled* samples $\{(x_i^\tau, y_i^\tau)\}_{\tau=1}^{t-d} \cup \{(x_i^t, \tilde{y}_i^t)\}$ using standard Cross Entropy loss. Once f_θ is updated, we update the parameters of the surrogate model g_ϕ following the momentum update policy [76] with hyper-parameter λ, such that $\phi_{\text{new}} = \lambda\phi_{\text{old}} + (1-\lambda)\theta_{\text{old}}$.

For simplicity, we ignore the computational cost of the surrogate model g_ϕ inferring the pseudo-labels $\tilde{y}$. Nevertheless, the main model f_θ is trained on double the amount of samples as Naïve, n labeled and n pseudo-labeled, therefore we define $\mathcal{C}_{\text{PL}} = 2$.

Test-Time Adaptation As done for other paradigms, we have extensively evaluated all reasonable candidates to adapt traditional TTA methods to our setting. We find performing the unsupervised TTA step the most effective when only a single update is taken (in Step 5 of Algorithm 2), exactly before the evaluation step (Step 2 of Algorithm 2)

of the next step. Therefore, for all the parameter updates apart from the last one we perform identical steps to Naïve Furthermore, we found TTA updates severly impact the continual learning process of the Naïve when the parameters are iteratively optimised across the two objectives. Thus, before each TTA step, we clone the model parameters θ to a surrogate model g_ϕ, by performing the TTA step (with ϵ hyper-parameter) using the newest batch of unlabeled data $\phi = \theta - \epsilon \nabla_\theta \mathcal{L}_{\text{TTA}}\{x_i^t\}$ and perform the evaluation (Step 2 of Algorithm 2) of the next time step.

To represent the state of the art in TTA, we adapt and compare the following methods: TENT [215], EATA [163], SAR [164], and CoTTA [217], in Figure 5.3. Furthermore, for the result of our hyper-parameter tuning is provided in the Supplementary Material 5.11.7.

For fair comparison, we train and evaluate all TTA methods under normalised computational budgets. However, several methods, such as CoTTA [217] and SAR [164] abuse the absence of formal computational constraints in traditional Test-Time Adaptation settings by computing the entropy of the predictions of the input data up to 32× different augmentations. Methods, such as EATA [163] further complicate the complexity normalisation problem by using multiple smaller-sized crops of the input image. To simplify our comparisons, we ignore the cost of model inference, thus $\mathcal{C}_{\text{TTA}} = 1$. More specifically, under a fixed computational budget $\mathcal{C}$, at every time step, we perform $\mathcal{C} - 1$ supervised steps on f_θ identically to Naïve followed by a single step of TTA.

5.7.2 Observations

Figure 5.3 illustrates our most important results of our work. It shows to what extent we can recover the accuracy gap caused by the label delay between the *Naïve without delay* and its delayed counterpart *Naïve*. We evaluate our proposed method, *IWMS*, and compare it against the three adopted paradigms, *S4L*, *PL* and *TTA*. We report the best performing method of each paradigm with hyper-parameters tuned on the first 10% of

each label delay scenario (further detailed in the Supplementary Material 5.11.6 and 5.13. To give the best representation of the landscape of how these techniques perform, we train and evaluate them over four datasets, three label delay settings ($d = 10, 50, 100$) and four computational budget constraints ($\mathcal{C} = 2, 8, 16, 16$).

IWMS. On the largest dataset, containing 39M samples, **CLOC** [20], the accuracy drop of Naïve is $G_d = -4.5\%, -7.5\%, -8.6\%$ for $d = 10, 50, 100$, respectively. Our proposed method, IWMS, achieves $\text{Acc}_d = 17.3\%, 14.2\%, 13.1\%$ final Online Accuracy, which translates to $R_d = 33\%, 19\%, 16\%$ recovery for $d = 10, 50, 100$, respectively. While there is a slow decline over increasing delays, the improvement over Naïve is consistent. On **CGLM** [169], the accuracy drop is $G_d = -7.8\%, -8.8\%, -9.3\%$ for the three increasing delays, respectively. IWMS exhibits outstanding results, $\text{Acc}_d = 24.1\%, 23.5\%, 22.9\%$ meaning that the accuracy gap *is fully recovered* by the method for $d = 10$. More specifically, the recovery is $R_d = 100\%, 93\%, 87\%$ for $d = 10, 50, 100$. The results on **FMoW** [43] are even more surprising, as IWMS not only recovers the accuracy gap but *outperforms* the non-delayed Naïve counterpart in the $d = 10$ scenario. More specifically, the accuracy drops for the increasing delays are $G_d = 2.5\%, 3.2\%4.4\%$ and $R_d = \underline{140\%}, 67\%, 45\%$. We hypothesise this is due to the fact that under a large $\mathcal{C}$, repeated parameter updates with sub-optimal sampling strategies lead to over-fitting to the outdated state of the data distribution, as explained in detail in Section 5.8. On **Yearbook** [70], IWMS performs on-par with Naïve in every scenario. The accuracy gaps are $G_d = -5\%, -20.5\%, -34\%$ whereas the recover scores are marginal: $R_d = 1\%, 0\%, 0\%$. We argue this is due to two factors: the brevity of the dataset in comparison to the other datasets and the difficulty of the task without prior knowledge on appearance and fashion trends.

Semi-Supervised Methods. S4L and PL performs very similarly to each other under all studied scenarios: the largest difference in their performance is 0.7% on Yearbook, under $d = 50$ label delay. Therefore, we report their performance together, picking the better

performing variant for numerical comparisons. Notice that in every scenario the delayed Naïve baseline performance is not be achieved, which is due to the computational budget constraint. More specifically, since $\mathcal{C}_{\text{SSL}} = 2 \times \mathcal{C}_{\text{Naïve}}$, optimising the standard classification objective over the older, supervised samples for twice the number of parameter updates is more beneficial across all scenarios than optimising the Pseudo-Labeling classification objective or the Contrastive loss over the newer unlabeled images. In the Supplementary Material 5.14, we provide further evidence and explanation of this claim. On **CLOC**, S4L slightly outperforms PL by +0.1% for all label scenarios, however $R_d = -27\%, -2\%, -7\%$ for $d = 10, 50, 100$, respectively. Similarly, on **CGLM**, S4L outperforms PL by +0.6%, for all label scenarios and achieves a negative recovery score $R_d = -29\%, -27\%, -23\%$. On **FMoW** and **Yearbook**, the differences between the accuracy of Naïve, S4L and PL are negligible as the largest improvement over Naïve is +2.3% on Yearbook under the large label delay scenario $d = 100$.

TTA. In Figure 5.3, we find that TTA consistently under-performs every method, *including* the delayed Naïve, under every delay scenario on the CLOC, CGLM and FMoW datasets Nevertheless, on Yearbook TTA successfully outperforms IWMS, S4L, PL and Naïve by up to +1.7% in the moderate label delay scenario $d = 50$. Over the four dataset, the exact extent of the recovery of the accuracy gap R_d for $d = 10, 50, 100$, respectively, is as follows: on **CLOC** $R_d = -87\%, -44\%, -36\%$, on **CGLM** $R_d = -77\%, -67\% - 62\%$, on **FMoW** $R_d = \underline{-480\%}, \underline{-227\%}, -159\%$ and on **Yearbook** $R_d = 4\%, 11\%, 11\%$. The disproportionately severe negative result on FMoW is due to the otherwise small accuracy gap $G_d = -2.5\%, -5.2\%, -7.4\%$. More importantly, we hypothesize that TTA fails to outperform Naïve because the common assumptions, upon which TTA methods were designed, are broken. Such assumptions of the Test-Time Adaptation settings are: (1) before the adaptation takes place, the model has already converged and achieved a good performance on the training data, (2) the test data distribution does not change over time and sufficient amount of unsupervised data is available for adaptation. In contrast, in

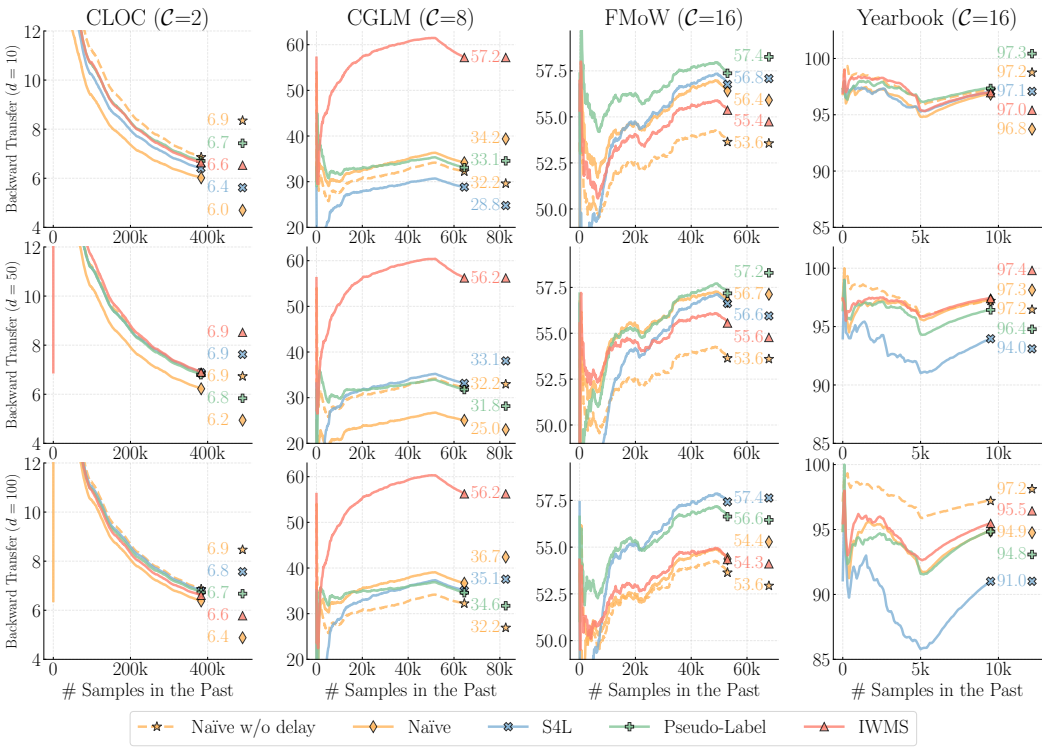


Figure 5.4: **Backward transfer.** Measuring forgetting on the withheld validation set.

our setting the source model is continuously updated between time steps and only a very limited number of samples are available from the newest distribution for adaptation.

5.8 Analysis of Importance Weighted Memory Sampling

We first perform an ablation study of our IWMS to show the effectiveness of the importance sampling. Then, we show our performances under different computational budgets and buffer sizes.

Analysis on forgetting over past samples In Figure 5.4, we report the backward transferability of the learned representation. This is done on a held-out, ordered validation set where the timestamp is used for ordering. On **CLOC**, all methods perform similarly due to poor data quality as reported in the Supplementary Material 5.11.9. On **CGLM**, our method not only surpasses the performance of others, but achieves $\sim 2\times$ the accuracy of the S4L, PL, Naïve and non-delayed Naïve baseline on CGLM. This means that the

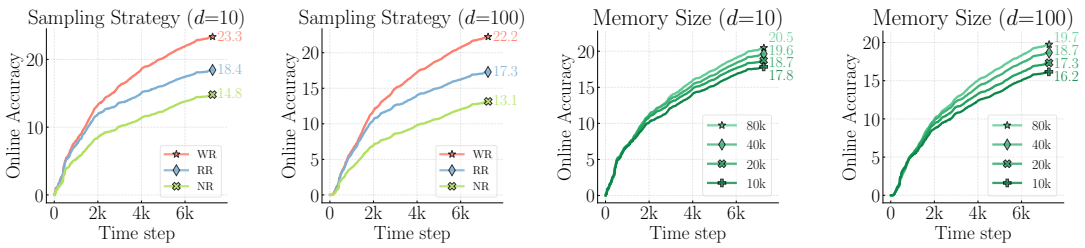


Figure 5.5: **Effect of sampling strategy (left), memory sizes (right).** We report the Online Accuracy under the least (top: $d = 10$) and the most challenging (bottom: $d = 100$) label delay scenarios on CGLM [170].

representation learned by our sampling technique is far more robust and generalises better not only to future but past examples as well. On **FMoW**, the best result is achieved by the Semi-Supervised methods, nevertheless our method outperforms the non-delayed Naïve in all scenarios. Finally, on **Yearbook** we see that under low label delay ($d = 10$) all results are clustered around 97%, however IWMS and Naïve performs best under larger delays ($d = 50, 10$).

Analysis on Memory Sampling Strategies. Note that while our method, IWMS is a prioritised sampling approach, it has some similarities to Naïve, except for the sampling strategy. While the Naïve method uses the most recent labeled data and a randomly sampled mini-batch from the memory buffer for each parameter update, our method provides a third option for constructing the training mini-batch, which picks the labeled memory sample that is most similar to the unlabeled data. When comparing sampling strategies, we refer to the newest batch of data as (N), the random batch of data as (R) and the importance weighted memory samples as (W).

In Figure 5.5 left, we first show that in both delay scenarios ($d = 10$ and $d = 100$) replacing the newest batch (N) with (W) results in almost doubling the performance: +8.5% and +9.1% improvement over Naïve, respectively. Interestingly enough, when we replace the (N) with uniformly sampled random buffer data (R) we report a significant increase in performance. We attribute this phenomenon to the detrimental effects of label delay:

even though Naïve uses the most recent supervised samples for training, the increasing discrepancy caused by the delay $d = 10$ and $d = 100$ forces the model to over-fit on the outdated distribution.

Analysis on the Memory Size. We study the influence of buffer size on our proposed IWMS. In particular, we show the performance of our algorithm under the buffer size from 10K to 80K in Figure 5.5 (right). Even though IWMS relies on the images sampled from the buffer to represent the new coming distribution, its performances remain robust under different buffer sizes: the largest performance gap between memory sizes of 10K and 80K is a marginal 2.5%.

5.9 Conclusion and Future Work

We motivate modeling real-world scenarios by introducing the label delay problem. We show how severely and unpredictably it hinders the performance of approaches which *naïvely* ignore the delay. To address the newfound challenges, we adopt the three most promising paradigms (Pseuodo-Labeling, S4L and TTA) and propose our own technique (IWMS). We provide extensive empirical evidence over four large-scale datasets posing various levels of distribution shifts, under multiple label delay scenarios and, most importantly, under normalised computational budget. IWMS simply stores and and reuses the embeddings of every observed sample during *memory rehearsal* where the most relevant labeled samples to the new unlabeled data are rehearsed. Due to its simplicity, the robustness against changes in the data distribution can be implemented very efficiently.

5.10 Acknowledgement

This work is supported by a UKRI grant Turing AI Fellowship (EP/W002981/1) and EPSRC/MURI grant: EP/N019474/1. Adel Bibi has received funding from the Amazon Research Awards. The authors thank Razvan Pascanu and João Henriques for their

insightful feedback. We also thank the Royal Academy of Engineering.

5.11 Supplementary Material

5.11.1 Dataset Statistics

We conduct our experiments on four large-scale online continual learning datasets, Continual Localization (CLOC) [20], Continual Google Landmarks (CGLM) [170], Functional Map of the World (FMoW) [43], and Yearbook [70]. The last two are adapted from the Wild-Time challenge [234]. More statistics of the benchmarks are in Supplementary.

The first, Continual Localization (CLOC) [20] which contains 39M images from 712 geolocation ranging from 2007 to 2014. The second is Continual Google Landmarks (CGLM) [170] which contains 430K images over 10788 classes. Followed by that, we report our experiments on Functional Map of the World (FMoW) [43] adapted from the Wild-Time challenge [234]. The dataset contains 14,696 satellite images, from 2002 to 2017, with the task of predicting the land type. Last, we show our results on the Yearbook dataset [70] containing 33,431 frontal-facing photos from American high-school yearbooks. The photos were taken in the time-period between 1930-2013 and represent changes in fashion, gender and ethnicity over the years. The task is a binary classification problem: predicting the gender of the student based on the photo.

5.11.2 Implementation Details of S4L

For integrating S4L methods, we adopt the most effective approach through iterative optimization of both supervised and unsupervised losses. This process involves optimising the standard Cross Entropy loss on labeled data (similar to Naïve) and minimising contrastive loss on unlabeled data, utilising a balanced approach until exhausting the computational budget. We conducted an exhaustive search over the possible multi-objective optimisation variants (such as iterative and joint optimisation) and determined the best result is

achieved when the contrastive loss is minimised separately for the first half of the parameter update steps, followed by minimising the supervised loss for the second half of the update steps. We report the best results across the three main families of contrastive losses, i.e.,, Deep Metric Learning Family (MoCo [88], SimCLR [34],and NNCLR [55]), Self-Distillation (BYOL [84] and SimSIAM [37], and DINO [25]), and Canonical Correlation Analysis (VICReg [15], BarlowTwins [241], SWAV [24], and W-MSE [56]).

For fair comparison, we normalise the computational complexity [169, 69] of the compared methods. We find that while SSL methods may take multiple forward passes, potentially with varying input sizes, the backward pass is consistently done only once among the variants, therefore, we choose the number of backward passes to measure the computational complexity of the resulting methods. According to this computational complexity constraint, Naïve augmented with SSL at each time step takes two backward passes, one for computing the gradients of the Cross Entropy over the labeled samples and one for the Contrastive Loss over the unlabeled samples, thus $\mathcal{C}_{\text{S4L}} = 2$.

5.11.3 Monotonous Online Accuracy Degradation

We argue the persistent drop in the Online Accuracy is due to the non-stationary property of the data distribution that creates a distribution shift. Our hypothesis is supported by the experimental results, illustrated in Figure 5.6: the Online Acc gradually decreases as the function of label delay d, at any given time step t. Furthermore, in Figure 5.7, we summarize the *final* Online Accuracy scores, i.e.,the Online Accuracy value at the final time step of each run

Our claims are reinforced by the findings of Yao [234]. A complementary argument is presented by Hammoud [87], stating that the underlying datasets have high temporal correlations across the labels, i.e.,images of the same categories arrive in bursts, allowing an online learning model to easily over-fit the label distribution even without using the

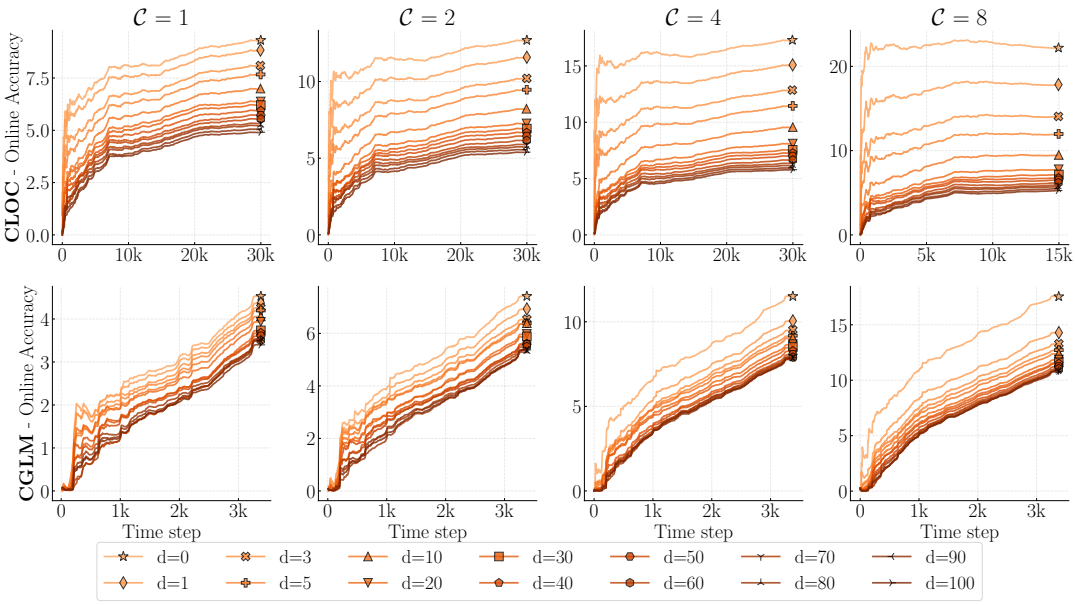


Figure 5.6: **Monotonous degradation of Online Accuracy** with regards to label delay d, over multiple datasets, CLOC [20] and CGLM [169], under various computational budgets, $\mathcal{C} = 1, 2, 4, 8$. The accuracy gradually drops at *every* time step t as the function of the label delay d. However the extent of the degradation is non-linear: The initial smallest increases in label delay have severe impact on the performance. In contrast, the rate of degradation slows down even for an order of magnitude larger increments when the labels are already delayed. See Figure 5.7 for the summary of the final values.

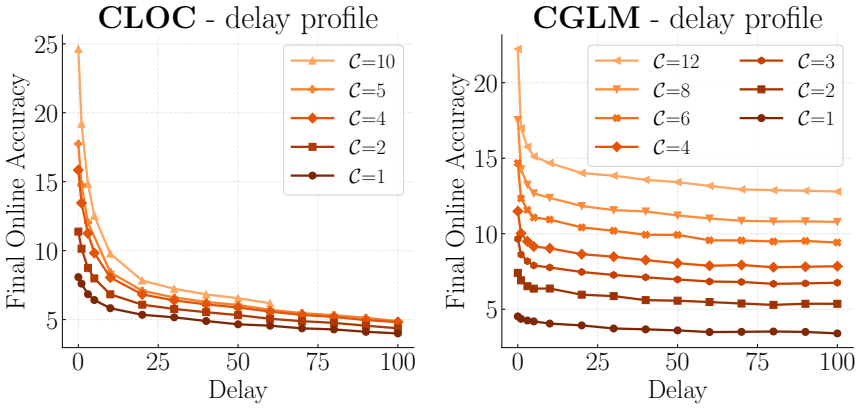


Figure 5.7: **Delay Profile.** Each trajectory shows the *Final Online Accuracy*, i.e.,the Online Accuracy evaluated at the last time step of each run, at a fixed computational budget $\mathcal{C}$. On both datasets the most severe accuracy degradation occurs in the first quarter ($d = 0 \rightarrow 25$). In contrast, on CGLM [169], the degradation is not significant in lower compute regimes $\mathcal{C} \leq 4$.

input images.

5.11.4 Qualitative Analysis of Label Delay

A case study of the distribution shift in the Yearbook experiments. While *Online Accuracy* is a well established performance metric for Online Continual Learning [20, 169, 170, 87], it can conceal some of the most important characteristics of the underlying dataset. To highlight a direct connection between the distribution shift and its immediate impact on the model performance, we illustrate the Top-1 Accuracy of the *current* batch at each time step in Figure 5.8. The experimental settings are identical to the main experiments on Naïve, detailed Section 5.6.2.

In this experiment, we describe several observations: first, the models perform at per-chance level accuracy until the first batch of labeled data arrives. Notice that the per-chance level is not 50% because the dataset is biased (contains more male than female portraits). However as the ratio improves over time, the random classifier's accuracy gets closer to 50%.

Before the distribution shift. In the smallest delay scenario (yellow curve), the delay is identical to a lag of three years between making the predictions and receiving the labels. Under such delay, the model quickly reaches close-to-optimal accuracy under just a few time steps and performs identically to the non-delayed counterpart (blue curve). In the moderate delay scenario (green curve), the model stays "idle" for a longer time (equivalent of 17 years) because of the delay of the first labeled batch. Nevertheless, the delayed model reaches similarly good performance after a time steps. Interestingly, the severely delayed model (red curve), exhibits a steep increase in performance, at $t = 1972$, exactly 34 years after observing the first sample ($t = 1938$).

During the distribution shift. The steep increase in the most severely delayed scenario (red curve) coincidentally overlaps with a major distribution shift in the appearance of one of the two classes. This shift simultaneously impacts the performance of all four models, however the rate at which their performance recovers differs, due to the label

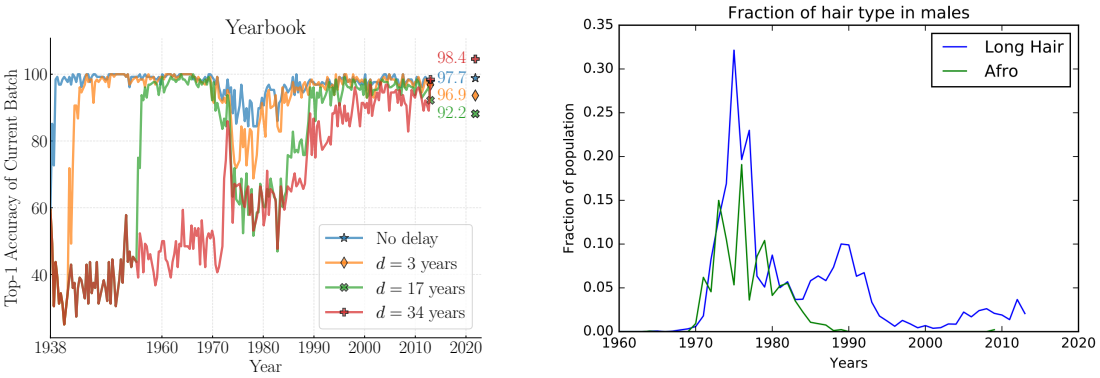


Figure 5.8: **(Left)** Top-1 Accuracy of Naïve on the *current* batch (of time step t) of Yearbook. **(Right)** Report from Ginosar [70] on "the fraction of male students with an afro or long hair." The drop in Top-1 Accuracy over time strongly correlates with the change in appearance of one of the two classes in the Yearbook [70] dataset. The larger the delay, the longer it takes to recover the close-to-perfect accuracy.

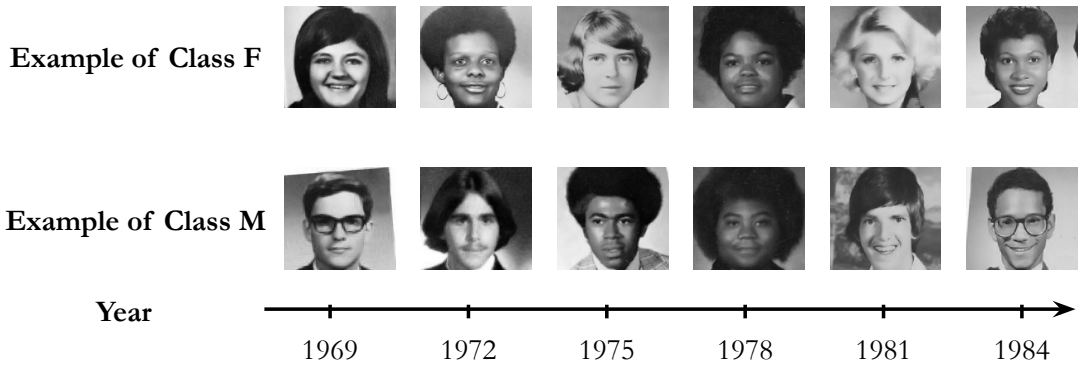


Figure 5.9: Examples from the Yearbook dataset [70] during the time where the visual appearance of men (bottom row) changes drastically resulting in an accuracy drop of an online classifier, regardless of the label delay.

delay. While in general it is an immensely difficult problem to detect and trace the changes of the data distribution, due to hidden latent variables (such as socio economic factors, genetic diversity of the population, cultural and political trends), Ginosar [70] identified and tracked many of such variables. One of these factors, namely the "fraction of male students with an afro or long hair", is highly correlated (in the temporal dimension) with the accuracy drop in our experiments, as illustrated in the right-hand side of Figure 5.8.

The reason behind the accuracy drop. In the qualitative experiments of the section

titled *"What time specific patterns is the classifier using for dating?"*, Ginosar [70] reports that convolutional neural networks, such as VGG [190], learn to extract features from the hairstyles of the subjects. Although the task is slightly different, classification of the year of the photograph, we hypothesise that one of the most discriminative features learned by the model are related to the hairstyles, as it is the most influential variable in terms of the accuracy of four independently trained models.

After the distribution shift. The recovery of the accuracy can be characterised by two factors: 1) the severity of the level degradation and 2) the duration of the recovery. Both factors show strong dependency on the underlying label delay factor: the larger the delay the larger the degradation and the longer the recovery length. Notice how closely the slightly delayed, yellow curve ($d = 3$ years) follows the non-delayed, blue curve in terms of duration, while the extent of the accuracy drop is larger for the delayed counterpart. On the other hand, the moderately and severely delayed models (green and red curves, respectively) apparently reach a lower-bound in performance degradation, where larger delay does not further reduce the accuracy. Nevertheless, the recovery of the severely delayed model is slower and occurs later than the moderately delayed model.

5.11.5 The impact of label delay on the scaling property of $\mathcal{C}$

The exploration of the impact of label delay on computational efficiency and accuracy across different settings reveals important insights into the performance and scalability of *Naïve*, an Experinece Replay model [27], which simply waits for every sample to receive its corresponding label before using it as a training data. In this section, through extensive **quantitative comparison** under different label delay d and computational budget $\mathcal{C}$ regimes, we offer a comprehensive overview of how these key factors interact to influence model performance on two large-scale datasets: CLOC [20] and CGLM [169].

Diminishing Returns. Figure 5.10 highlights the phenomenon of diminishing returns

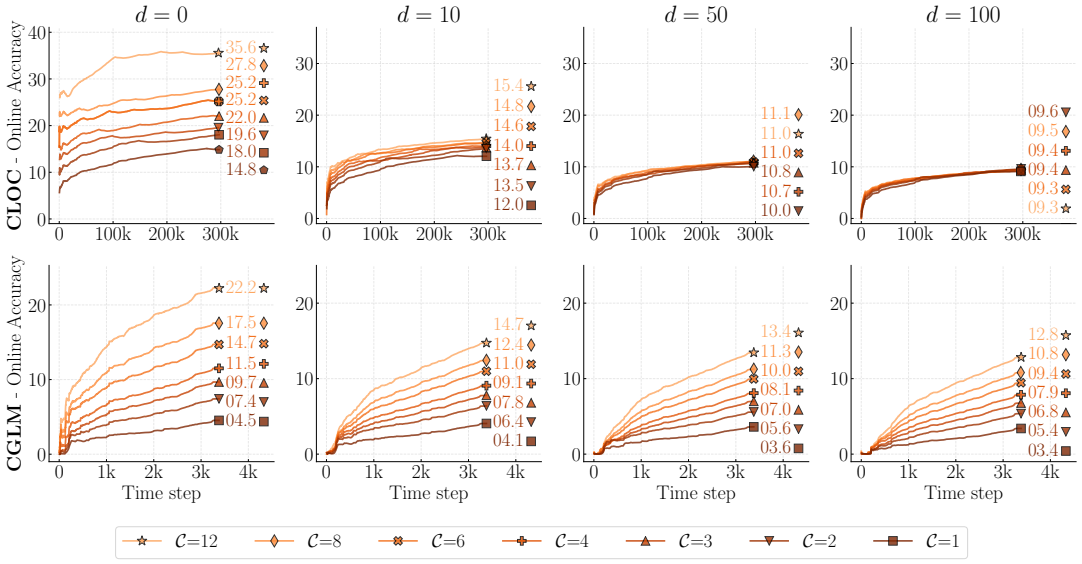


Figure 5.10: **Diminishing returns** of increasing the computational budget $\mathcal{C}$ over four label delay regimes $d = 0, 10, 50, 100$, on two datasets. While in many real-world scenarios simply increasing the budget $\mathcal{C}$ to improve the overall performance, when the labels are delayed the improvements *may* become marginal. Interestingly, this phenomena is emphasized on the CLOC [20] dataset, as the trajectories collapse to a single curve as the delay increases $d = 0 \rightarrow 100$. In contrast, on CGLM [169] the relative improvements, i.e.,the vertical distances between the lines, may shrink going from $d = 0 \rightarrow 10$, but stay consistent for $d = 10 \rightarrow 100$. The final scores are summarized by Figure 5.11.

on investment in the computational budget $\mathcal{C}$ across four different label delay regimes ($d = 0, 10, 50, 100$). Notably, while augmenting $\mathcal{C}$ typically yields performance improvements, these gains become increasingly marginal in the presence of delayed labels. The impact of label delay is markedly pronounced in the CLOC dataset, where the performance trajectories converge into a singular trend as the delay escalates from $d = 0$ to $d = 100$. Conversely, the CGLM dataset exhibits a contraction in the relative improvements (vertical distances between performance trajectories) as delay transitions from $d = 0$ to $d = 10$, yet these differences remain relatively stable for delays extending from $d = 10$ to $d = 100$.

Compute Scaling Profile. In Figure 5.11, the concept of a Compute Scaling Profile is introduced, displaying the *Final Online Accuracy* – the accuracy measured at the last time step of each run – for various levels of computational budget $\mathcal{C}$. This figure elucidates the sub-linear scaling of performance improvements with respect to incremental increases

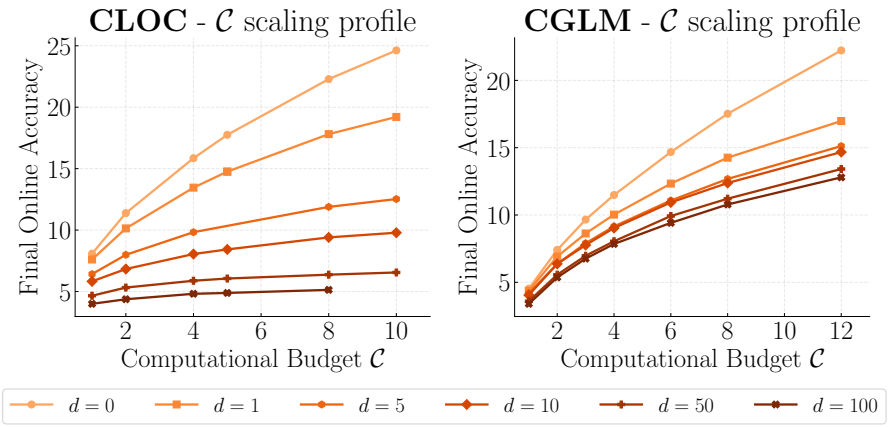


Figure 5.11: **Compute Scaling Profile.** Each trajectory shows the *Final Online Accuracy*, i.e.,the Online Accuracy evaluated at the last time step of each run, at a fixed computational budget $\mathcal{C}$. We show sub-linear improvement w.r.t. subsequent increases in $\mathcal{C}$, even in the non-delayed ($d = 0$) scenario. Moreover, the influence of label delay on the scaling property varies between the two datasets: while on CLOC [20] large delays ($d = 100$) prevent the model from benefiting from more parameter updates, on CGLM [169] label delay (for $d > 1$) only seems to offset the Final Online Accuracy, but does not impact rate of improvement.

in $\mathcal{C}$, a trend observable even without label delays ($d = 0$). The effects of label delay diverge between the datasets; CLOC sees a significant impediment to performance gains from additional parameter updates at high delays ($d = 100$), while in CGLM, the delay primarily shifts the Final Online Accuracy without diminishing the rate of improvement.

Gradual Monotonous Degradation. Figure 5.6 presents a nuanced view of how Online Accuracy monotonically degrades with increasing label delay (d) across different computational budgets ($\mathcal{C} = 1, 2, 4, 8$). This degradation is not linear; initial increments in label delay incur a steep decline in performance, whereas the rate of decline moderates for larger increments of delay, showcasing a nonlinear impact on model accuracy over time.

Delay Profile. Finally, Figure 5.7 encapsulates the Delay Profile, depicting the Final Online Accuracy at various computational budgets ($\mathcal{C}$). Both datasets exhibit the most substantial accuracy reductions in the initial quarter of delay increments ($d = 0 \rightarrow 25$). Interestingly, the CGLM dataset demonstrates a negligible degradation in lower computational regimes ($\mathcal{C} \leq 4$), indicating a potential resilience or adaptive capability under

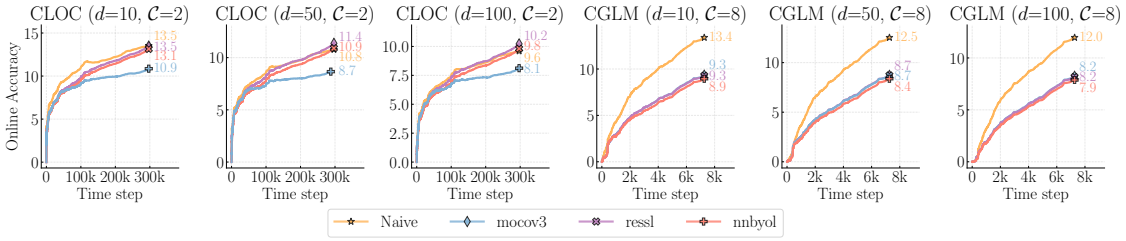


Figure 5.12: Comparison of the best performing SSL based methods after hyper-parameter tuning

specific conditions.

While increased computational budget generally improves the performance, the presence of label delays introduces a complex dynamic that can significantly hinder these benefits. The distinct behaviors observed across the CLOC and CGLM datasets further suggest that the dataset characteristics play a pivotal role in the decision making whether investment in additional compute is warranted or not. We suggest that such decision should be made on a case by case basis, rather than extrapolating from publicly available benchmarks.

5.11.6 Breakdown of SSL methods

In Figure 5.12 we show the performance of the best performing SSL based methods after hyper-parameter tuning. We observe that the performance of the SSL methods is highly dependent on the dataset and the delay setting. However, we apart from MoCo v3 [35], the methods perform similarly to Naïve on CLOC. On the other hand on CGLM they have insignificant differences in performance, but consistently underperform Naïve.

5.11.7 Breakdown of TTA methods

In Figure 5.13 we show the performance of the best performing TTA based methods after hyper-parameter tuning. We observe that the performance of the TTA methods are consistently worse than Naïve on both CLOC and CGLM, under all delay settings. We observe that in the most severe delay scenario ($d = 100$) the performance of EAT [163] and SAR [164] is comparable to Naïve on CLOC, while CoTTA [217] avoids the catastrophic

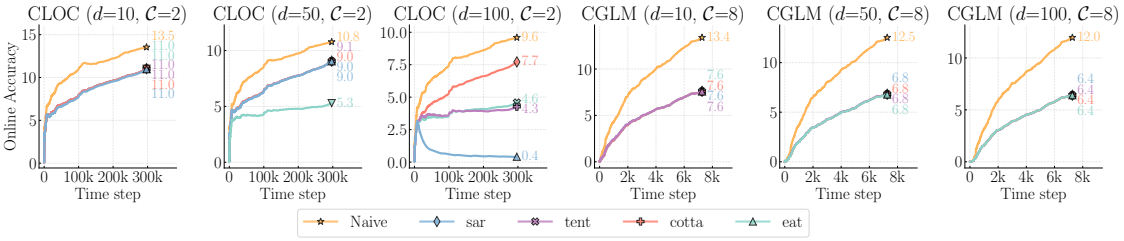


Figure 5.13: Comparison of the best performing TTA based methods after hyper-parameter tuning

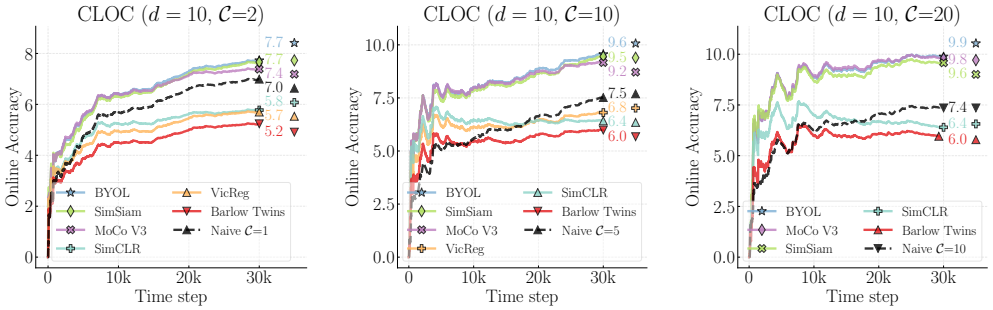


Figure 5.14: Detailed breakdown of various Self-Supervised Learning methods from each family. Results are shown across varying number of parameter updates $\mathcal{C} = 2, 10, 20$ under the $d = 10$ scenario.

performance drop.

5.11.8 Comparison of S4L to Naïve when using the same amount of supervised data

While in our main experiments S4L fails to outperform Naïve (in Section 5.7), we show that it is mostly due to the computational constraint of our experiments. In order to test our hypothesis, we run a series of experiments on the S4L variants, illustrated in Figure 5.14. In this experiment, instead of limiting the Computational Budget $\mathcal{C}$, we directly restrict the number of parameter updates to test if optimising the joint objective of Naïve and the given Self-Supervised Learning method improves the performance of the model at all. Our results indicate positive improvement over Naïve for MoCo-V3 [35], SimSiam [37] and BYOL [84] consistently across multiple settings with increasing number of parameter updates.

First, on the **left** hand side of Figure 5.14, both the Naïve and the S4L variants take only a single parameter update per time step (thus $\mathcal{C} = 2$ for all, except Naïve, where $\mathcal{C} = 1$). On the first 10% of the CLOC dataset [20], this results in a modest, nevertheless clear improvement over Naïve, up to +0.7%. Followed by that, in the **middle**, every model takes five parameter updates per time step. Notice that Naïve has a stricter computational budget, $\mathcal{C} = 5$, to match the rest of the experiments. Consistently with our findings in Section 5.11.5, Naïve only benefited marginally from the increase in compute, due to diminishing returns, 7.0% $\rightarrow$ 7.5%. On the contrary, the previously highlighted S4L variants show a larger improvement over the increase in number of updates, e.g.,7.7% $\rightarrow$ 9.6%. Consequently, this increases the gap between the Naïve and the S4L methods. Finally, on the **right** hand side of the figure, we show when the models are updated ten times in each time step, the improvement plateaues for both the Naïve and the S4L variants.

Conclusion of this set of experiments is two-fold: when granted equal amount of parameter updates, S4L methods outperform Naïve across different settings. However, computing the parameter gradients w.r.t. the joint objective of S4L costs approximately twice the amount that of the Naïve: $\mathcal{C}_{\text{S4L}} \simeq 2 \times \mathcal{C}_{\text{Naïve}}$. Due to the well-known property of Self-Supervised Learning methods, *sample inefficiency*, our main experiments show that "spending" the compute on more frequent Naïve updates is more beneficial than optimising the joint S4L objective, even when the training data is heavily delayed.

5.11.9 Examples of the Importance Weighted Memory Sampling on CLOC

On CLOC, we report similar scores to Naïve due to high noise in the data. To provide evidence for our claims we visualize the supervised data sampled from the memory buffer by our Importance Weighted Memory Sampling method. In Figure 5.15, we show that our method is capable of guessing the correct location of the unsupervised sample (the left hand side of the image pairs) and recalling a relevant sample from memory. In contrast,

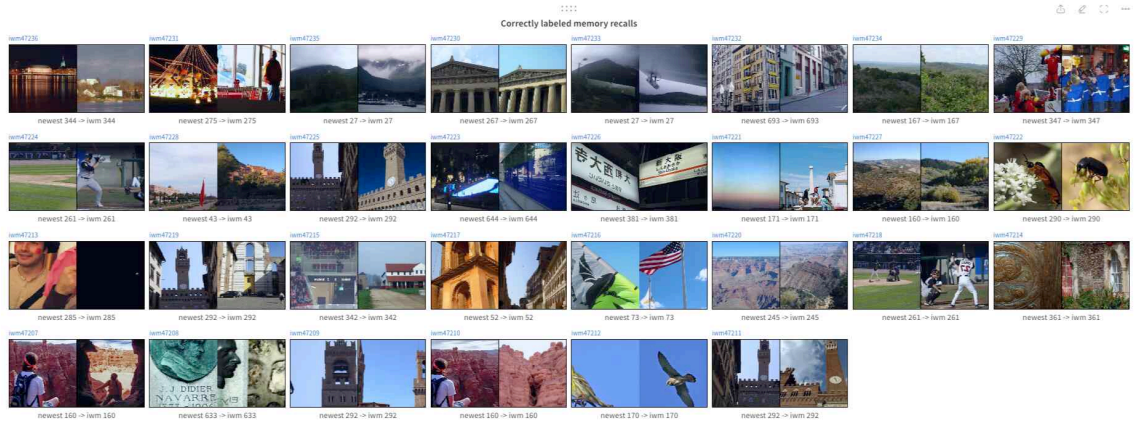


Figure 5.15: **Correctly labeled memory recalls.** In the subfigure's caption "Newest" refers to the newest unsupervised image observed by the model and "iwm" refers to the sample drawn from the memory by our proposed sampling method. The numbers refer to the corresponding true label IDs.

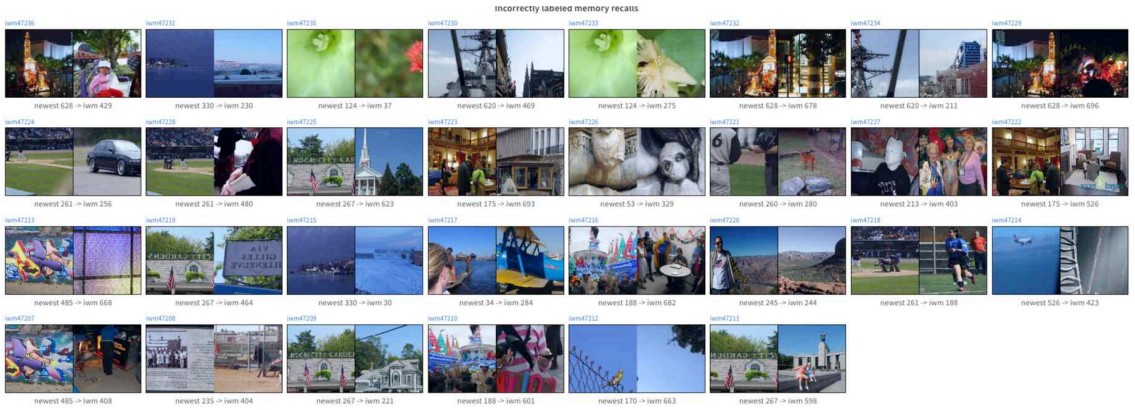


Figure 5.16: **Incorrectly labeled memory recalls.** In the subfigure's caption "Newest" refers to the newest unsupervised image observed by the model and "iwm" refers to the sample drawn from the memory by our proposed sampling method. The numbers refer to the corresponding true label IDs.

the incorrect memory recalls hurt the performance even though the content of the samples might match. We illustrate such cases in Figure 5.16, where it is obvious that in some cases the underlying image content has no information related to the location where the picture was taken at. In such scenarios, the only way a classifier can correctly predict the labels is by exploiting label correlations, e.g.,classifying all close-up images of flowers to belong to the same geo-location, even though the flowers are not unique to the location itself. Or consider the pictures taken at social gatherings (second row, second column

from the right), where a delayed classifier without being exposed to that specific series of images has no reason to correctly predict the location ID. Our claims are reinforced by the findings of [87].

5.11.10 Visual Explanation of our Experimental Framework

We provide visual guides for explaining our experimental framework. In Figure 5.17, we emphasize the main difference between our setting and the general setting of partially labeled data-streams: while prior art does not differentiate between old and new unsupervised data, our work focuses specifically on the scenario when *all* unsupervised data is newer then the supervised data. In Figure 5.18, we show the two types of data that our models work with: outdated supervised data, and newer, unsupervised data. The task is to find a way to utilize the newer unsupervised data to augment the Naïve approach, that simply just waits for the labels to become available to update its parameters. The most challenging component in our experiments is the computational budget factor that allows only a certain amount of forward and backward passes through the backbone.

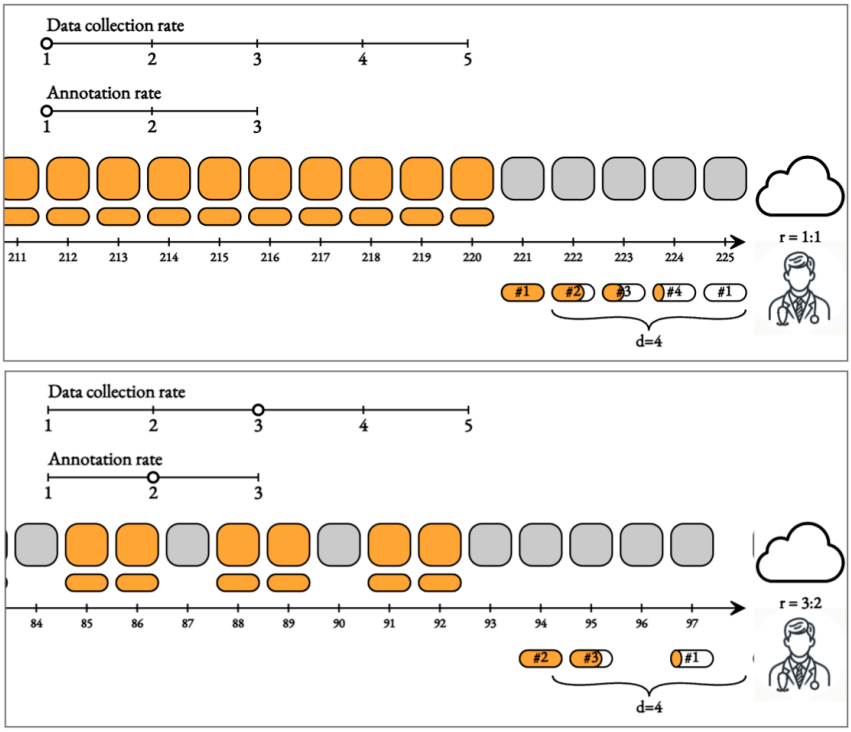


Figure 5.17: **Our experimental setup (top)**: After a fixed amount of time steps all labels become available. This allows us to focus on utilizing future unsupervised samples effectively. **Partial labeling setup (bottom)**: in the generic setting, when the data collection rate is higher than the annotation rate, some samples might never receive labels.

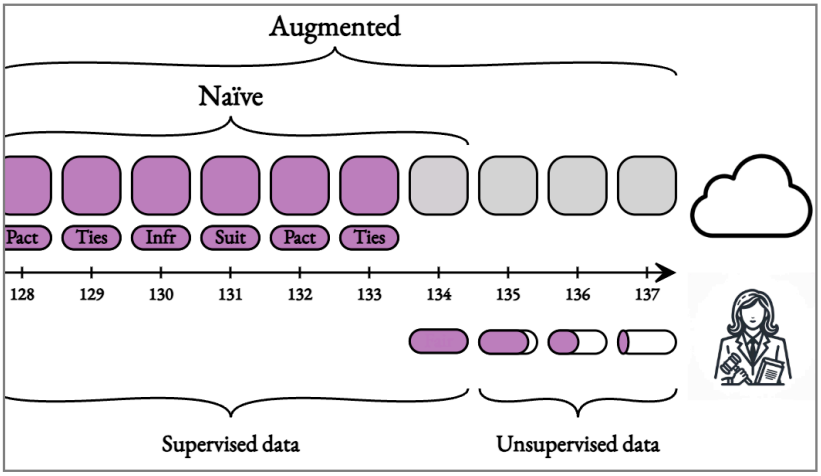


Figure 5.18: **Experimental setup**: in our experiments we show how increased label delay affects the Naïve approach that simply just waits for the labels to arrive. To counter the performance degradation we evaluate three paradigms (Self-Supervised Learning, Test-Time Adaptation, Importance Weighted Memory Sampling) that can augment the Naïve method by utilizing the newer, unsupervised data.

5.11.11 Two-stage vs single-shot sample selection

In Section 5.5, we outlined our proposed two-stage sample selection method, IWMS. In this experiment we show empirical evidence and analysis on why predicting the class-labels first then doing similarity matching leads to better results than simply using a similarity score over all the memory samples. In Figure 5.19, we illustrate the evolution of the similarity scores of the two matching policies. On the left, the matching is done purely based on the similarity scores, whereas on the right only those samples were compared against those memory samples whose labels match the predicted labels. In the middle plot, we show that by implementing the two-stage selection, we increase the effectivity of the similarity matching by a large margin, +7.8%.

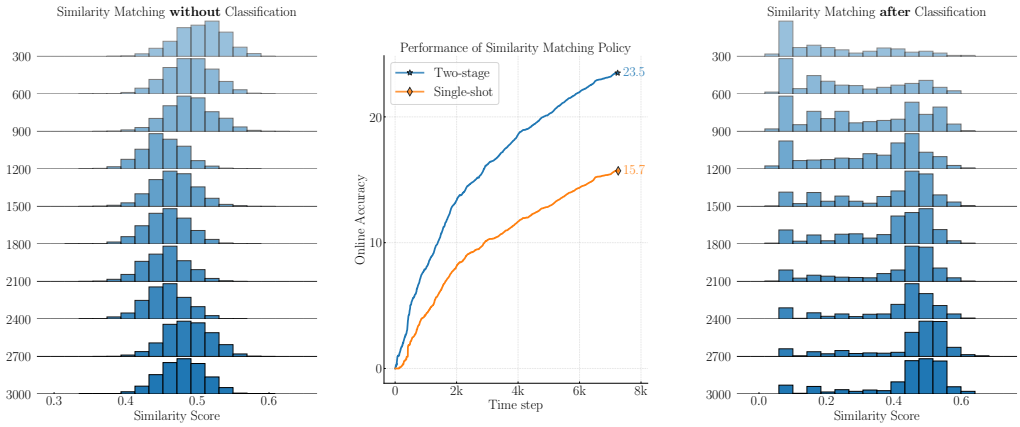


Figure 5.19: The evolution of Similarity Scores between the unsupervised and memory samples over time. On each histogram, we plot the distribution of the cosine similarity scores between the feature representations of the yet to be labeled samples and the samples in the memory that already received their labels. On the top row we show the initial distributions and going from top down, the evolution of the two distribution is illustrated over the time steps.

5.11.12 Extended Literature Review on Online Learning

Online Learning vs Online Continual Learning: Online Learning and Online Continual Learning, while both involve learning from data arriving sequentially, differ fundamentally in scope. Online Learning typically deals with single-task streams, often assumed to be from an i.i.d. distribution, as outlined in section 2.3 of [41] and the introduction of [60]. In contrast, Online Continual Learning (OCL) is more concerned with non-stationary streams that undergo frequent changes in distribution, where mitigating forgetting is one of several challenges [60, 146, 14].

Non-i.i.d. distribution of unsupervised data: While our work focuses on evolving distributions, work such as Weinberger [220] and Flaspohler [61] only considers label delay while the distribution a time-invariant, consequently completely omitting the problem of distribution shift. Majority of the prior online learning work [234, 155, 76, 64, 168, 99, 193, 220, 120, 172, 61] ignores the difference between past and future unsupervised data. In our proposal, all unsupervised data is newer than the last supervised data. We illustrate the difference between the two different types of unsupervised data in Figure 5.17.

Considering catastrophic forgetting: Continual Learning, both online and offline, is concerned about performing well on previously observed data, often referred to as backward transfer of the learned representations [234, 155, 76, 64, 168, 99, 193, 220, 120, 172, 61]. This is different from Online learning where the problem of forgetting is not considered. Even in more recent Online Continual Learning work, backward transfer have been given slightly lower priority [87, 69, 20] where the authors have reported them only in the appendix.

5.12 Conclusion

We present a novel experimental setup for online continual learning that incorporates the observation of the time gap between input samples and their corresponding supervised labels. This framework enables the model to leverage unlabeled data in an unsupervised manner during the initial stages and seamlessly incorporate labeled data as it becomes available over time. By bridging the gap between unsupervised learning and online continual learning, our methods adapt dynamically to evolving data distributions, achieve competitive performance, and contribute to the development of more flexible and efficient learning systems in real-world scenarios.

Conclusion

The field of machine learning has witnessed remarkable progress in recent years, with models becoming increasingly capable of tackling complex, real-world challenges. This thesis has explored the critical issue of distribution shift under strong computational constraints - a pervasive challenge that hinders the deployment of machine learning models in many practical applications.

Through a progression of contributions, this work has demonstrated the effectiveness of unsupervised methods in bridging distribution discrepancies and enabling efficient, adaptable models suitable for resource-constrained environments. From unsupervised clustering for efficient image recognition and generation to unsupervised domain adaptation for cross-domain object detection, the techniques presented showcase the power of leveraging unlabeled data to improve model robustness and generalization.

6.1 Summary of Contributions

Chapter 2: Unsupervised Clustering for Efficient Image Recognition

In this paper we demonstrate the superiority of networks trained on a subset of the training set holding similar properties, which we refer to as *local experts*. We address the two main challenges of training and employing local experts in real life scenarios, where subset

labels are not available during test nor training time. Followed by that, we propose a method, called Diversified Dynamic Routing that is capable of jointly learning local experts and subset labels without supervision. In a controlled study, where the subset labels are known, we showed that we can recover the original subset labels with 98.2% accuracy while maintaining the performance of a hypothetical *Oracle* model in terms of both accuracy and efficiency.

To analyse how well this improvement translates to real life problems we conducted extensive experiments on complex computer vision tasks such as segmenting street objects on images taken from the driver's perspective, as well as detecting common objects in both indoor and outdoor scenes. In each scenario we demonstrate that our method outperforms Dynamic Routing [129].

Even though this approach is powerful in a sense that it could improve on a strong baseline, we are aware that the clustering method still assumes subsets of *equal* and more importantly *sufficient* size. If the dataset is significantly imbalanced w.r.t. local biases the K-means approach might fail. One further limitation is that if the subsets are too small for the *local experts* to learn generalizable representations our approach might also fail to generalize. Finally, since the search space of the architectures in this work is defined by Dynamic Routing [129] which is heavily focused on scale-varience. We believe that our work can be further generalized by analyzing and resolving the challenges mentioned above.

Chapter 3: Unsupervised Clustering for Image Generation

We conclude that it is not necessary for a generator to have equal capacity adversary to converge, meaning that the standard GAN training procedure could be enhanced with multiple (and even weaker) discriminators specialized only in attracting the model distribution of the generator to their corresponding modes.

DoPaNet is proven experimentally to utilize the capability of multiple discriminators by partitioning the target distributions into several identifiable modes and making each discriminator work on a separate mode. Thus, it reduces the complexity of the modes to be learnt by each discriminator. We show qualitatively and quantitatively that DoPaNet is able to better cover the real distribution. We observe that the generator is also able to sample from different identifiable modes of the data distribution given the corresponding code vectors.

Chapter 4: Unsupervised Domain Adaptation for Cross-Domain Object Detection

In this work we have addressed the problem of cross-domain object detection. After conducting extensive studies on previous approaches, we categorized different components into 3 classes: image, feature and output level domain adaptation. Our proposed method is the first to successfully employ all levels of domain adaptation. Our method consists of 3 stages: 1) translate labeled source images to make them appear similar to target images 2) train a teacher network on translated images while aligning its features to the target domain 3) use the teacher model to generate pseudo-labels, then train a student model on the translated source images and the pseudo-labeled target images. Compared to recent state-of-the art works, our method uses simple adaptation techniques at each stage and still outperforms complex algorithms on every commonly used benchmark.

Chapter 5: Label Delay in Online Continual Learning

We motivate modeling real-world scenarios by introducing the label delay problem. We show how severely and unpredictably it hinders the performance of approaches which *naïvely* ignore the delay. To address the newfound challenges, we adopt the three most promising paradigms (Pseuodo-Labeling, S4L and TTA) and propose our own technique (IWMS). We provide extensive empirical evidence over four large-scale datasets posing

various levels of distribution shifts, under multiple label delay scenarios and, most importantly, under normalised computational budget. IWMS simply stores and and reuses the embeddings of every observed sample during *memory rehearsal* where the most relevant labeled samples to the new unlabeled data are rehearsed. Due to its simplicity, the robustness against changes in the data distribution can be implemented very efficiently.

6.2 Future challenges

Scalability Studies

The scalability and computational efficiency of unsupervised learning methods remain significant challenges as datasets continue to grow in size and complexity. While this thesis has demonstrated the effectiveness of unsupervised methods in addressing distribution shift, further research is needed to develop even more computationally efficient techniques that can handle larger datasets and more complex models. This may involve exploring novel architectures, optimization strategies, or data preprocessing techniques that can reduce the computational burden while preserving the benefits of unsupervised learning. Additionally, investigating the trade-offs between model performance and computational efficiency will be crucial in determining the practical applicability of unsupervised methods in real-world settings.

Bootstrapping from Expert Knowledge

Another important challenge lies in integrating domain knowledge into unsupervised learning methods. While unsupervised learning allows models to discover patterns and structures in data without explicit labels, incorporating domain-specific knowledge can potentially lead to more accurate and interpretable models. This may involve exploring ways to combine unsupervised learning with expert feedback or guidance, such as through active learning or semi-supervised learning approaches. Additionally, developing methods to

interpret and explain the learned representations in unsupervised models can help build trust and facilitate the adoption of these techniques in domain-specific applications. Further research into effective strategies for integrating domain knowledge and improving the interpretability of unsupervised models will be crucial in addressing this challenge.

Federated Learning

Federated learning has emerged as a paradigm for training machine learning models on decentralized data while preserving the privacy of individual users or institutions. However, most existing federated learning approaches rely on supervised learning, which requires labeled data from all participating clients. Extending unsupervised learning methods to the federated setting could enable the discovery of patterns and structures in decentralized data without the need for explicit labels, opening up new possibilities for collaborative learning and analytics.

However, federated unsupervised learning also poses several challenges, particularly in the presence of computational constraints and label delay. Developing communication-efficient algorithms that can minimize the amount of data exchanged between clients and the central server is crucial for scaling federated unsupervised learning to large-scale distributed systems. Additionally, the limited computational resources available on individual devices may require the development of lightweight unsupervised learning methods that can operate under strict memory and processing constraints.

Another key challenge in federated unsupervised learning is dealing with label delay and distribution shift. In many real-world scenarios, labels may arrive with a significant delay or may not be available at all, requiring unsupervised methods to adapt to evolving data distributions and learn from unlabeled data. Investigating techniques for aligning and combining representations learned from multiple clients in the presence of distribution shift and label delay is an important direction for future research.

Addressing these challenges will require a combination of advances in unsupervised learning, federated optimization, and computational efficiency. Techniques such as model compression, quantization, and pruning may help reduce the computational cost of unsupervised learning methods, while communication-efficient aggregation protocols and adaptive learning rates may help optimize the performance of federated algorithms. Incorporating techniques from domain adaptation and transfer learning may also help mitigate the effects of distribution shift and label delay in federated unsupervised learning.

6.3 Concluding remarks

As machine learning continues to evolve and tackle increasingly complex real-world challenges, the importance of unsupervised learning will only grow. This thesis represents a significant step forward in the development of efficient and adaptable unsupervised methods, paving the way for future research and applications in this exciting field. By addressing the challenges of distribution shift under computational constraints, this work contributes to the democratization of machine learning, making it more accessible and applicable to a broader range of real-world problems.

In conclusion, this thesis has explored the use of unsupervised methods to bridge distribution discrepancies in deep learning vision tasks, demonstrating their effectiveness and efficiency in increasingly complex and realistic settings. The contributions of this work have the potential to impact a wide range of real-world applications and inspire future research in unsupervised learning. As the field continues to advance and tackle ever more challenging problems, the importance of unsupervised learning in enabling robust, efficient, and adaptable models will only continue to grow.

Bibliography

[1] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *arXiv preprint arXiv:2403.13187*, 2024.

[2] Motasem Alfarra, Hani Itani, Alejandro Pardo, Shyma Alhuwaider, Merey Ramazanova, Juan C Pérez, Zhipeng Cai, Matthias Müller, and Bernard Ghanem. Revisiting test time adaptation under online evaluation. *arXiv preprint arXiv:2304.04795*, 2023.

[3] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[4] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 11849–11860. Curran Associates, Inc., 2019.

[5] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[7] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

[8] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.

[9] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[10] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (GANs). In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[11] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*, 2017.

[12] John O Awoyemi, Adebayo O Adetunmbi, and Samuel A Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 international conference on computing networking and informatics (ICCNI)*, pages 1–9. IEEE, 2017.

[13] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

[14] Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on a contaminated data stream with

blurry task boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9275–9284, 2022.

[15] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[16] Yohann Benchetrit, Hubert Banville, and Jean-Rémi King. Brain decoding: toward real-time reconstruction of visual perception. *arXiv preprint arXiv:2310.19812*, 2023.

[17] David Berthelot, Thomas Schumm, and Luke Metz. Began: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[18] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations (ICLR)*, 2021.

[19] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11457–11466, 2019.

[20] Zhipeng Cai, Ozan Sener, and Vladlen Koltun. Online continual learning with natural distribution shifts: An empirical study with visual data. In *International Conference on Computer Vision (ICCV)*, 2021.

[21] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020.

[22] Fabio Maria Cariucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota

Bulo. Autodial: Automatic domain alignment layers. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5077–5085. IEEE, 2017.

[23] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[24] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignment. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[25] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Eemerging properties in self-supervised vision transformers. In *International Conference on Computer Vision (ICCV)*, 2021.

[26] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018.

[27] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc' Aurelio Ranzato. Continual learning with tiny episodic memories. *arXiv preprint*, 2019.

[28] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. In *ICLR*, 2017.

[29] Liang-Chieh Chen, Maxwell D Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jonathon Shlens. Searching for efficient multi-scale architectures for dense image prediction. *arXiv:1809.04184*, 2018.

[30] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional

nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[31] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017.

[32] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, 2018.

[33] Luyang Chen, Markus Pelger, and Jason Zhu. Deep learning in asset pricing. *Management Science*, 70(2):714–750, 2024.

[34] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[35] X. Chen, S. Xie, and K. He. An empirical study of training self-supervised vision transformers. In *International Conference on Computer Vision (ICCV)*, 2021.

[36] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

[37] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[38] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1992–2001, 2017.

[39] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.

[40] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2018.

[41] Zhiyuan Chen and Bing Liu. *Lifelong machine learning*, volume 1. Springer, 2018.

[42] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[43] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6172–6180, 2018.

[44] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[45] Botos Csaba, Wenxuan Zhang, Matthias Müller, Ser-Nam Lim, Mohamed Elhoseiny, Philip Torr, and Adel Bibi. Label delay in continual learning. *arXiv preprint arXiv:2312.00923*, 2023.

[46] Gabriela Csurka. A comprehensive survey on domain adaptation for visual applications. In *Domain adaptation in computer vision applications*, pages 1–35. Springer, 2017.

[47] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-

based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[48] Alexandre Défossez, Charlotte Caucheteux, Jérémy Rapin, Ori Kabeli, and Jean-Rémi King. Decoding speech perception from non-invasive brain recordings. *Nature Machine Intelligence*, 5(10):1097–1107, 2023.

[49] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[50] Mingyu Ding, Xiaochen Lian, Linjie Yang, Peng Wang, Xiaojie Jin, Zhiwu Lu, and Ping Luo. Hr-nas: searching efficient high-resolution neural architectures with lightweight transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[51] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ICLR*, 2017.

[52] Lixin Duan, Ivor W Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012.

[53] Lixin Duan, Dong Xu, Ivor Wai-Hung Tsang, and Jiebo Luo. Visual event recognition in videos by learning from web data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1667–1680, 2011.

[54] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. In *ICLR*, 2017.

[55] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning

of visual representations. In *International Conference on Computer Vision (ICCV)*, 2021.

[56] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning (ICML)*, 2021.

[57] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[58] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

[59] Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[60] Enrico Fini, Stéphane Lathuiliere, Enver Sangineto, Moin Nabi, and Elisa Ricci. Online continual learning under extreme memory constraints. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 720–735. Springer, 2020.

[61] Genevieve E Flaspohler, Francesco Orabona, Judah Cohen, Soukayna Mouatadid, Miruna Oprescu, Paulo Orenstein, and Lester Mackey. Online learning with optimism and delay. In *International Conference on Machine Learning*. PMLR, 2021.

[62] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[63] 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*, 17(1):2096–2030, 2016.

[64] Haoran Gao and Zhijun Ding. A novel machine learning method for delayed labels. In *2022 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2022.

[65] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[66] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[67] Arnab Ghosh, Viveka Kulharia, and Vinay Namboodiri. Message passing multi-agent gans. *arXiv preprint arXiv:1612.01294*, 2016.

[68] Arnab Ghosh, Viveka Kulharia, Vinay Namboodiri, Philip HS Torr, and Puneet K Dokania. Multi-agent diverse generative adversarial networks. *arXiv preprint arXiv:1704.02906*, 1(4), 2017.

[69] Yasir Ghunaim, Adel Bibi, Kumail Alhamoud, Motasem Alfarra, Hasan Abed Al Kader Hammoud, Ameya Prabhu, Philip HS Torr, and Bernard Ghanem. Real-time evaluation in online continual learning: A new hope. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[70] Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei A Efros. A century of portraits: A visual historical record of american high school yearbooks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1–7, 2015.

[71] Ross Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, 2015.

[72] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[73] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[74] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[75] George Gkotsis, Anika Oellrich, Sumithra Velupillai, Maria Liakata, Tim JP Hubbard, Richard JB Dobson, and Rina Dutta. Characterisation of mental health conditions in social media using informed deep learning. *Scientific reports*, 7(1):45141, 2017.

[76] Heitor Murilo Gomes, Maciej Grzenda, Rodrigo Mello, Jesse Read, Minh Huong Le Nguyen, and Albert Bifet. A survey on semi-supervised learning for delayed partially labelled data streams. *ACM Computing Surveys*, 2022.

[77] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of machine learning research*, 12(Jul):2211–2268, 2011.

[78] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012.

[79] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[80] 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, 2014.

[81] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[82] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006. IEEE, 2011.

[83] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 2021.

[84] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[85] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[86] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[87] Hasan Abed Al Kader Hammoud, Ameya Prabhu, Ser-Nam Lim, Philip H. S. Torr, Adel Bibi, and Bernard Ghanem. Rapid adaptation in online continual learning: Are

we evaluating it right? In *International Conference on Computer Vision (ICCV)*, 2023.

[88] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[89] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[90] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[91] Zhenwei He and Lei Zhang. Multi-adversarial faster-rcnn for unrestricted object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6668–6677, 2019.

[92] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

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

[94] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[95] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko,

Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[96] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[97] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018.

[98] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–5, 2019.

[99] Hanqing Hu and Mehmed Kantardzic. Sliding reservoir approach for delayed labeling in streaming data classification. In *2017 Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[100] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2007.

[101] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[102] Sheng-Wei Huang, Che-Tsung Lin, Shu-Ping Chen, Yen-Yi Wu, Po-Hao Hsu, and Shang-Hong Lai. Auggan: Cross domain adaptation with gan-based data augmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 718–731, 2018.

[103] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018.

[104] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. *arXiv:1811.11721*, 2018.

[105] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018.

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

[107] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019.

[108] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[109] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[110] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Deven-

dra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[111] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*, 2016.

[112] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International Conference on Machine Learning*, pages 1453–1461. PMLR, 2013.

[113] Felix Juefei-Xu, Vishnu Naresh Boddeti, and Marios Savvides. Gang of gans: Generative adversarial networks with maximum margin ranking. *arXiv preprint arXiv:1704.04865*, 2017.

[114] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[115] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G Macready. A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 480–490, 2019.

[116] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12456–12465, 2019.

[117] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[118] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning Deep Learning Workshop*, 2015.

[119] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR 2011*, pages 1785–1792. IEEE, 2011.

[120] Ludmila I Kuncheva and J Salvador Sánchez. Nearest neighbour classifiers for streaming data with delayed labelling. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008.

[121] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *ArXiv*, abs/1610.02242, 2016.

[122] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

[123] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[124] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2, 2013.

[125] Martin Leo, Suneel Sharma, and Koilakuntla Maddulety. Machine learning in banking risk management: A literature review. *Risks*, 7(1):29, 2019.

[126] Chongxuan Li, Kun Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *NIPS*, 2017.

[127] Wen Li, Zheng Xu, Dong Xu, Dengxin Dai, and Luc Van Gool. Domain generalization and adaptation using low rank exemplar svms. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1114–1127, 2017.

[128] Xin Li, Yiming Zhou, Zheng Pan, and Jiashi Feng. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[129] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[130] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[131] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *arXiv preprint arXiv:2303.15361*, 2023.

[132] 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 (ICML)*, 2020.

[133] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[134] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[135] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss

for dense object detection. In *IEEE International Conference on Computer Vision*, 2017.

[136] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

[137] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

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

[139] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[140] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017.

[141] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *NIPS*, 2016.

[142] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[143] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[144] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[145] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in neural information processing systems*, pages 136–144, 2016.

[146] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[147] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, et al. Deepseek-vl: Towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

[148] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12012–12038, 2021.

[149] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *The fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[150] Tharsanee Maganathan, Soundariya Senthilkumar, and Vishnupriya Balakrishnan. Machine learning and data analytics for environmental science: a review, prospects and challenges. In *IOP conference series: materials science and engineering*, volume 955. IOP Publishing, 2020.

[151] Eric Martínez. Re-evaluating gpt-4' s bar exam performance. *Artificial Intelligence and Law*, pages 1–24, 2024.

[152] Imran Md Jelas, Mohd Asyraf Zulkifley, Mardina Abdullah, and Martin Spraggon.

Deforestation detection using deep learning-based semantic segmentation techniques: a systematic review. *Frontiers in Forests and Global Change*, 7:1300060, 2024.

[153] Lars Mescheder. On the convergence properties of gan training. *arXiv preprint arXiv:1801.04406*, 2018.

[154] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, 2018.

[155] Chris Mesterharm. On-line learning with delayed label feedback. In *International Conference on Algorithmic Learning Theory*, 2005.

[156] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017.

[157] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[158] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *Neural Information Processing Systems Deep Learning Workshop*, 2013.

[159] 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*, pages 5715–5725, 2017.

[160] Ravi Teja Mullapudi, William R Mark, Noam Shazeer, and Kayvon Fatahalian. Hydranets: Specialized dynamic architectures for efficient inference. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[161] Jamal Abdul Nasir, Osama Subhani Khan, and Iraklis Varlamis. Fake news de-

tection: A hybrid cnn-rnn based deep learning approach. *International Journal of Information Management Data Insights*, 1(1):100007, 2021.

[162] Vladimir Nekrasov, Hao Chen, Chunhua Shen, and Ian Reid. Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[163] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning (ICML)*, 2022.

[164] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations (ICLR)*, 2023.

[165] Julio-Omar Palacio-Niño and Fernando Berzal. Evaluation metrics for unsupervised learning algorithms. *arXiv preprint arXiv:1905.05667*, 2019.

[166] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 754–763, 2017.

[167] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE international conference on computer vision*, pages 4930–4939, 2017.

[168] Joshua Plasse and Niall Adams. Handling delayed labels in temporally evolving data streams. In *2016 IEEE International Conference on Big Data (Big Data)*, 2016.

[169] Ameya Prabhu, Hasan Abed Al Kader Hammoud, Puneet K Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, and Adel Bibi. Computationally budgeted continual learning: What does matter? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[170] Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, and Ozan Sener. Online continual learning without the storage constraint. *arXiv preprint arXiv:2305.09253*, 2023.

[171] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European Conference on Computer Vision (EECV)*, 2020.

[172] Kent Quanrud and Daniel Khashabi. Online learning with adversarial delays. *Advances in neural information processing systems*, 2015.

[173] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[174] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[175] Anant Raj, Vinay P Namboodiri, and Tinne Tuytelaars. Subspace alignment based domain adaptation for rcnn detector. *arXiv preprint arXiv:1507.05578*, 2015.

[176] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[177] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[178] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[179] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[180] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. *arXiv:1511.05939*, 2015.

[181] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[182] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019.

[183] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018.

[184] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016.

[185] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[186] Albert Shaw, Daniel Hunter, Forrest Landola, and Sammy Sidhu. Squeezenas: Fast neural architecture search for faster semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[187] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 299–315, 2018.

[188] 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.

[189] Prashant Shrestha, Sanskar Amgain, Bidur Khanal, Cristian A Linte, and Binod Bhattarai. Medical vision language pretraining: A survey. *arXiv preprint arXiv:2312.06224*, 2023.

[190] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[191] Woo-Sik Son and Young-Jai Park. Delayed feedback on the dynamical model of a financial system. *Chaos, Solitons & Fractals*, 44(4-5):208–217, 2011.

[192] Lin Song, Yanwei Li, Zeming Li, Gang Yu, Hongbin Sun, Jian Sun, and Nanning Zheng. Learnable tree filter for structure-preserving feature transform. In *Advances in Neural Information Processing Systems*, 2019.

[193] Vinicius MA Souza, Diego F Silva, Gustavo EAPA Batista, and João Gama. Classification of evolving data streams with infinitely delayed labels. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015.

[194] Akash Srivastava, Lazar Valkoz, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318, 2017.

[195] Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[196] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[197] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain

adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[198] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.

[199] Richard Sutton. The bitter lesson. *Incomplete Ideas (blog)*, 13(1):38, 2019.

[200] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[201] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

[202] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. *arXiv preprint arXiv:1911.09070*, 2019.

[203] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[204] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[205] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[206] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang,

and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018.

[207] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1456–1465, 2019.

[208] Eric Tzeng, Kaylee Burns, Kate Saenko, and Trevor Darrell. Splat: semantic pixel-level adaptation transforms for detection. *arXiv preprint arXiv:1812.00929*, 2018.

[209] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[210] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.

[211] Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *European Conference on Computer Vision*, 2018.

[212] Kaushik P Venkatesh, Marium M Raza, Grace Nickel, Serena Wang, and Joseph C Kvedar. Deep learning models across the range of skin disease. *npj Digital Medicine*, 7(1):32, 2024.

[213] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NIPS*, 2016.

[214] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*. PMLR, 2013.

[215] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

[216] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[217] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[218] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *European Conference on Computer Vision*, 2018.

[219] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 2009.

[220] Marcelo J Weinberger and Erik Ordentlich. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*, 2002.

[221] Qingsong Wen, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, Tieniu Tan, et al. Onenet: Enhancing time series forecasting models under concept drift by online ensembling. *Advances in Neural Information Processing Systems*, 36, 2024.

[222] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[223] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models

improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[224] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019.

[225] Zijing Wu, Ce Zhang, Xiaowei Gu, Isla Duporge, Lacey F Hughey, Jared A Stabach, Andrew K Skidmore, J Grant C Hopcraft, Stephen J Lee, Peter M Atkinson, et al. Deep learning enables satellite-based monitoring of large populations of terrestrial mammals across heterogeneous landscape. *Nature communications*, 14(1):3072, 2023.

[226] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[227] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, 2021.

[228] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Self-training with noisy student improves imagenet classification. *arXiv preprint arXiv:1911.04252*, 2019.

[229] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[230] Jiaolong Xu, Sebastian Ramos, David Vázquez, and Antonio M López. Domain adaptation of deformable part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2367–2380, 2014.

[231] Jiaolong Xu, Sebastian Ramos, David Vázquez, Antonio M López, and D Ponsa. Incremental domain adaptation of deformable part-based models. In *BMVC*, 2014.

[232] Xinyi Xu, Yanhua Yang, Cheng Deng, and Feng Zheng. Deep asymmetric metric learning via rich relationship mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4076–4085, 2019.

[233] Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang, and Changjun Jiang. Random forest for credit card fraud detection. In *2018 IEEE 15th international conference on networking, sensing and control (ICNSC)*, pages 1–6. IEEE, 2018.

[234] Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei W Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. *Advances in Neural Information Processing Systems*, 35:10309–10324, 2022.

[235] Zhenpeng Yao, Yanwei Lum, Andrew Johnston, Luis Martin Mejia-Mendoza, Xin Zhou, Yonggang Wen, Alán Aspuru-Guzik, Edward H Sargent, and Zhi Wei Seh. Machine learning for a sustainable energy future. *Nature Reviews Materials*, 8(3):202–215, 2023.

[236] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

[237] Zhonghui You, Kun Yan, Jinmian Ye, Meng Ma, and Ping Wang. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. *arXiv:1909.08174*, 2019.

[238] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision*, 2018.

[239] Xiaofan Yu, Yunhui Guo, Sicun Gao, and Tajana Rosing. Scale: Online self-supervised lifelong learning without prior knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[240] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, 2019.

[241] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning (ICML)*, 2021.

[242] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1476–1485, 2019.

[243] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. The visual task adaptation benchmark. 2019.

[244] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2020–2030, 2017.

[245] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6810–6818, 2018.

[246] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[247] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin,

and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *European Conference on Computer Vision*, 2018.

[248] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[249] Yangtao Zheng, Di Huang, Songtao Liu, and Yunhong Wang. Cross-domain object detection through coarse-to-fine feature adaptation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13766–13775, 2020.

[250] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv:1904.07850*, 2019.

[251] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.

[252] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 687–696, 2019.

[253] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020.

[254] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.