Representation Learning for Few-Shot Image Classification

Thèse

Arman Afrasiyabi

Doctorat en génie électrique
Philosophiæ doctor (Ph. D.)

Québec, Canada

Representation Learning for Few-Shot Image Classification

Thèse

Arman Afrasiyabi

Sous la direction de:

Christian Gagné, directeur de recherche
Jean-François Lalonde, codirecteur de recherche

Résumé

En tant qu'algorithmes d'apprentissage automatique à la pointe de la technologie, les réseaux de neurones profonds nécessitent de nombreux exemples pour bien fonctionner sur une tâche d'apprentissage. La collecte et l'annotation de multiples échantillons nécessitent un travail humain important et c'est même impossible dans la plupart des problèmes du monde réel tel que l'analyse de données biomédicales. Dans le contexte de la vision par ordinateur, la classification d'images à quelques plans vise à saisir la capacité humaine à apprendre de nouveaux concepts avec peu de supervision. À cet égard, l'idée générale est de transférer les connaissances des catégories de base avec plus d'encadrement vers des classes nouvelles avec peu d'exemples. En particulier, les approches actuelles d'apprentissage à quelques coups pré entraînent un modèle sur les classes de base disponible pour généraliser aux nouvelles classes, peut-être avec un réglage fin. Cependant, la généralisation du modèle actuel est limitée en raison de certaines hypothèses lors de la préformation et de restrictions lors de l'étape de mise au point.

Cette thèse vise à assouplir trois hypothèses des modèles d'apprentissage à quelques plans actuels et nous proposons un apprentissage de **représentation pour la classification d'images à quelques plans**.

Tout d'abord, le gel d'un modèle préformé semble inévitable dans la phase de réglage fin en raison de la forte possibilité de surentraînement sur quelques exemples. Malheureusement, l'apprentissage par transfert avec une hypothèse de **modèle gelé** limite la capacité du modèle puisque le modèle n'est pas mis à jour avec aucune connaissance des nouvelles classes. Contrairement au gel d'un modèle, nous proposons un alignement associatif qui permet d'affiner et de mettre à jour le réseau sur de nouvelles catégories. Plus précisément, nous présentons deux stratégies qui détectent et alignent les nouvelles classes sur les catégories de base hautement liées. Alors que la première stratégie pousse la distribution des nouvelles classes au centre de leurs catégories de base associées, la seconde stratégie effectue une correspondance de distribution à l'aide d'un algorithme d'entraînement contradictoire. Dans l'ensemble, notre alignement associatif vise à éviter le surentraînement et à augmenter la capacité du modèle en affinant le modèle à l'aide de nouveaux exemples et d'échantillons de base associés.

Deuxièmement, les approches actuelles d'apprentissage à quelques coups effectuent le transfert de connaissances vers de nouvelles classes distinctes sous l'hypothèse **uni modale**, où tous les exemples d'une seule classe sont représentés par un seul cluster. Au lieu de cela, nous proposons une approche d'apprentissage de l'espace des caractéristiques basée sur le mélange (MixtFSL) pour déduire une

représentation multimodale. Alors qu'un précédent travail basé sur un modèle de mélange d'Allen et al. citeallen2019infinite est basé sur une méthode de clusters classique de manière non différentielle, notre MixtFSL est un nouveau modèle multimodale de bout en bout et entièrement différentielle. MixtFSL capture la multimodale des classes de base sans aucun algorithme de clusters classique à l'aide d'un cadre en deux étapes. La première phase s'appeler formation initiale et vise à apprendre la représentation préliminaire du mélange avec une paire de fonctions de perte. Ensuite, l'étape suivante progressive, la deuxième étape, stabilise la formation avec un cadre de formation de type enseignant-élève utilisant une fonction de perte unique.

Troisièmement, contrairement aux techniques actuelles à quelques prises de vue consistant à représenter chaque exemple d'entrée avec une seule entité à la fin du réseau, nous proposons un extracteur d'entités d'ensemble et des ensembles d'entités correspondantes qui assouplissent l'hypothèse typique basée sur **une seule entité** en raisonnant sur des ensembles d'entités. Ici, nous émettons l'hypothèse que l'hypothèse d'une seule caractéristique est problématique dans la classification d'images à quelques prises de vue puisque les nouvelles classes sont différentes des classes de base préformées. À cette fin, nous proposons nouvel extracteur de caractéristiques d'ensemble d'apprentissage profond basé sur les réseaux de neurones hybrides convolution-attention. De plus, nous suggérons trois métriques ensemble à ensemble non paramétriques pour séduire la classe de l'entrée donnée.

Cette thèse utilise plusieurs indicateurs standards publiés dans la littérature sur l'apprentissage en peu d'exemples et l'ossature de réseau pour évaluer les méthodes que nous proposons.

Abstract

As the current state-of-the-art machine learning algorithms, deep neural networks require many examples to perform well on a learning task. Gathering and annotating many samples requires significant human labor, and it is even impossible in most real-world problems such as biomedical data analysis. Under the computer vision context, few-shot image classification aims at grasping the human ability to learn new concepts with little supervision. In this respect, the general idea is to transfer knowledge from base categories with more supervision to novel classes with few examples. In particular, the current few-shot learning approaches pre-train a model on available base classes to generalize to the novel classes, perhaps with fine-tuning. However, the current model's generalization is limited because of some assumptions in the pre-training and restrictions in the fine-tuning stage.

This thesis aims to relax three assumptions of the current few-shot learning models, and we propose **representation learning for few-shot image classification.**

First, freezing a pre-trained model looks inevitable in the fine-tuning stage due to the high possibility of overfitting on a few examples. Unfortunately, transfer learning with a **frozen model** assumption limits the model capacity since the model is not updated with any knowledge of the novel classes. In contrast to freezing a model, we propose associative alignment that enables fine-tuning and updating the network on novel categories. Specifically, we present two strategies that detect and align the novel classes to the highly related base categories. While the first strategy pushes the distribution of the novel classes to the center of their related base categories, the second strategy performs distribution matching using an adversarial training algorithm. Overall, our associative alignment aims to prevent overfitting and increase the model capacity by refining the model using novel examples and related base samples.

Second, the current few-shot learning approaches perform transferring knowledge to distinctive novel classes under the **uni-modal** assumption, where all the examples of a single class are represented with a single cluster. Instead, we propose a mixture-based feature space learning (MixtFSL) approach to infer a multi-modal representation. While a previous mixture-model-based work of Allen et al. [1] is based on a classical clustering method in a non-differentiable manner, our MixtFSL is a new end-to-end multi-modal and fully differentiable model. MixtFSL captures the multi-modality of base classes without any classical clustering algorithm using a two-stage framework. The first phase is called initial training and aims to learn preliminary mixture representation with a pair of loss functions. Then, the progressive following stage, the second stage, stabilizes the training with a teacher-student kind of

training framework using a single loss function.

Third, unlike the current few-shot techniques of representing each input example with a single feature at the end of the network, we propose a set feature extractor and matching feature sets that relax the typical **single feature-based** assumption by reasoning on feature sets. Here, we hypothesize that the single feature assumption is problematic in few-shot image classification since the novel classes are different from pre-trained base classes. To this end, we propose a new deep learning set feature extractor based on the hybrid convolution-attention neural networks. Additionally, we offer three non-parametric set-to-set metrics to infer the class of the given input.

This thesis employs several standard benchmarks of few-shot learning literature and network backbones to evaluate our proposed methods.

Contents

List of Tables

List of Figures

To my family: Veedaa, Simin ...
for their unceasing support and love

Acknowledgments

I express tremendous gratitude to my advisers Jean-François Lalonde and Christian Gagné. Without your support, I wouldn't be writing these lines. Thank you so much, Christian, for guiding me throughout my Ph.D. and providing me the freedom to follow my research interest. At the same time, a big thank you, Jean-François, for helping me develop the whys and hows of this dissertation. Dear JF, I will never forget your professional and ethic-based research character. I was fortunate to have the opportunity to collaborate with you, JF and Christian. Besides, I am grateful to collaborate with Hugo Larochelle for the last project of this thesis. Hugo advanced the project, and I learned much valuable knowledge.

My Ph.D. study builds new personal friendships. With the fear of forgetting a name here, I am grateful to ALL of my lovely friends for the joyful time we had while working on this thesis. I am also thankful to all the lab students for their valuable comments on my work and for proofreading our papers, reading groups, seminars, and other academic activities.

What about the family? I honestly don't know how to thank my family for their love. They taught me to respect the knowledge, work hard and follow my supervisors. I promised them to share my academic experience with the young researchers upon request unconditionally. Regarding the family, I am the luckiest and richest person in the world. Dear Simin (Ana), Sana (Balam), and other sweethearts, I am getting back to see you in a week after 1657 days, since the starting data of my study and the pandemic Covid is ending :). Last but least, I am eternally grateful to the love of my life, Veedaa; you alone were my refuge and safe place during this journey. Veedaa, your passion, love, and professional personality motivated and inspired me to follow my dreams. From the bottom of my heart, THANK YOU!

Foreword

This document is an article-based Ph.D. thesis that follows three objectives. The first objective (chap. 2) has been accepted as a spotlight presentation (5% acceptance rate) and published in the European Conference on Computer Vision (ECCV) 2020. ECCV is one of the top-level international computer vision conferences.

> *Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In European Conference on Computer Vision (ECCV), 2020*

Arman Afrasiyabi was the main contributor to this work. Jean-François Lalonde and Christian Gagné were the director of the project. The paper was submitted on 5 March 2020 and was accepted on 3 July 2020.

The second objective of this thesis (chap. 3) has been accepted and published in the International Conference on Computer Vision (ICCV) 2021 (26% acceptance rate). ICCV is one of the top-level international computer vision conferences.

> *Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Mixture-based feature space learning for few-shot image classification. In International Conference on Computer Vision (ICCV), 2021*

Arman Afrasiyabi was the main contributor to this work. Jean-François Lalonde and Christian Gagné were the supervisors of this project. The paper was submitted on 17 March 2021 and was accepted on 22 July 2021.

The third objective (chap. 4) has been accepted to the International Conference on Computer Vision and Pattern Recognition (CVPR) 2022. CVPR is the top-tier conference in computer vision with 25% acceptance rate.

> *Arman Afrasiyabi, Hugo Larochelle, Jean-François Lalonde, and Christian Gagné. Matching feature sets for few-shot image classification. In International Conference on Computer Vision and Pattern Recognition (CVPR), 2022*

Arman Afrasiyabi was the main contributor to this work. Hugo Larochelle was the collaborator from Google Brain. Jean-François Lalonde and Christian Gagné were the directors of this project. The paper was submitted on 18 November 2021 and was accepted on 2 March 2022.

Introduction

Unlike modern artificial intelligence (AI) algorithms, humans can generally learn new concepts using little supervisions. A child can recognize the "mammoth" class after seeing a few examples at a museum, while other related animals (e.g., elephants) have learned before. As a branch of AI, machine learning methods such as deep learning aims to learn from data to imitate the human learning ability by automating analytical model building. Under the machine learning context, supervised learning uses annotated data to supervise the automated learning of patterns.

Recent machine learning approaches such as supervised learning still have difficulties generalizing new concepts with few annotated data. Specifically, supervised machine learning algorithms such as deep learning require much supervision with annotated data to recognize a category. For example, the current deep learning models can perform better than the human object recognition ability on ImageNet [2] benchmark, which contains more than 1.2 million annotated instances over 1000 classes.

Unfortunately, gathering and annotating a large amount of data to obtain a good learning model might not be possible for some real-world applications, such as biomedical data analysis. Therefore, researchers are highly interested in the *learning-to-learn* paradigm, where the idea is to use available annotated data to learn a model such that it can generalize on novel tasks with few examples. To do so, few-shot learning focuses on making predictions on the novel concepts containing few labeled data by learning prior experiences with many annotated data.

The real-world application of a well generalized few-shot model could extend from natural language processing to computer vision. This thesis focuses on a supervised few-shot image classification under computer vision to recognize visual concepts with a few images. To do so, we cast the problem of few-shot learning in the form of using pre-trained experiences. In particular, we aim at building a robust learning model using a large collection of labeled data available over the so-called base classes to adapt and learn new classes with few examples, even one instance in the extreme case.

Nowadays, the state-of-the-art approach for few-shot image classification is transfer learning. We pre-train a neural network model on the available base classes to generalize on the novel classes in the second learning stage called fine-tuning stage. In this respect, we could have either one or both of the following pre-training frameworks: 1) standard transfer learning [3, 4, 5, 6], and 2) meta-learning [7, 8, 9, 10].

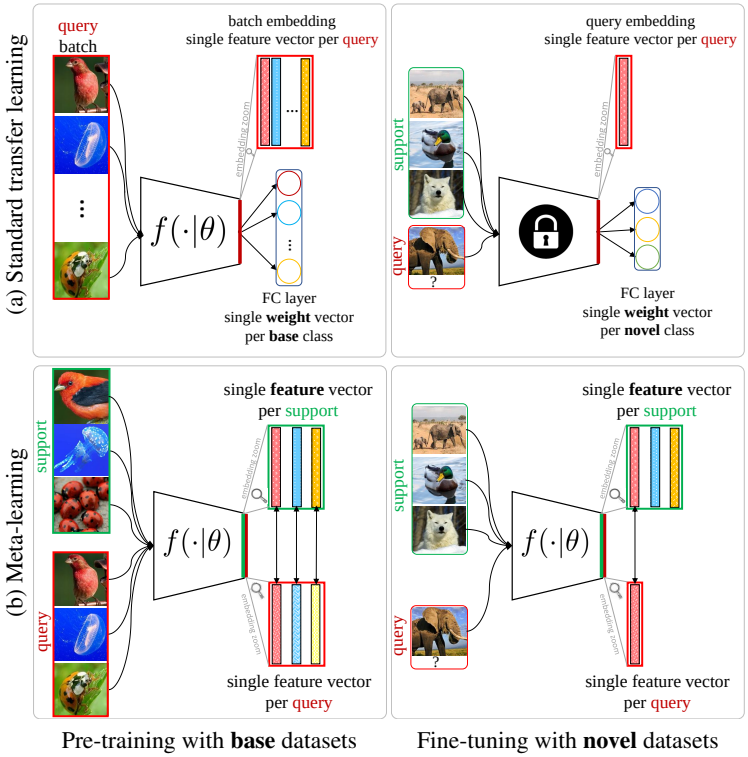


Figure 0.1: Standard transfer learning (top row) vs. meta-learning (bottom row) while learning on the *base* or *novel* datasets presented in the first and second columns, respectively. (a) In standard transfer learning (top row), we first pre-train a learning neural network model $f(\cdot|\theta)$, parameterized by θ, using single weight vectors per class in the fully connected (FC) layer. Then, we replace the base class weights (base FC) with the single novel class weight vectors (novel FC) and refine a new FC layer during the learning with novel classes. Finally, we predict the label of the query using the frozen feature extractor and the weight vectors. (b) In meta-learning (bottom row), we sample support examples that are different examples from the same classes of the query. We then use support set features at the end of the network instead of trainable weight vectors of standard transfer learning in both learning with base and novel classes. Note that we should freeze the network during the novel class generalization to prevent overfitting.

Standard transfer learning consists of two stages: 1) pre-training and 2) fine-tuning. As presented in fig. 0.1 (top row; left column), we pre-train a few-shot model using base classes with a specified base class classification layer, where there is *single weight vector* per each of the base classes. Intuitively, the learning process performs representation learning by maximizing the similarity between feature embedding of an input query example with its corresponding class *weight* vector. Consequently, the resulting feature space at the end of the network tends to group the examples of the same class into a single cluster. After the pre-training, we remove the classification layer during the novel class generalization, as shown in fig. 0.1 (first row; second column). Next, we fix the feature extractor and only fine-tune a new classification layer which is specified for the novel categories. Finally, we map the unseen query example into the feature space with the network, and the class of the query example is inferred by measuring its similarity with the novel class weights. Notably, we have a *single* weight

vector per class in both learning with base and novel classes.

As an alternative training framework and illustrated in fig. 0.1 bottom row, meta-learning (also known as episodic training) repeatedly samples episodes of N classes from all of the M base classes, where $N \leq M$ and each episode contains *support* set and *query* set of examples per each class. In fact, during the base class learning, the idea of meta-learning is to effectively imitate learning from novel classes with few examples. In this aspect, the support set is annotated data used to infer the classes of the query set. Here, we employ support examples in meta-learning instead of defining the set of weights employed in standard transfer learning. In other words, the network is updated with stochastic gradient descent (SGD) algorithm which backpropagates the error based on the discrepancy between support set and query set. Then, the model generalizes on the novel classes with a fixed feature extractor and uses a classification algorithm such as centroid nearest neighbor [9]. In other words, the meta-learning approach leaves out the usage of the classification weights in learning with both base and novel categories.

Most recently, a hybrid framework of standard transfer learning and meta-learning resulted in better performance [11, 12, 13] in the few-shot image classification compared to employing only one approach. In the hybrid pre-training framework, we train a model using standard transfer learning (top row of fig. 0.1) using base categories. We again pre-train the model with meta-learning (bottom row of fig. 0.1) on base classes to better generalize the novel classes.

Apart from training strategy, the state-of-the-art few-shot image classification approaches rely on three assumptions. First, the network is considered to be fixed or frozen during learning with novel class examples as shown in fig. 0.1 (second column). Though the fixed network assumption looks inevitable since any model adaptation would result in overfitting on few examples, fixing procedure limits the model capacity, while learning with novel class concepts. As the second assumption presented in fig. 0.1, we assume that only single weight vector of standard transfer learning or single features of support examples can be used to infer the class label of the query examples. Indeed, the discussed assumption is unimodal, limiting the model's adaptability. As the final assumption, we assume that we can represent each input query example with a single feature in the feature space. Here, the single feature assumption might be strong for few-shot learning tasks where the base and novel classes are distinct. Notably, the fixed model and unimodal assumptions are in learning with novel and base classes, respectively, and the single feature assumption stays in both learning and novel classes.

This dissertation proposes specified ***representation learning for few-shot image classification*** to relax the discussed three assumptions. With this, we investigate our dissertation's proposal with the following three objectives:

1. **improving the model capacity to perform effective fine-tuning of novel samples by aligning the novel classes to the examples from base data that are most similar**; this is because almost all of the few-shot methods [9, 10, 3] assume to freeze the feature extractor while classifying

the novel categories. Indeed, freezing a model might look inevitable since the network would potentially overfit the novel few examples. However, freezing a network reduces the model capacity, since we doe not update the model;

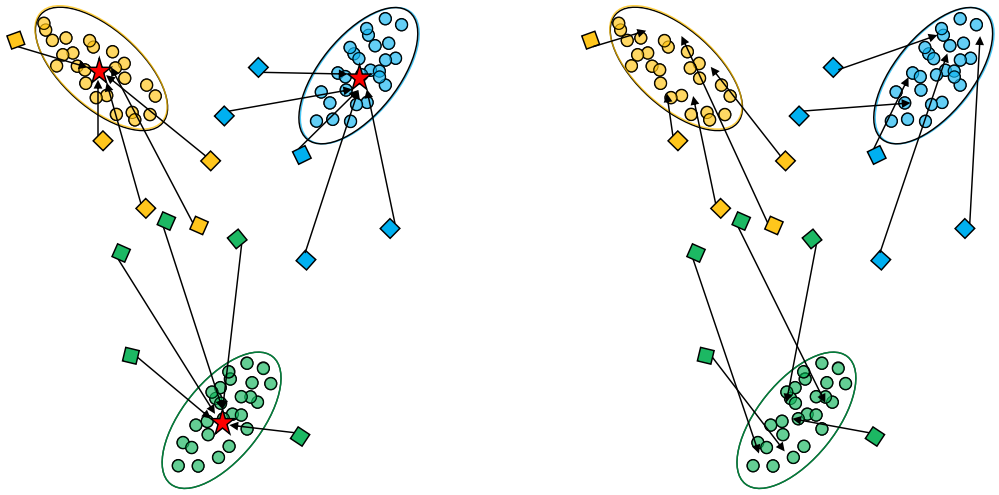
2. **boosting the model adaptability through a fully differentiable and an end-to-end mixture based feature space learning which captures the multimodalities of the base classes**; this is because, almost all of the current few-shot approaches rely on a unimodal assumption where each base class is represented in a single mode. Interestingly, Allen *et al.* [1] illustrated the advantage of multimodal representation learning. However, the proposed model [1] suffer from some limitations, such as using a post hoc non-differentiable classical clustering algorithm in an offline manner. In contrast to Allen *et al.* [1], we propose an advanced end-to-end fully differential model that does not require any classical clustering algorithm.

3. **enhancing the model representation power with feature sets extractor and set-based reasoning instead of relying on generic single feature extractor**, where multiple feature are extracted from a given image for effective transferring knowledge. Depending on a single feature assumption might not be a severe problem in a generic image classification problem where the new query image is from one of the base classes. However, relying on a single feature would be a strong assumption in the case of few-shot image classification since the base classes are distinct from the novel classes. In other words, a single feature-based model can be specialized (or overfit) on the base classes and harden the transfer learning to the novel classes. As the third contribution, we propose to build a rich feature space that is both more informative and easier to transfer to the novel domain.

To achieve these objectives, we propose three approaches: 1) *associative alignment* to detect and align base examples to their related novel categories to update the model during the novel class generalizations, 2) *mixture-based feature space learning* to capture the multimodalities of each base class with a two-stage algorithm to boost the model adaptability, and 3) *matching feature sets* with our proposed set feature extractor three set-to-set metrics to improve representation learning during both learning with base and novel classes.

Objective 1: enhancing the model capacity with associative alignment In few-shot image classification, we have two options after pre-training the learner on the base classes: 1) freezing the network, which limits the model capacity but prevents the overfitting problem, 2) fine-tuning the model with few novel samples, which increases the model capacity but results in an over-fitted model. Except for the model agnostic meta-learning (MAML) approach [7] with few gradient steps, all the other standard transfer learning approaches [3] and all the other meta-learning approaches [10] freeze the network to omit the overfitting on novel categories, as shown in the fig. 0.1 the second column.

We hypothesize that freezing a pre-trained neural network model while generalizing to few-shot novel class regimes can hinder the classification accuracy. Therefore, as the first objective, we propose an



(a) centroid alignment (b) adversarial alignment

Figure 0.2: Illustration of our associative alignment strategies in 3-way (color coded) classification problem, where diamonds and circles represent the few novel classes and their large related base, respectively. (a) Centroid alignment aims at pushing the novel examples to the center of their associated base class distribution. (b) Adversarial alignment performs a naïve distribution matching between novel classes and their related bases without pushing them to a specific region of the distribution.

associative alignment approach to balance the discussed model capacity and overfitting trade-off for the few-shot image classification. Accordingly, we introduce associative alignment for few-shot image classification that: *categories to the distribution of pre-trained related base categories, where the associated bases are recognized based on their similarity to the novel classes*.

In particular, we explore the associative alignment under two presented alignment strategies. First, we propose centroid alignment, inspired by ProtoNet [9], that reduces the intra-class variations of the associative alignment procedure. Here, the assumption is that the aligned class distributions can be approximated unimodal. Second, we present adversarial alignment, inspired by WGAN [14], with greater training complexity due to the critic network.

We contribute to this objective in the following forms:

1. we propose the idea of associative alignment to align the novel classes to their related base classes in the feature space. In particular, after pre-training the model using base classes and given novel classes with few-shot, we propose a straightforward algorithm to detect the most related class to a specific novel class. With this, we present two alignment strategies: centroid and adversarial, presented in fig. 0.2. While centroid alignment pushes novel examples to the centroid of their related base classes, the adversarial alignment performs distribution matching between the novel and related base classes motivated by using the recent adversarial methods.

2. we adopt an additive angular margin loss (originally proposed by Deng et al. [15] for face

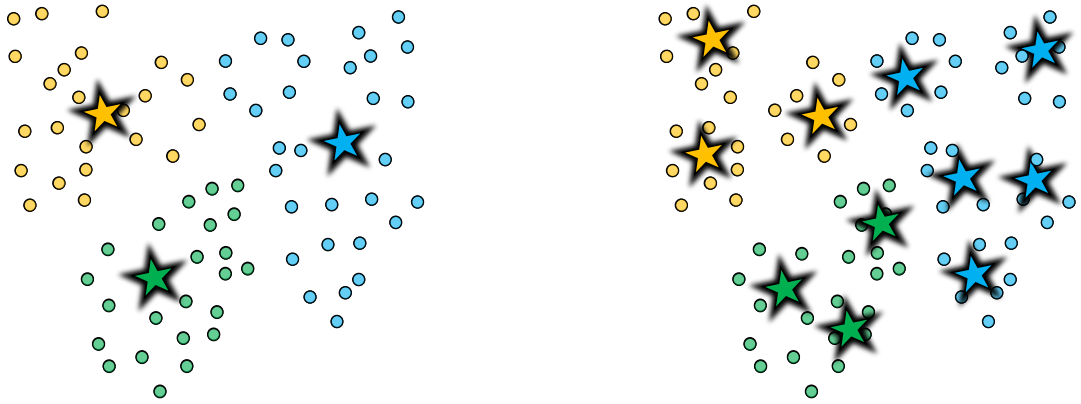
recognition) along with early stopping to regularize the network to propose a new baseline for few-shot image classification;

3. we investigate the effect of associative alignment through extensive experiments—with four standard benchmarks and three backbones. Specifically, we evaluate our associative alignment under generic object recognition with mini-ImageNet [10], tieredImageNet [16] and FC100 [17], fine-grained classification with CUB [18], and cross-domain adaptation from mini-ImageNet to CUB scenarios using three backbones used in literature: Conv4, WideResNet, and ResNet-18. For example, our centroid alignment achieves absolute accuracy improvements of 4.4%, 1.2%, and 6.2% in 5-shot learning over state-of-the-art for object recognition, fine-grained classification, and cross-domain adaptation, respectively.

Objective 2: improving adaptability with mixture-based feature space learning Another general assumption of few-shot learning approaches is unimodal, where each class is represented with a single point in the feature space shown in fig. 0.3 (a). For instance, ProtoNet [9], as a meta-learning approach, reduces the intra-class variations between query sets and the centroids of the corresponding support sets by employing the meta-learning framework. Besides, as a standard transfer learning approach, Baseline++ [3] uses a single FC-layer with a single learnable component per class in the form of standard transfer learning. After presenting the adaptability limitation of the unimodal assumption, Allen *et al.* [1] propose a multimodal method in an offline manner using a classical clustering DP-means [19] algorithm. Indeed, the offline procedure of [19] causes a severe limitation of the approach—clustering is restricted to be based on episodes containing few examples that would potentially result in misestimation of the clusters.

In this dissertation, instead of relying on the unimodal assumption fig. 0.3 (a) or using an offline multimodal technique [1] shown in fig. 0.3 (b), we propose a mixture-based approach that captures the multimodal representation of the base classes in an online manner without any classical clustering algorithm. We aim to improve the model adaptability by learning the mixture model within classes in an unsupervised manner while discovering the between class representations in a supervised fashion. In particular, we present "mixture-based feature space learning" that: *learns the multimodal representation of the base classes with a mixture model—trainable vectors that are iteratively refined by the stochastic gradient descent procedure* . Here, the goal is to train the feature extractor to capture a multimodal discriminative representation of a class without collapsing it on a single mode. To this end, we propose a two-stage algorithm for training the feature extractor and learnable components. In the first stage, the initial training contains two losses to ensure the initial model would not collapse to unimodal representation. However, the combination of two losses in the initial training stage results in instability in the training procedure. In the second training stage, the progressive following stage, the model employs a single loss function and a leader-follower network training mechanism, stabilizing training and improving performance.

Our contribution to the discussed objective is divided into the follow parts:



(a) unimodal assumption (b) multimodal assumption

Figure 0.3: Unimodal vs. multimodal in the three-class classification problem. Circles present instances and stars represent the cluster centroids used for inference of a given query. The figure presents a three-class classification problem, where classes are color-coded. While under unimodal-based representation learning, we aim at representing the whole infers the class label of the query with a single point at the feature space, the multimodal-based assumption method aims to capture the mixture representation of a class with multiple points in the feature space.

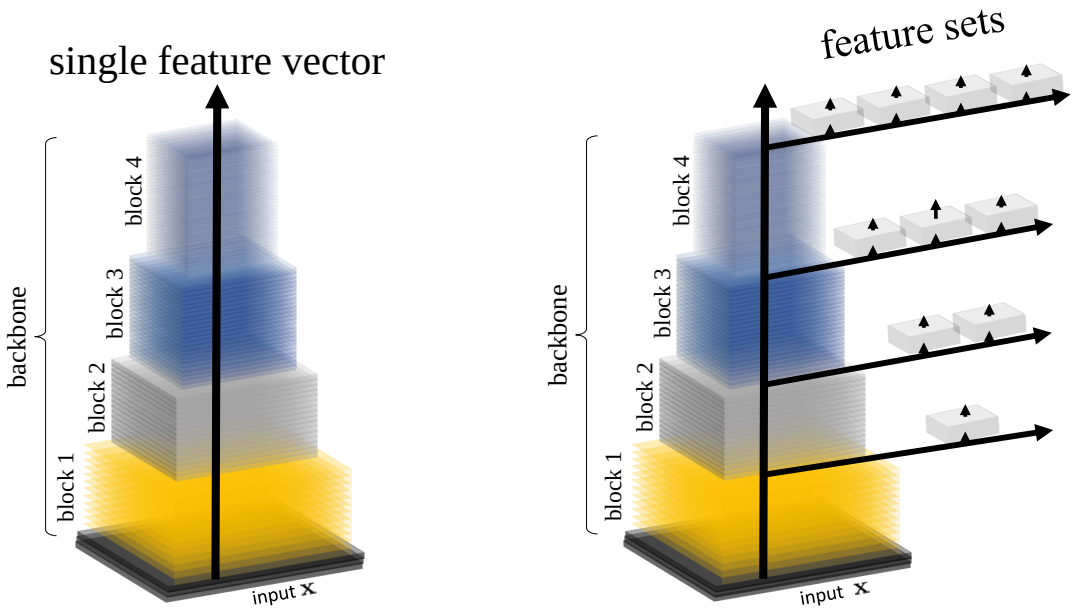
1. we introduce the idea of "Mixture-based Feature Space Learning", or MixtFSL, to learn an adaptable representation by capturing the multimodal representation of the base classes. As presented in fig. 0.3 (b), our MixtFSL aims at learning both between and within-class distributions;

2. we present a robust end-to-end and fully differentiable two-stage algorithm to train MixtFSL. While the first stage aims at learning mixture model representations of the classes with two loss functions, the second stage of MixtFSL completes the training procedure with single criteria and a kind of target-follower training paradigm;

3. we evaluate our model using four standard datasets and four backbones. Given mini-ImageNet [10], tieredImageNet [16], FC100 [17], and CUB [18] as the few-shot image classification benchmarks, we evaluate our MixtFSL using Conv4, ResNet-12, and ResNet-18 backbones;

4. we unify our MixtFSL and the associative alignment (our first objective). We named it MixtFSL-alignment, where we employ MixtFSL in base class learning and associative alignment during the novel class adaptations. Our evaluation showed the superiority of our MixtFSL-alignment.

Objective 3: informative representation learning with matching feature sets Current few-shot image classification models [7, 9, 10] generally rely on feature extractors that extract a single vector for a given input image as presented in fig. 0.4 (a). However, transferring knowledge using a single feature is optimistic for a few-shot classification since the base classes are distinct from novel categories. Instead of a single feature-based model, we propose to represent images as *sets* of the feature shown in fig. 0.4 (b). We present new representation learning methods that extract more informative feature sets from given an input image.



(a) common practice

(b) ours

Figure 0.4: While common practice is to extract a single feature (a) per input image at the end of the feature extract or backbone, our multi-feature-based model (b) aims at extracting multiple features from different parts of the neural architecture given input image $\mathbf{x}$. Here, the multiple features are extracted using our proposed self-attention-based modules shown by rectangles in (b) attached to different blocks of the backbone.

In this thesis, we propose new hybrid attention-convolution neural networks to build a set feature (SetFeat) extractor illustrated in fig. 0.4 (b), which extracts *sets* of feature vectors from images using approximately the same number of parameters as the compared state-of-the-art backbones. In particular, we propose to adapt attention-based functions called *mappers* to pull out features from different layers of ConvNets. In this respect, each mapper extracts a single feature using a self-attention mechanism inspired by [20, 21].

Note that current distance metrics are all built under a single feature set assumption. Therefore, we would need a new set-based metric to measure the feature sets discrepancy between a query and support set. Accordingly, our goal is to propose a novel set-based metric for deep representation learning, and we argue that: *set-based reasoning and inference intrinsically provide an informative representation of images from the base categories, which can subsequently perform better transfer learning*.

More specifically, we contribute to the discussed objective as follows:

1. we propose the idea of reasoning on *sets* using our SetFeat, which is a set feature extractor built by combining attention-based functions and convolution neural networks; this is because we aim at effective transfer learning with feature sets where each element focuses on different characteristics of an input image.

2. to evaluate differences between query and support set, we present set-based inference using one of our three different set-to-set metrics: match-sum, min-min, and sum-min;

3. we evaluated our matching feature set approach with extensive experiments on three popular few-shot datasets. Specifically, we evaluate our SetFeat and three metrics using mini-ImageNet [10], tieredImageNet [16], and CUB [18], and present comparisons three conventional backbones: Conv4, ResNet-12, and ResNet-18. Evaluation of our three metrics after extracting feature sets with SetFeat shows that the sum-min metric reaches new state-of-the-art results.

Chapter 1

Related Work

Deep neural networks have gained immense popularity over the last decade in many learning tasks, from natural language processing to computer vision. The state-of-the-art few-shot learning methods use deep neural networks to transfer knowledge from base categories with many examples to the novel classes containing few instances (or few-shot). To this end, we first pre-train a model on the base classes, where the proposed few-shot models use one or both of the two training algorithms: standard transfer learning and meta-learning. Then, the pre-trained model acts as an encoder or feature extractor to map the input image into a feature space. Here, the idea is to train the feature extractor to provide an informative representation of an input image to generalize a classifier of the novel classes.

Standard transfer learning [3, 4, 5, 6, 22] employs batch-based training using the feature extractor and the fully-connected (FC) classification layer, where we have a single output element per class at the FC [3]. In particular, we randomly sample a batch from all base classes and compute the loss based on the discrepancy between the batch and the classification layer. Chen *et al.* [3] showed the superiority of standard transfer learning with a metric-based cross-entropy loss for few-shot image classification in this aspect. As one of the first works to show the advantage of standard transfer learning, our associate alignment (chap. 2) and MixtFSL (chap. 3) aims at novel representation learning.

Meta-learning framework [7, 8, 9, 10], also called episodic training, trains the feature extractor with the sampled episodes from base categories without using FC. The idea of meta-learning is to simulate the novel-class stage in the pre-training phase. In particular, we train a feature extractor with the sampled subset of base classes. Vinyals *et al.* [10] proposed randomly sampling N-class if we have M base classes, where $N \leq M$. Then, for each sampled class, they randomly sample two sets of *support* and *query* examples. The idea is to compute the loss based on the difference between the query and support sets. Finally, the network is updated iteratively by backpropagating the calculated loss over different episodes.

Several studies [11, 12, 13] have recently illustrated that unifying standard transfer learning and meta-learning frameworks improves classification accuracy. Specifically, the idea is to pre-train the model based on standard transfer learning, and then the model is fine-tuned with meta-learning again using

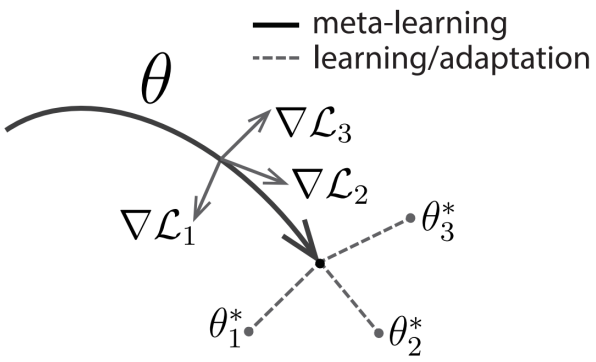


Figure 1.1: Initialization-based model-agnostic meta-learning (MAML) [7] model, which optimizes the parameters of the network θ (without adding extra parameters) for quick adaptation with two step gradient. Figure from Finn *et al.* [7].

base categories with the dropped FC-layer. The feature extractor could either be fixed or fine-tuned using novel classes. However, our marching feature set (chap. 4) benefits from both and combines standard transfer learning and meta-learning.

The remainder of this chapter elaborates on the related state-of-the-art methods. First, we present the idea of initialization-based approaches for few-shot learning. Second, the metric-based techniques are covered to build an informative feature space. Third, we would cover the data-augmentation methods to generate or hallucinate new data for few-shot image classification for better generalization. Finally, the highly related representation learning algorithms are covered.

1.1 Initialization-based techniques

Generic stochastic gradient descent (SGD) methods tend to slowly refine a neural network to tackle the overfitting problem in few-shot learning problem. Respectively, initialization-based approaches [7, 23, 24, 25] have been proposed to pre-train model parameters such that the model can adapt quickly to the novel categories with few examples. Fin *et al.* [7] offered model-agnostic meta-learning (MAML) to achieve rapid adaptation via the meta-learning framework. MAML aims at pre-training the model's parameters to have the best possible classification accuracy on a new novel task by proposing a two-step gradient procedure.

MAML does not add any extra learning parameters to the model architecture. Instead, it proposes to train the model parameters to generalize well on the novel task with few gradient steps or even a single gradient step in the case of 1-shot learning. As figure 1.1 illustrates, meta-gradient refines the parameters of the network with a gradient through a gradient. The network parameters θ become the gradient θ_i' of i-th task $\mathcal{T}_i$ of the episode as:

$$\theta_i' = \theta - \alpha\nabla_\theta\mathcal{L}_{\mathcal{T}_i}(f_\theta), \tag{1.1}$$

where α is the step size, and $\mathcal{L}_{\mathcal{T}_i}$ is the feedback generated by $\mathcal{T}_i$. Next, the meta-objective θ is

calculated with respect to θ_i' across the samples tasks in the episodes. Finally, the network parameters θ is updated with meta-optimization in form of stochastic gradient descent (SGD)

$$\theta = \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}), \tag{1.2}$$

where β is the meta step size.

However, the proposed MAML algorithm could not be well-generalized in high dimensional feature space [25]. Therefore, Rusu *et al. [25]* extends MAML to latent embedding optimization (LEO) to separate the underlying model parameters from gradients of task-based adaptation. In particular, LEO is built on MAML [7] which has difficulty in the case of high-dimensional parameter space given a few-shot learning task. To resolve a few samples in high-dimension, LEO changes the parameter space of MAML to a latent lower-dimensional space. The reported experimental evaluation with LEO [25] showed the effectiveness of handling dimensionality in few-shot regimes.

With the aim of increasing adaptability, our mixture-based feature space learning (MixtFSL) (presented in chap. 3) is related to initialization-based approaches. Notably, our MixtFSL follows different research directions and proposes a mixture-model representation instead of a two-level gradient. However, in contrast to the initialization-based approaches, the other proposed methods of this thesis focus on increasing the model capacity (chap. 2) and feature set representations (chap. 4).

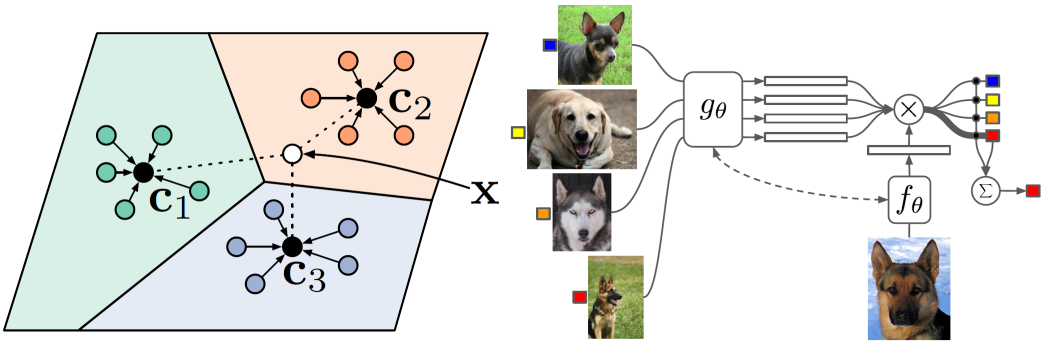
1.2 Metric-based techniques

The metric-based family of approaches [9, 10, 13, 17, 26, 27, 28, 29, 30, 31, 32, 33, 34] aim to propose new criteria to build a distance-based feature space. Generally, the metric-based methods can be grouped into three cases: without an extra learning module [9, 13], with an embedding function [10, 30], and with an adaptation function [11]. In the following sections, we discuss three examples of these approaches: ProtoNet [9], matching networks [10], and embedding adaptations [11].

1.2.1 Prototypical neural networks

In learning from severely limited data, we should address the overfitting since a neural network model with millions of parameters always has a high tendency to overfit on a few examples. By addressing the overfitting problem, Snell *et al*. [9] hypothesis to build a feature space with a simple classifier such as centroid nearest neighbors work. To do so, prototypical neural networks [9] (ProtoNet) cluster the samples of a class around a single prototype representation of a category.

As a well-known few-shot learning model and presented in figure 1.2 (a), ProtoNet proposes to reduce intra-class variations using the meta-learning training strategy. Therefore, ProtoNet removes the fully-connected layer requirement while learning a metric feature space. In particular, given a new query example $\mathbf{x}$, ProtoNet computes the probability distribution p_θ on the episodic classes using the



(a) ProtoNet [9] (b) Matching networks [10]

Figure 1.2: Metric based few-shot image classification method. ProtoNet [9] (a) aims at learning centroid based metric space by minimizing the query sets to the centroids of the support sets. Matching networks [10] (b) builds and attention based metric space feature space by stacking and embedding function over the convolutional feature extractor. Figures (a) and (b) are from [9], [10], respectively.

following softmax function:

$$p_\theta(y = n|\mathbf{x}) = \frac{\exp(-d(f_\theta(\mathbf{x}), \mathbf{c}_n))}{\sum_{n'} \exp(-d(f_\theta(\mathbf{x}), \mathbf{c}_{n'}))}, \tag{1.3}$$

where $d(\cdot)$ is a distance function, $f_\theta(\cdot)$ is the feature extractor, and $\mathbf{c}_n$ is the centroid of the n-the class and defined as: $\mathbf{c}_n = \frac{1}{S_n} \sum_{(\mathbf{x}_i, y_i) \in S_n} f_\theta(\mathbf{x}_i)$. To this end, the minimization of the negative log-probability $J(\theta) = -\log p_\theta(y = n|\mathbf{x})$ proceeds the training of $f_\theta(\cdot)$.

In this thesis, we inspired by ProtoNet [9] to propose the centroid alignment strategy (chap. 2) which increases the model capacity in the fine-tuning stage. Besides, our matching feature sets (chap. 4) method is related to ProtoNet from centroid-based feature space learning. However, unlike ProtoNet, which relies on a single feature representation learning, our matching feature set method is based on the extracted feature sets and our defined set-based metrics.

1.2.2 Matching networks

In contrast to inferring the label of a query example out of a single prototype feature per class, such as ProtoNet [9], matching networks [10] aim at inferring the label of a query example by examining all of the examples from different classes in attention form. As presented in figure 1.2 (b), Vinyals *et al.* [10] proposed to embed the input labeled support examples using g_θ model and the unlabeled query example with f_θ. Then the idea is to adapt an extra embedding function to learn the matching of the unlabeled query with the support set features in the feature space. In other words, matching networks [10] employ an extra embedding function beside the feature extractor to infer the query label.

Vinyals *et al*. [10] proposed a bidirectional Long-Short Term Memory (LSTM) [35] as an attention-based embedding function attLSTM$(\cdot)$ to match the query examples $\mathbf{x}_q$ to the support set $\mathcal{X}_s$:

$$h(\mathbf{x}_q, \mathcal{X}_s) = \text{attLSTM}(g(\mathbf{x}), f(\mathcal{X}_s), T), \tag{1.4}$$

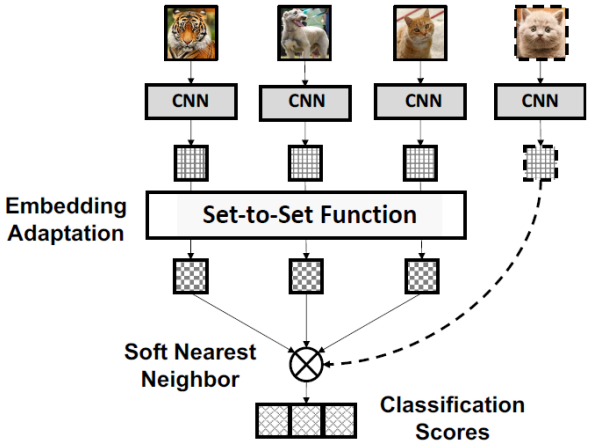


Figure 1.3: Metric based embedding adaptation [11] proposes a set-to-set adaptation function over the support set to match with the embedding of the query examples. The set-to-set function illustrated in the figure can be deep sets [36], LSTM [35], and transformers [20]. Figure is from [11].

where $g(\cdot)$ and $f(\cdot)$ are feature extractors for the query and support set, respectively, and support set $\mathcal{X}_s = \{\{(\mathbf{x}_k, y_c)\}_{k=1}^{K}\}_{c=1}^{C}$ containing K labeled examples from C classes, and T is the number of unrolling steps. In practice, Vinyals *et al.* [10] set both $g(\cdot)$ and $f(\cdot)$ to same ConvNet. Notably, the idea of the matching network is to build an attention-based feature space, where the input query is represented in the attentional form of the support set.

Evaluation of matching networks results in superiority over the baseline method, specifically in the 1-shot scenario. However, we should note that the matching network benefits from an additional model capacity of LSTM module that contains other parameters to the overall model.

Our *matching feature sets* approach, presented in chapter 4 of this dissertation, is related to matching networks by attention-based feature extraction. However, our method is set-based representation learning, unlike a single feature-based matching network. Besides, our feature sets method contains light-weighted self-attention functions called *mappers* instead of LSTM.

1.2.3 Embedding adaptation

The discussed few-shot learning approaches are task-agnostic, where we use the same embedding function on both labeled support and unlabeled query examples under the meta-learning paradigm. Unlike these task-agnostic methods, Ye *et al.* [11] proposed an embedding function specified to learn information concerning the query examples. As fig. 1.3 presents, the embedding adaptation method of [11] that applies a set-to-set function only on support examples. The learned embedding function leverages the relationship between support and query samples, leading to informative and discriminate representation learning. This is different from matching network [10] presented in fig. 1.2 (b) that applies an embedding function on both support and query examples.

Specifically, embedding adaptation [11] performs partial adaptation in an embedded episode. After embedding both support and query instances using ConvNet feature extractor, Ye *et al.* [11] investigate different adaptation functions such as deep sets [36], LSTM [35], and transformers [20] as the set-to-set matching to compare the query example to the support set. The evaluations on different datasets and setups illustrate the advantage of employing the transformer [20]. As presented in figure 1.3 the set-to-set adaptation function is only applied on the support set to adapt for the query representation. Here, the term *set-to-set* refers to support set to query set adaptation. Evaluation with different embedding functions resulted in the superiority of Transformers [20] over other compared methods such as Deep Sets [36] and bidirectional LSTM employed in matching network.

Our matching feature set (chap. 4) is closely related to the set-to-set method proposed by Ye *et al.* [11]. Nevertheless, our work generalizes the term *set* to the extraction of feature sets for each example in the support set where each query is treated independently.

1.3 Data augmentation techniques

Data augmentation approaches [16, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50] aim to expand the size of the support set to increase the generalization of the few-shot models. Generally, the augmented data can be generated, hallucinated, or selected from a large pool of unlabeled data.

1.3.1 Generating auxiliary data

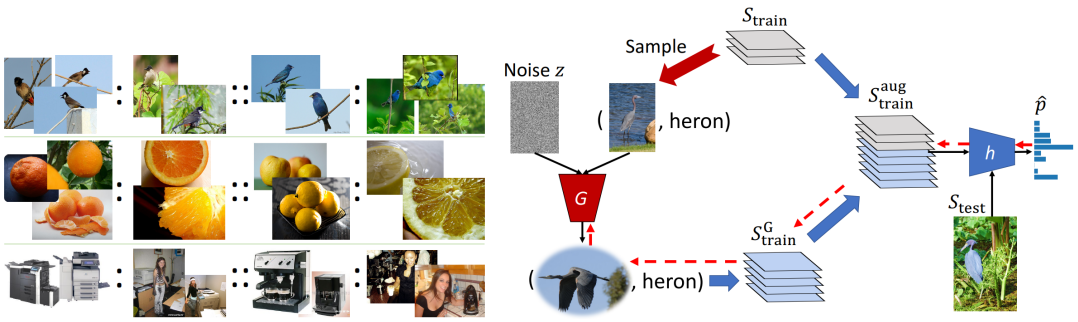
Inspired by transferring certain modes of within class information to between classes representation learning, Hariharan *et al.* [38] propose feature hallucination (FH) using a hallucination function G. In particular, FH [38] propose to "hallucinating" extra samples for the novel classes by transferring modes of variation from the base classes.

To perform a robust representation learning, FH first clusters the base classes. Then, for each pairs of cluster centroids (c_1^a, c_2^a) in one category, FH searches for the pairs of centroids (c_1^b, c_2^b) in another category such that the cosine similarity between ($c_1^a - c_2^a$) and ($c_1^b - c_2^b$) would be minimum. The idea is to train G using the detected quadruplets ($c_1^a, c_2^a, c_1^b, c_2^b$) data as presented in figure 1.4 (a), where each row presents a quadruplet. Particularly, FH uses all quadruplets to train the generator G, such that G estimates $\hat{c}_2^a = G([c_1^a, c_1^b, c_2^b])$, and FH minimises the following loss function:

$$\lambda \mathcal{L}_{mse}(\hat{c}_2^a, c_2^a) + \mathcal{L}_{cls}(\theta, \hat{c}_2^a), \tag{1.5}$$

where, λ is coefficient, $\mathcal{L}_{mse}(\cdot, \cdot)$ is the mean squared error, $\mathcal{L}_{cls}(\theta, \cdot)$ is the classification loss used to train the parameters of the neural network θ. With this learning algorithm, FH improve the 1-shot accuracy on novel classes by 15 points.

Alas, training the discussed FH [38] is hard, except for trivial tasks. The other alternative would be taking the inspiration from generative adversarial models to generate novel extra realistic samples in



(a) Hallucinated data [38] (b) Imaginary data [46]

Figure 1.4: Motivation and ideas of three state-of-the-art generating auxiliary data techniques. (a) feature hallucination (FH) [38] presented in three-row, where each row shows the detected quadruplets clusters used to train the hallucination function. (b) imaginary data [46] generation motivated by the generative adversarial network (GAN) G to build a robust model. Figures (a) and (b) are from [38] and [46], respectively.

the input image space. However, most current generative models would fail since they would suffer from mode collapse. In other words, applying a generic generative model would transfer only specific modes. Motivated by this, as another data generator approach, Wang *et al.* [46] proposed to generate imaginary data in the input image space using generative adversarial networks (GANs). Here, the key idea is to hallucinate additional informative examples for learning classifiers instead of generating realistic images.

The idea of FH inspired Wang *et al.* [46] to unify the idea of meta-learning and hallucination to generate examples with generative adversarial models. Indeed, the focus of Wang *et al.* [46] was on generating discriminative features instead of focusing on the realism of the generated images. As figure 1.4 (b) presents, the few-shot learning model gets both examples of episodes and develops examples to improve the generalization performers of the final model. The classification error backpropagates through the generator to refine it for generating informative instances. The performed experimental evaluation results in accuracy gain of the discussed generative model [46] over FH [38].

Similar to FH proposed by Hariharan *et al.* [38] and the generative model proposed by Wang [46], our associative alignment (chap. 2) and matching feature sets (chap. 4) also effectively increase the size of the feature in the embedding space, but in the different manners. However, Unlike generating auxiliary data, our associative alignment increases the feature set size by detecting the related base classes and using them to boost accuracy. Besides, matching feature sets also follows different to increase the size of the feature set by extracting multiple features from each input image.

1.3.2 Employing unlabeled data

As another approach to using the auxiliary data, several works [16, 42, 44, 47, 51] have been proposed to select extra data from a large pool of unlabeled data instead of generating new data with a generator. Similar to the generative approaches [38, 46], the idea of using available unlabeled data (besides

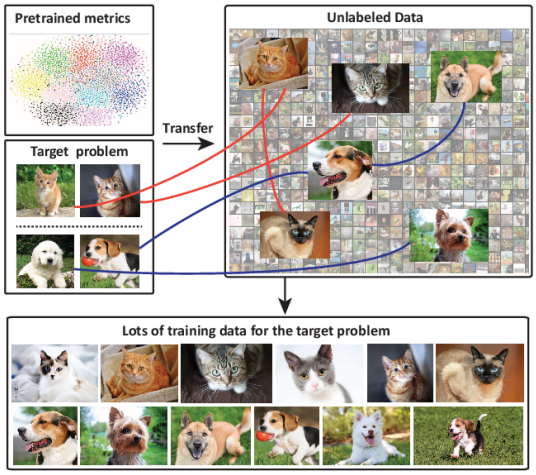


Figure 1.5: In label propagation method, Liu *et al*. [44] proposes to propagate the label of few examples of the novel classes to the pool of auxiliary unlabeled data using a pre-trained model in the form of supervised or self-supervised learning. Figure from [44].

the base classes) is to tackle the overfitting problem and build a well-generalized model. From the representation learning perspective, the discussed methods aim to transfer the learning from the annotated data to pick informative data from a pool of unlabeled data. Then, the selected data are used for rich representation learning through increasing adaptability.

As a model which uses unlabeled auxiliary data and shown in fig. 1.5 (c), Liu *et al.* [44] proposed to adapt the similarity metric to choose and label examples from the large unlabeled pool. In particular, the proposed approach by Liu *et al.* [44] works on three data sources: base, novel, and unlabeled auxiliary datasets. As in fig. 1.5 (c), the approach can use both unlabeled or labeled base data in the form of unsupervised (self-supervised) learning or supervised learning, respectively. Then, given the few labeled examples of the novel classes, the label information of these few observations propagates to the unlabeled examples. In this form, we would end up with an abundance of labeled data for learning a classifier. Next, a supervised model is trained in standard format using both a few novel examples and the propagated labels.

Similarly, but without any extra unlabeled data, our mixture model-based method (chap. 3) aims at increasing the adaptability by unsupervised learning within each class while performing supervised learning between classes. Additionally, dispute resemblance to the discussed research path of using real data, our associative alignment approach (chapt. 2) does not require additional unlabeled data instead of manipulating the unlabeled data.

1.4 Representation learning

Though most of the few-shot learning methods discussed in this chapter, such as distance-based methods, can be considered representation learning approaches, this section covers the works that

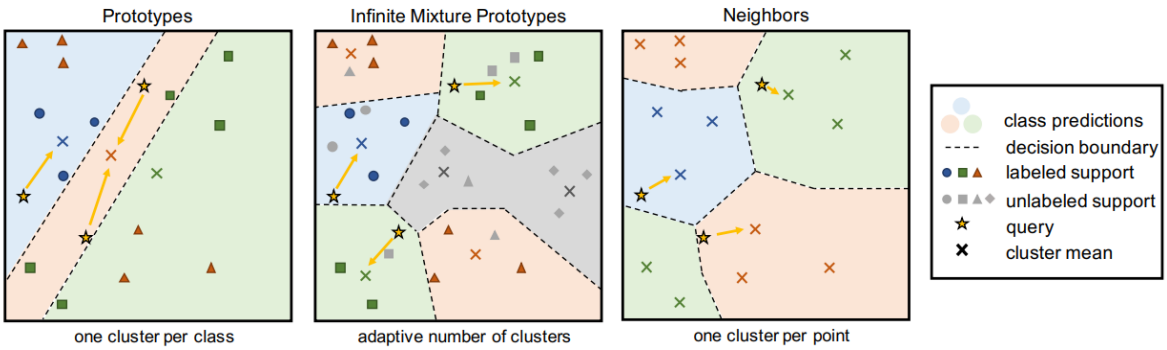


Figure 1.6: Comparing ProtoNet [9], infinite mixture prototypes (IMP) [1], and Neighbouring method. ProtoNet (left) minimize the query to the cluster centroid of the corresponding class. Infinite mixture prototypes (IMP), which can also extend to unsupervised model (gray region), performs clustering and minimize the query to the nearest centroid of the corresponding cluster. The neighboring method pushes the query to the nearest example from the corresponding cluster. Figure from Allen *et al.* [1].

are completely focused on representation learning and highly related to the proposed approaches of this thesis. In particular, we will cover infinite mixture prototypes [1], vector quantized variational auto-encoder VQ-VAE [52, 53], Swapping Assignments between multiple Views of the same image (SwAV) [54], and Feature Pyramid Network (FPN) [55], and vision transformers [21].

1.4.1 Mixture model

As discussed in sec. 1.2 and shown in the figure 1.6 (left), ProtoNet [9] aims at representing each classes with single centroids of the support set. In other words, ProtoNet constructs unimodal representations, one cluster per class. Unfortunately, unimodal approaches suffer underfitting problem from a representation perspective, since we assume that a single point (support set mean) can represent the whole distribution of a class.

As a naive solution to relax the unimodal assumption, we can train the model to learn a nearest neighboring feature space 1.6 (right) by mapping each query to the nearest support example of its corresponding class at the embedding space. In other words, instead of building a centroid-based feature space, *how about building 1-nearest neighbor*? Indeed with careful consideration, the neighboring methods would result in overfitting from a data representation perspective since the network would easily copy the input space at the embedding space without reducing the within-class distance. Therefore, ProtoNet and 1-nearest neighbor stays are thus two ends of a spectrum from simple to complex decision boundaries, respectively [1].

To balance the discussed overfitting-underfitting trade-off between nearest neighbour and centroid nearest (ProtoNet), Allen *et al.* [1] proposed infinite mixture prototypes (IMP) to cluster labeled and unlabeled data into multi-modal prototypes, presented in the figure 1.6 (middle). Indeed, IMP is motivated by infinite mixture model [56] which explores Bayesian methods [57, 58] to infer the number

cluster in a multi-modal approach.

Respectively, IMP clusters the support set using a classical clustering algorithm called DP-means that can infer a dynamic number of clusters for each class. Then, clusters mean μ_c are computed the be the within-class mixture model. Finally, the inference of the query example $\mathbf{x}$ proceeds by finding the nearest centroids. To this end, IMP computes the posterior probability of class y given $\mathbf{x}$ by the softmax over distances to the closest cluster in each class n:

$$p_\theta(y = n|\mathbf{x}) = \frac{\exp(-d(f_\theta(\mathbf{x}), \mu_{c_n^*}))}{\sum_n \exp(-d(f_\theta(\mathbf{x}), \mu_{c_n^*}))}, \tag{1.6}$$

where $d(\cdot)$ is the distance function such as euclidean function, $c_n^* = \text{argmin}_{c:l_c=n} d(f_\theta(\mathbf{x}), \mu_n)$ indexing the cluster, where each cluster c has label l_c.

The presented IMP [1], unfortunately, suffers from some limitations. First, IMP achieves multi-modality by an offline post hoc DP-means algorithm non-differentiable in a non-end-to-end way. As a result, we would have two learning models: neural network and clustering algorithm, that work in a non-synchronized manner. Second, IMP performs temporary clustering inside each batch with a few examples per class offline. Assuming that DP-means, as a classical clustering algorithm, would result in informative clusters with a small support set would be problematic.

In chapter 3, we propose mixture-based feature space learning (MixtFSL) to compensate for the discussed problems of IMP, which infer the multi-modal representation in a fully differentiable end-to-end model without any post hoc algorithm. Our MixtFSL defines multiple trainable components per class to capture the mixture model (the internal distribution) with the base classes while performing the representation learning between the base classes.

1.4.2 Vector quantized-variational autoencoder (VQ-VAE)

Generative methods are a class of algorithms to learn the underlying data distribution. Currently, there is a tremendous interest in employing generative methods for representation learning. As an example of generative models, variational autoencoder (VAE) [59] aims to learn the underlying distribution $p(\mathbf{x})$ of the input data $\mathbf{x}$ by mapping the data set's unknown distribution to one of the tractable distribution like a Gaussian using a parameterized encoder function. To do so, we use a decoder that takes the embedded sample and reconstructs it. Vector quantized variational autoencoder (VQ-VAE) aims to discretize the feature space using quantized differentiable weight vectors.

Figure 1.7 presents the idea of VQ-VAE [52, 53], where we have a encoder that maps the input space to the quantized vector space, unlike typical VAE that projects the input into a tractable continuous distribution. However, VQ-VAE proposes the following posterior categorical distribution $q(z|\mathbf{x})$ probability as the following one-hot:

$$q(z = k|\mathbf{x}) = \begin{cases} 1 & \text{for k = argmin}_j ||z_e(\mathbf{x}) - e_j||_2 \\ 0 & \text{otherwise,} \end{cases} \tag{1.7}$$

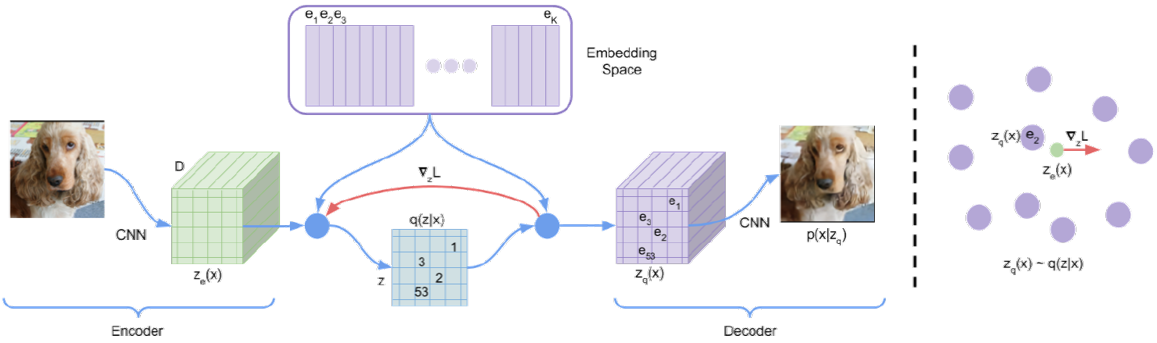


Figure 1.7: Vector quantized variational auto-encoder (VQ-VAE) proposed by Oord *et al.* [53]. In the embedding space, the output of the encoder $z(\mathbf{x})$ is mapped to the most similar point e_2. Then, e_2 is decoded to reconstruct the input image. Figure is from [53].

where $z_e(\mathbf{x})$ is the embedding obtained by the encoder network, and $z_e(\mathbf{x})$ is get though discretisation bottleneck followed by mapping onto the closest element e by:

$$z_q(\mathbf{x}) = e_k, \quad \text{where} \quad k = \text{argmin}_j ||z_e(\mathbf{x}) - e_j||_2 \tag{1.8}$$

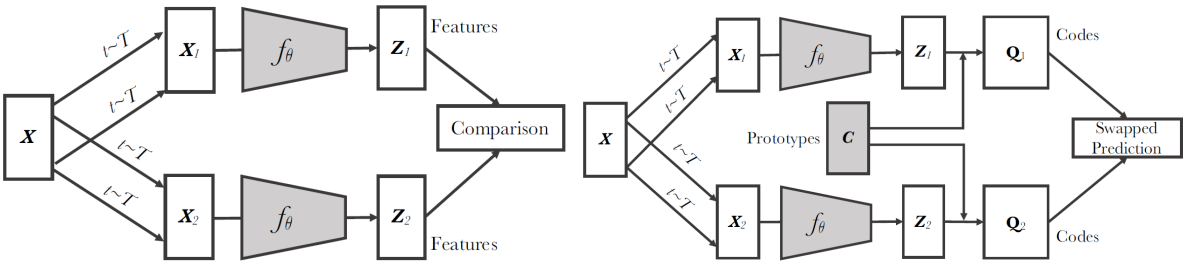
Though our MixtFSL (chap. 3) is related to VQ-VAE by employing quantized vectors, MixtFSL is a mixture model representation learning instead of a generative model. Indeed, MixtFSL is a few-shot image classification model to capture the between-class representation. Meanwhile, MixtFSL is the mixture model-based model to capture the in-class representations in unsupervised learning while capturing class representation with supervised learning.

1.4.3 Swapping assignments between multiple views

In the previous sections, we discuss the relation of this thesis to some of the unsupervised learning approaches. This thesis is also related to swapping assignments between multiple views (SwAV) [54] which is self-supervised learning (SSL). Generally, the SSL approach aims to learn from unlabeled sample data to conclude regression and classification tasks. In fact, SSL is different from unsupervised learning approaches that typically focused on learning from unlabeled data for clustering, grouping, and dimensionality reduction.

Figure 1.8 presents the difference between SwAV [54] and the stat-of-the-art SSL methods. Under SSL representation learning context, while contrastive instance learning research direction of SSL (presented in figure 1.8; left) is based on the discrepancy between different views of an input image, SwAV employs auxiliary trainable prototypes to enhance the representation learning. To this end, SwAV [54] uses a set of centroids as shown in figure 1.8 (right).

In particular, given two different views of an input image $\mathbf{x}_1$ and $\mathbf{x}_2$, Caron *et al.* [54] proposed to compute their embedding $\mathbf{z}_1$ and $\mathbf{z}_2$. Then, SwAV computes the code $\mathbf{q}_1$ and $\mathbf{q}_2$ by matching $\mathbf{z}_1$ and



(a) Contrastive instance learning

(b) SwAV [54]

Figure 1.8: Comparing contrastive instance learning (a) and Swapping Assignments between multiple Views of the same image (SwAV) (b). Figure from [54]. While contrastive learning maps different views of an image, SwAV uses persistent learnable components to swap the input image's different views. Thus, SwAV does not directly compare image features, unlike contrastive instance learning.

$\mathbf{z}_2$ to the K prototypes $\{\mathbf{c}_1, ..., \mathbf{c}_K\}$. Finally, SwAV swapped the prediction and compute the following self-supervised loss:

$$\mathcal{L}(\mathbf{z}_1, \mathbf{z}_2) = \ell(\mathbf{z}_1, \mathbf{q}_2) + \ell(\mathbf{z}_2, \mathbf{q}_1), \tag{1.9}$$

where $\ell(\mathbf{z}, \mathbf{q})$ measures the similarity between an embedding $\mathbf{z}$ and a code $\mathbf{q}$.

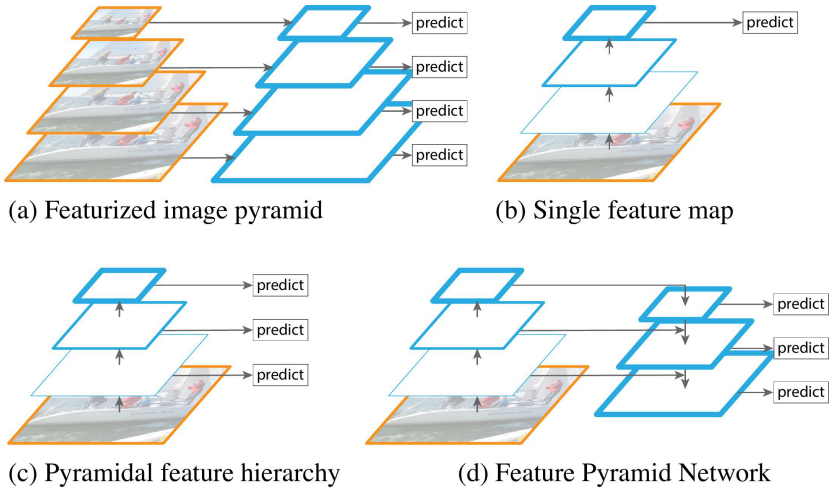
The idea of using learning centroids components in SwAV is related to MixtFSL of chapter 3. However, MixtFSL uses the multiple learning components to learn the within-class mixture representation simultaneously with the between-class representations in supervised few-shot image classification. In other words, our MixtFSL aims to learn between and with class distributions, and MixtFSL is different from SwAV, which is a self-supervised approach.

1.4.4 Feature pyramids

Our presented method in chapter 4 is also related to feature pyramids [55], a popular approach for object detection tasks obtained in several forms. For example, the traditionally featured image pyramids (illustrated in fig. 1.9) get pyramid representation in different scales to help detect objects on different scales.

Initial advances of ConvNet for object detection aimed to imitate a single feature map with non-hand-designed representation learning. Therefore, a single feature map [60, 61] is proposed to detect an object at the end of ConvNet. In addition, Liu *et al.* [62] proposed Single Shot Detector (SSD) to use a ConvNet's pyramidal feature hierarchy as if it were a featured image pyramid (fig 1.9(c)). Under object detection, feature pyramid networks combine low-resolution (but semantically strong features) with high-resolution to semantically weak ones.

To take advantage of the pyramidal nature of ConvNet's feature hierarchy, Lin *et al.* [55] presents a feature pyramid with strong semantics at different scales. In particular, Lin *et al.* [55] proposed



(a) Featurized image pyramid (b) Single feature map

(c) Pyramidal feature hierarchy (d) Feature Pyramid Network

Figure 1.9: The idea of feature pyramid networks (FPN) for object detection. (a) traditional featurized image pyramid at different image scale. (b) single feature-based object detection system. (c) ConvNet's based hierarchy feature pyramid. (d) the proposed Feature Pyramid Network (FPN) by Lin *et al.* [55]. Figure is from [55].

Feature Pyramid Network (FPN) that combines low-resolution, semantically strong features with high-resolution, semantically weak features via a top-down pathway and lateral connections (fig. 1.9(d)).

As presented in chapter 4, our proposed set-feature extractor (SetFeat) architecture is related to pyramidal feature representation learning since it extracts the pyramidal form of feature sets. However, our SetFeat extracts feature to build a set-based representation learning for few-shot image classification, compared to the discussed pyramidal methods for object detection. Additionally, our proposed SetFeat employs attention-based parametrized functions called mappers to extract features from the different scaled features at ConvNet instead of using different feature scales of feature pyramids.

1.4.5 Vision transformers

After discussing how our SetFeat (at chapter 3) is related to FPN [55] at sec. 1.4.4, lets discuss how SetFeat is inspired to extract independent features from different with self-attention [20] mechanism. Here, we adapted the self-attention module to differentially weight the features obtained at different stages of the convolution neural networks.

Attention-based representation learning approaches gained high interests due to their superiority in natural language processing (NLP) and computer vision tasks. Vaswani *et al.* [20] proposed stacked multiple layers of multi-head attention to build a deep learning model, where multiple self-attention model (fig. 1.10 left) are concatenated to build a multi-head attention presented in fig. 1.10 right. Particularly, the i-th self-attention is first computed using two parameterized query $Q(\cdot|\theta_i^q)$ and key $K(\cdot|\theta_i^k)$ functions parametrized by θ_i^q and θ_i^k, respectively:

$$\boldsymbol{\beta}_i = \text{Softmax}\left(Q(\mathbf{z}|\theta_i^q)K(\mathbf{z}|\theta_i^k)^\top/\sqrt{d_k}\right), \tag{1.10}$$

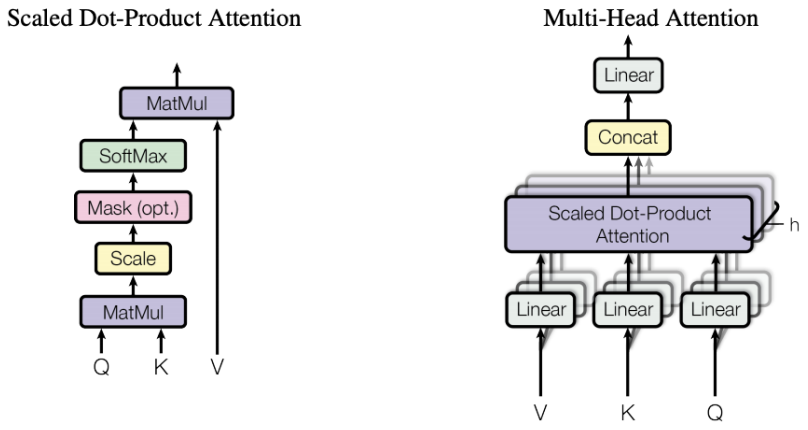


Figure 1.10: Scaled dot-product attention (left) and multi-head attention (right) module proposed by [20]. (left) in scaled do-product attention and with three trainable components, query (Q), key (K), and value (V), we compute the self-attention operation. (right) the stack of single-scaled dot-product attention builds multi-head attention. Figure from from [20].

where $\boldsymbol{\beta}_i \in \mathbb{R}^{P \times P}$ is the attention score over the patches of $\mathbf{z}$, and $\sqrt{d_k}$ is the scaling factor. Then, we compute the dot-product attention over the patches of $\boldsymbol{\beta}_i$ using a value function $V(\cdot|\theta_i^v)$ parametrized by θ_i^v in the following form:

$$\mathbf{a}_i = \boldsymbol{\beta}_i \, V(\mathbf{z}|\theta_i^v) \,, \tag{1.11}$$

where $\mathbf{a}_i \in \mathbb{R}^{P \times D^a}$ consists of P patches of D^a dimensions, and D^a is the dimension of $\mathbf{z}_b$.

Dosovitskiy *et al.* [21] extended multi-head attention of NLP to image classification task by proposing vision transformers (ViT) presented in figure 1.11. First, the input image is divided into small patches, then a linear projection layer embeds each patch and prepares them for a transformer encoder. Inside the transformer encoder, a multi-head self-attention algorithm aims at modeling the relation between embedded patches. Here, the self-attention model is originally proposed by [20] in the form of stacked multi-head attention networks based on scaled dot-product attention presented in fig. 1.10. At the end of multi-layered transformer encoder, a MLP head is employed as classification layer.

The third contribution of this thesis proposes to extract feature sets to apply our set-to-set matching. Our matching feature sets are related to vision transformers [20] and [21] from extracting a single feature point of view, and it is related to deep set from handling set matching. However, unlike transformers, our proposed attention modules are single layered and does not concatenate the self-attention modules. Indeed the representation power of our adapted light weighted attention modules are obtained from per convolution layers.

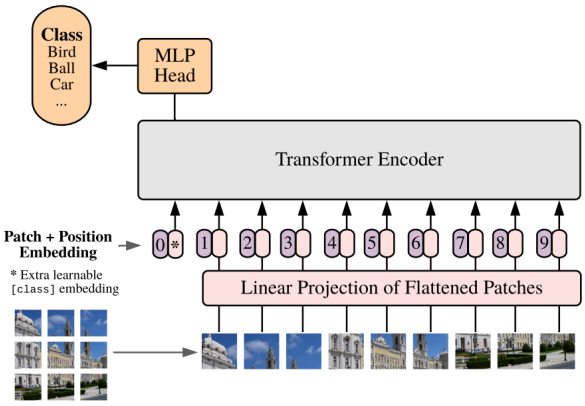


Figure 1.11: Vision transformer (ViT) adapts self-attention-based transformer for image classification task. Figure from [21].

Chapter 2

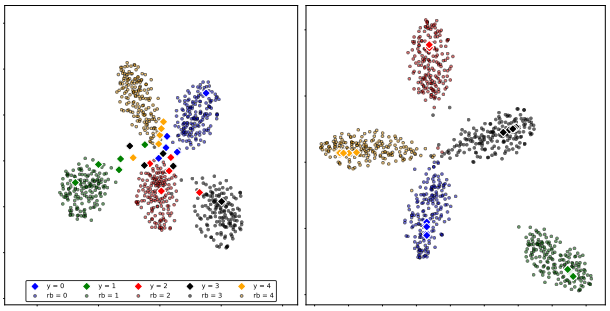
Associative Alignment for Few-shot Image Classification

2.1 Résumé

La classification d'images avec peu d'exemples (few-shot classification) vise à former un modèle à partir de quelques exemples seulement pour chacune des nouvelles classes. Cet article propose l'idée d'un alignement associatif pour exploiter une partie des données de base en les alignant avec les exemples d'entraînement pour les nouvelles classes. Nous proposons deux stratégies d'alignement associatif: 1) une fonction de perte qui minimise la distance entre les échantillons de base liés et le centroïde de nouvelles instances dans l'espace des caractéristiques, et 2) une perte d'alignement antagoniste conditionnelle basée sur la distance de Wasserstein. Des expériences sur quatre jeux de données standards et trois réseaux populaires démontrent que la combinaison de notre perte d'alignement basée sur les centroïdes entraîne des améliorations de la précision absolue de 4,4%, 1,2% et 6,2% dans l'apprentissage avec 5 données d'entraînement par rapport à l'état de l'art pour la reconnaissance d'objets, la classification granuleuse et l'adaptation inter-domaines, respectivement.

2.2 Abstract

Few-shot image classification aims at training a model from only a few examples for each of the "novel" classes. This paper proposes the idea of associative alignment for leveraging part of the base data by aligning the novel training instances to the closely related ones in the base training set. This expands the size of the effective novel training set by adding extra "related base" instances to the few novel ones, thereby allowing a constructive fine-tuning. We propose two associative alignment strategies: 1) a metric-learning loss for minimizing the distance between related base samples and the centroid of novel instances in the feature space, and 2) a conditional adversarial alignment loss based on the Wasserstein distance. Experiments on four standard datasets and three backbones demonstrate that combining our centroid-based alignment loss results in absolute accuracy improvements of 4.4%, 1.2%,



(a) before alignment (b) after alignment

Figure 2.1: The use of many related bases (circles) in addition to few novel classes samples (diamonds) allows better discriminative models: (a) using directly related bases may not properly capture the novel classes; while (b) aligning both related base and novel training instances (in the feature space) provides more relevant training data for classification. Plots are generated with t-SNE [63] applied to the ResNet-18 feature embedding before (a) and after (b) the application of the centroid alignment. Points are color-coded by class.

and 6.2% in 5-shot learning over the state of the art for object recognition, fine-grained classification, and cross-domain adaptation, respectively.

2.3 Introduction

Despite recent progress, generalizing on new concepts with little supervision is still a challenge in computer vision. In the context of image classification, few-shot learning aims to obtain a model that can learn to recognize novel image classes when very few training examples are available.

Meta-learning [7, 8, 9, 10] is a possible approach to achieve this, by extracting common knowledge from a large amount of labeled data (the "base" classes) to train a model that can then learn to classify images from "novel" concepts with only a few examples. This is achieved by repeatedly sampling small subsets from the large pool of base images, effectively simulating the few-shot scenario. Standard transfer learning has also been explored as an alternative method [3, 4, 5]. The idea is to pre-train a network on the base samples and then fine-tune the classification layer on the novel examples. Interestingly, Chen *et al.* [3] demonstrated that doing so performs on par with more sophisticated meta-learning strategies. It is, however, necessary to freeze the feature encoder part of the network when fine-tuning on the novel classes since the network otherwise overfits the novel examples. We hypothesize that this hinders performance and that gains could be made if the entire network is adapted to the novel categories.

In this paper, we propose an approach that simultaneously prevents overfitting without restricting the learning capabilities of the network for few-shot image classification. Our approach relies on the standard transfer learning strategy [3] as a starting point, but subsequently exploits base categories that are most similar (in the feature space) to the few novel samples to effectively provide additional training examples. We dub these similar categories the "related base" classes. Of course, the related

base classes represent different concepts than the novel classes, so fine-tuning directly on them could confuse the network (see fig. 2.1-(a)). The key idea of this paper is to *align*, in feature space, the novel examples with the related base samples (fig. 2.1-(b)).

To this end, we present two possible solutions for associative alignment: by 1) centroid alignment, inspired by ProtoNet [9], benefits from explicitly shrinking the intra-class variations and is more stable to train, but makes the assumption that the class distribution is well-approximated by a single mode. Adversarial alignment, inspired by WGAN [14], does not make that assumption, but its train complexity is greater due to the critic network. We demonstrate, through extensive experiments, that our centroid-based alignment procedure achieves state-of-the-art performance in few-shot classification on several standard benchmarks. Similar results are obtained by our adversarial alignment, which shows the effectiveness of our associative alignment approach.

We present the following contributions. First, we propose two approaches for aligning novel to related base classes in the feature space, allowing for effective training of entire networks for few-shot image classification. Second, we introduce a strong baseline that combines standard transfer learning [3] with an additive angular margin loss [15], along with early stopping to regularize the network while pre-training on the base categories. We find that this simple baseline actually improves on the state of the art, in the best case by 3% in overall accuracy. Third, we demonstrate through extensive experiments—on four standard datasets and using three well-known backbone feature extractors—that our proposed centroid alignment significantly outperforms the state of the art in three types of scenarios: generic object recognition (gain of 1.7%, 4.4% 2.1% in overall accuracy for 5-shot on *mini*-ImageNet, tieredImageNet and FC100 respectively), fine-grained classification (1.2% on CUB), and cross-domain adaptation (6.2% from *mini*-ImageNet to CUB) using the ResNet-18 backbone.

2.4 Related work

The main few-shot learning approaches can be broadly categorized into meta-learning and standard transfer learning. In addition, data augmentation and regularization techniques (typically in meta-learning) have also been used for few-shot learning. We briefly review relevant works in each category below. Note that several different computer vision problems such as object counting [64], video classification [65], motion prediction [66], and object detection [67] have been framed as few-shot learning. Here, we mainly focus on works from the image classification literature.

Meta-learning This family of approaches frames few-shot learning in the form of episodic training [7, 8, 67, 9, 25, 68, 69, 70]. An episode is defined by pretending to be in a few-shot regime while training on the base categories, which are available in large quantities. Initialization- and metric-based approaches are two variations on the episodic training scheme relevant for this work. Initialization-based methods [7, 24, 23] learn an initial model able to adapt to few novel samples with a small number of gradient steps. In contrast, our approach performs a larger number of updates, but requires that the

alignment be maintained between the novel samples and their related base examples. Metric-based approaches [9, 10, 26, 71, 27, 28, 29, 17, 30, 31, 32, 33] learn a metric with the intent of reducing the intra-class variations while training on base categories. For example, ProtoNet [9] were proposed to learn a feature space where instances of a given class are located close to the corresponding prototype (centroid), allowing accurate distance-based classification. Our centroid alignment strategy borrows from such distance-based criteria but uses it to match the distributions in the feature space instead of building a classifier.

Standard transfer learning The strategy behind this method is to pre-train a network on the base classes and subsequently fine-tune it on the novel examples [3, 4, 5]. Despite its simplicity, Chen *et al.* [3] recently demonstrated that such an approach could result in similar generalization performance compared to meta-learning when deep backbones are employed as feature extractors. However, they have also shown that the weights of the pre-trained feature extractor must remain frozen while fine-tuning due to the propensity for overfitting. Although the training procedure we are proposing is similar to standard fine-tuning in base categories, our approach allows the training of the entire network, thereby increasing the learned model capacity while improving classification accuracy.

Regularization trick Wang *et al.* [72] proposed regression networks for regularization purposes by refining the parameters of the fine-tuning model to be close to the pre-trained model. More recently, Lee *et al.* [73] exploited the implicit differentiation of a linear classifier with hinge loss and $\mathcal{L}_2$ regularization to the CNN-based feature learner. Dvornik *et al.* [74] uses an ensemble of networks to decrease the classifiers variance.

Data augmentation Another family of techniques relies on additional data for training in a few-shot regime, most of the time following a meta-learning training procedure [39, 40, 41, 43, 38, 37, 45, 46, 49, 48]. Several ways of doing so have been proposed, including Feature Hallucination (FH) [38], which learns mappings between examples with an auxiliary generator that then hallucinates extra training examples (in the feature space). Subsequently, Wang *et al.* [46] proposed to use a GAN for the same purpose, and thus address the poor generalization of the FH framework. Unfortunately, it has been shown that this approach suffers from mode collapse [41]. Instead of generating artificial data for augmentation, others have proposed methods to take advantage of additional unlabeled data [42, 16, 51, 47]. Liu *et al.* [44] propose to propagate labels from few labeled data to many unlabeled data, akin to our detection of related bases. We also rely on more data for training, but in contrast to these approaches, our method does not need any new data, nor does it require to generate any. Instead, we exploit the data that is *already available* in the base domain and align the novel domain to the relevant base samples through fine-tuning.

Previous work has also exploited base training data, most related to ours are the works of [39] and [75]. Chen *et al.* [39] propose to use an embedding and deformation sub-networks to leverage additional training samples, whereas we rely on a single feature extractor network which is much simpler to

implement and train. Unlike random base example sampling [39] for interpolating novel example deformations in the image space, we propose to borrow the internal distribution structure of the detected related classes in feature space. Besides, our alignment strategies introduce extra criteria to keep the focus of the learner on the novel classes, which prevents the novel classes from becoming outliers. Focused on object detection, Lim *et al.* [75] proposes a model to search similar object categories using a sparse grouped Lasso framework. Unlike [75], we propose and evaluate two associative alignments in the context of few-shot image classification.

From the alignment perspective, our work is related to Jiang *et al.* [76] which stays in the context of zero-shot learning, and proposes a coupled dictionary matching in visual-semantic structures to find matching concepts. In contrast, we propose associative base-novel class alignments along with two strategies for enforcing the unification of the related concepts.

2.5 Preliminaries

Let us assume that we have a large base dataset $\mathcal{X}^b = \{(\mathbf{x}_i^b, y_i^b)\}_{i=1}^{N^b}$, where $\mathbf{x}_i^b \in \mathbb{R}^d$ is the i-th data instance of the set and $y_i^b \in \mathcal{Y}^b$ is the corresponding class label. We are also given a small amount of novel class data $\mathcal{X}^n = \{(\mathbf{x}_i^n, y_i^n)\}_{i=1}^{N^n}$, with labels $y_i^n \in \mathcal{Y}^n$ from a set of distinct classes $\mathcal{Y}^n$. Few-shot classification aims to train a classifier with only a few examples from each of the novel classes (e.g., 5 or even just 1). In this work, we used the standard transfer learning strategy of Chen *et al.* [3], which is organized into the following two stages.

Pre-training stage The learning model is a neural network composed of a feature extractor $f(\cdot|\theta)$, parameterized by θ, followed by a linear classifier $c(\mathbf{x}|\mathbf{W}) \equiv \mathbf{W}^\top f(\mathbf{x}|\theta)$, described by matrix $\mathbf{W}$, ending with a scoring function such as softmax to produce the output. The network is trained from scratch on examples from the base categories $\mathcal{X}^b$.

Fine-tuning stage In order to adapt the network to the novel classes, the network is subsequently fine-tuned on the few examples from $\mathcal{X}^n$. Since overfitting is likely to occur if all the network weights are updated, the feature extractor weights θ are frozen, with only the classifier weights $\mathbf{W}$ being updated in this stage.

2.6 Associative alignment

Freezing the feature extractor weights θ indeed reduces overfitting, but also limits the learning capacity of the model. In this paper, we strive for the best of both worlds and present an approach which controls overfitting while maintaining the original learning capacity of the model. We borrow the internal distribution structure of a subset of *related* base categories, $\mathcal{X}^{rb} \subset \mathcal{X}^b$. To account for the discrepancy between the novel and related base classes, we propose to *align* the novel categories to the related base categories in feature space. Such a mapping allows for a bigger pool of training data while making

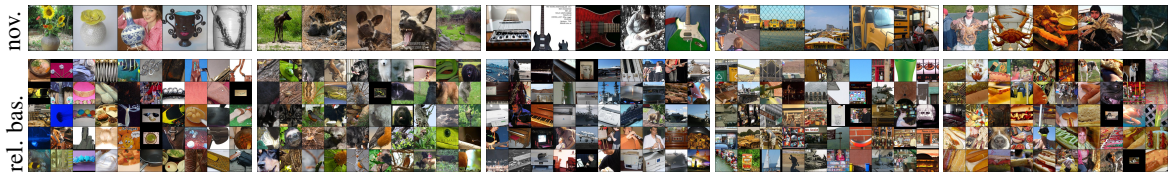


Figure 2.2: Results of related base algorithm in a 5-way 5-shot scenario. Each column represents a different novel class. The top row shows the 5 novel instances, while the bottom row shows 60 randomly selected related base instances with $B = 10$.

instances of these two sets more coherent. Note that, as opposed to [39], we do not modify the related base instances in any way: we simply wish to align novel examples to the distributions of their related class instances.

In this section, we first describe how the related base classes are determined. Then, we present our main contribution: the "centroid associative alignment" method, which exploits the related base instances to improve classification performance on novel classes. We conclude by presenting an alternative associative alignment strategy, which relies on an adversarial framework.

2.6.1 Detecting the related bases

We develop a simple, yet effective procedure to select a set of base categories related to a novel category. Our method associates B base categories to each novel class. After training $c(f(\cdot|\theta)|\mathbf{W})$ on $\mathcal{X}^b$, we first fine-tune $c(\cdot|\mathbf{W})$ on $\mathcal{X}^n$ while keeping θ fixed. Then, we define $\mathbf{M} \in \mathbb{R}^{K^b \times K^n}$ as a base-novel similarity matrix, where K^b and K^n are respectively the number of classes in $\mathcal{X}^b$ and $\mathcal{X}^n$. An element $m_{i,j}$ of the matrix $\mathbf{M}$ corresponds to the ratio of examples associated to the i-th base class that are classified as the j-th novel class:

$$m_{i,j} = \frac{1}{|\mathcal{X}_i^b|} \sum_{(\mathbf{x}_l^b, \cdot) \in \mathcal{X}_i^b} \mathbb{I}\left[j = \arg\max_{k=1}^{K^n} \left(c_k(f(\mathbf{x}_l^b|\theta) \,|\, \mathbf{W}) \right) \right], \tag{2.1}$$

where $c_k(f(\mathbf{x}|\theta)|\mathbf{W})$ is the classifier output $c(\cdot|\mathbf{W})$ for class k. Then, the B base classes with the highest score for a given novel class are kept as the related base for that class. Fig. 2.2 illustrates example results obtained with this method in a 5-shot, 5-way scenario.

2.6.2 Centroid associative alignment

Let us assume the set of instances $\mathcal{X}_i^n$ belonging to the i-th novel class $i \in \mathcal{Y}^n$, $\mathcal{X}_i^n = \{(\mathbf{x}_j^n, y_j^n) \in \mathcal{X}^n \,|\, y_j^n = i\}$, and the set of related base examples $\mathcal{X}_i^{rb}$ belonging to the same novel class i according to the $g(\cdot|\mathbf{M})$ mapping function, $\mathcal{X}_i^{rb} = \{(\mathbf{x}_j^b, y_j^b) \in \mathcal{X}^{rb} \,|\, g(y_j|\mathbf{M}) = i\}$. The function $g(y_j|\mathbf{M}) : \mathcal{Y}^b \rightarrow \mathcal{Y}^n$ maps base class labels to the novel ones according to the similarity matrix $\mathbf{M}$. We wish to find an alignment transformation for matching probability densities $p(f(\mathbf{x}_{i,k}^n \,|\, \theta))$ and $p(f(\mathbf{x}_{i,l}^{rb} \,|\, \theta))$. Here, $\mathbf{x}_{i,k}^n$ is the k-th element from class i in the novel set, and $\mathbf{x}_{i,l}^{rb}$ is the l-th element from class i in

Algorithm 1:
Centroid alignment.

Input: pre-trained model $c(f(\cdot|\theta)|\mathbf{W})$, novel class $\mathcal{X}^n$, related base set $\mathcal{X}^{rb}$.
Output: fine-tuned $c(f(\cdot|\theta)|\mathbf{W})$.
while *not done* **do**
  $\widetilde{\mathcal{X}}^n \leftarrow$ sample a batch from $\mathcal{X}^n$
  $\widetilde{\mathcal{X}}^{rb} \leftarrow$ sample a batch from $\mathcal{X}^{rb}$

  evaluate $\mathcal{L}_{\text{ca}}(\widetilde{\mathcal{X}}^n, \widetilde{\mathcal{X}}^{rb})$, (eq. 2.3)
  $\theta \leftarrow \theta - \eta_{\text{ca}} \nabla_\theta \mathcal{L}_{\text{ca}}(\widetilde{\mathcal{X}}^n, \widetilde{\mathcal{X}}^{rb})$
  evaluate $\mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^{rb})$, (eq. 2.7)
  $\mathbf{W} \leftarrow \mathbf{W} - \eta_{\text{clf}} \nabla_{\mathbf{W}} \mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^{rb})$
  evaluate $\mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^n)$, (eq. 2.7)
  $\mathbf{W} \leftarrow \mathbf{W} - \eta_{\text{clf}} \nabla_{\mathbf{W}} \mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^n)$
  $\theta \leftarrow \theta - \eta_{\text{clf}} \nabla_\theta \mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^n)$
end

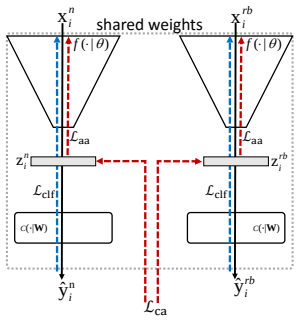


Figure 2.3: Schematic overview of our centroid alignment. The feature learner $f(\cdot|\theta)$ takes an example from novel category $\mathbf{x}^n$ and an example related base $\mathbf{x}_i^{rb}$. A Euclidean centroid based alignment loss $\mathcal{L}_{\text{ca}}$ (red arrow) aligns the encoded $\mathbf{x}_i^n$ and $\mathbf{x}_i^{rb}$. Blue arrows represent classification loss $\mathcal{L}_{\text{clf}}$.

the related base set. This approach has the added benefit of allowing the fine-tuning of all of the model parameters θ and $\mathbf{W}$ with a reduced level of overfitting.

We propose a metric-based centroid distribution alignment strategy. The idea is to enforce intra-class compactness during the alignment process. Specifically, we explicitly push the training examples from the i-th novel class $\mathcal{X}_i^n$ towards the centroid of their related examples $\mathcal{X}_i^{rb}$ in feature space. The centroid $\boldsymbol{\mu}_i$ of $\mathcal{X}_i^{rb}$ is computed by

$$\boldsymbol{\mu}_i = \frac{1}{|\mathcal{X}_i^{rb}|} \sum_{(\mathbf{x}_j, \cdot) \in \mathcal{X}_i^{rb}} f(\mathbf{x}_j|\theta)\,, \tag{2.2}$$

where N^n and N^{rb} are the number of examples in $\mathcal{X}^n$ and $\mathcal{X}^{rb}$, respectively. This allows the definition of the centroid alignment loss as

$$\mathcal{L}_{\text{ca}}(\mathcal{X}^n) = -\frac{1}{N^n N^{rb}} \sum_{i=1}^{K^n} \sum_{(\mathbf{x}_j, \cdot) \in \mathcal{X}_i^n} \log \frac{\exp[-\|f(\mathbf{x}_j|\theta) - \boldsymbol{\mu}_i\|_2^2]}{\sum_{k=1}^{K^n} \exp[-\|f(\mathbf{x}_j|\theta) - \boldsymbol{\mu}_k\|_2^2]}\,. \tag{2.3}$$

Our alignment strategy bears similarities to [9] which also uses eq. 2.3 in a meta-learning framework. In our case, we use that same equation to match distributions. Fig. 2.3 illustrates our proposed centroid alignment, and algorithm 1 presents the overall procedure. First, we update the parameters of the feature extraction network $f(\cdot|\theta)$ using eq. 2.3. Second, the entire network is updated using a classification loss $\mathcal{L}_{\text{clf}}$ (defined in sec. 2.7).

Algorithm 2:
Adversarial alignment

Input: pre-trained model $c(f(\cdot|\theta)|\mathbf{W})$, novel class $\mathcal{X}^n$, related base set $\mathcal{X}^{rb}$.
Output: fine-tuned $c(f(\cdot|\theta)|\mathbf{W})$.
while *not done* **do**
  $\widetilde{\mathcal{X}}^n \leftarrow$ sample a batch from $\mathcal{X}^n$
  $\widetilde{\mathcal{X}}^{rb} \leftarrow$ sample a batch from $\mathcal{X}^{rb}$
  for i = *0,...,*n_{critic} **do**
    evaluate $\mathcal{L}_h(\widetilde{\mathcal{X}}^n, \widetilde{\mathcal{X}}^{rb})$, (eq. 2.5)
    ▷ update critic:
    $\phi \leftarrow \phi + \eta_h \nabla_\phi \mathcal{L}_h(\widetilde{\mathcal{X}}^n, \widetilde{\mathcal{X}}^{rb})$
    $\phi \leftarrow \text{clip}(\phi, -0.01, 0.01)$
  end
  evaluate $\mathcal{L}_{\text{aa}}(\widetilde{\mathcal{X}}^n)$, (eq. 2.6)
  $\theta \leftarrow \theta - \eta_{\text{aa}} \nabla_\theta \mathcal{L}_{\text{aa}}(\widetilde{\mathcal{X}}^n)$
  evaluate $\mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^{rb})$, (eq. 2.7)
  $\mathbf{W} \leftarrow \mathbf{W} - \eta_{\text{clf}} \nabla_{\mathbf{W}} \mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^{rb})$
  evaluate $\mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^n)$, (eq. 2.7)
  $\mathbf{W} \leftarrow \mathbf{W} - \eta_{\text{clf}} \nabla_{\mathbf{W}} \mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^n)$
  $\theta \leftarrow \theta - \eta_{\text{clf}} \nabla_\theta \mathcal{L}_{\text{clf}}(\widetilde{\mathcal{X}}^n)$
end

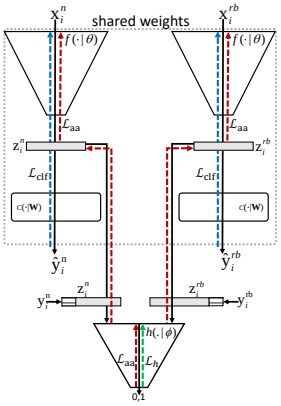


Figure 2.4: Overview of our adversarial alignment. The feature learner $f(\cdot|\theta)$ takes an image $\mathbf{x}_i^n$ from the i-th novel class and an example $\mathbf{x}_i^{rb}$ of the related base. The critic $h(\cdot|\phi)$ takes the feature vectors and the one-hot class label vector. Green, red and blue arrows present the critic $\mathcal{L}_h$, adversarial $\mathcal{L}_{aa}$ and classification $\mathcal{L}_{\text{clf}}$ losses respectively.

2.6.3 Adversarial associative alignment

As an alternative associative alignment strategy, and inspired by WGAN [14], we experiment with training the encoder $f(\cdot|\theta)$ to perform adversarial alignment using a conditioned critic network $h(\cdot|\phi)$ based on Wasserstein-1 distance between two probability densities p_x and p_y:

$$D(p_x, p_y) = \sup_{\|h\|_L \leq 1} \mathbb{E}_{x \sim p_x}[h(x)] - \mathbb{E}_{x \sim p_y}[h(x)]\,, \tag{2.4}$$

where sup is the supremum, and h is a 1-Lipschitz function. Similarly to Arjovsky *et al.* [14], we use a parameterized critic network $h(\cdot|\phi)$ conditioned by the concatenation of the feature embedding of either $\mathbf{x}_i^n$ or $\mathbf{x}_j^{rb}$, along with the corresponding label y_i^n encoded as a one-hot vector. Conditioning $h(\cdot|\phi)$ helps the critic in matching novel categories and their corresponding related base categories. The critic $h(\cdot|\phi)$ is trained with loss

$$\begin{aligned}\mathcal{L}_h(\mathcal{X}^n, \mathcal{X}^{rb}) = \frac{1}{N^{rb}} \sum_{(\mathbf{x}_i^{rb}, y_i^{rb}) \in \mathcal{X}^{rb}} h\left([f(\mathbf{x}_i^{rb}|\theta)\ y_i^{rb}]\,|\,\phi\right) \\ - \frac{1}{N^n} \sum_{(\mathbf{x}_i^n, y_i^n) \in \mathcal{X}^n} h\left([f(\mathbf{x}_i^n|\theta)\ y_i^n]\,|\,\phi\right)\,,\end{aligned} \tag{2.5}$$

where, $[\cdot]$ is the concatenation operator. Then, the encoder parameters θ are updated using

$$\mathcal{L}_{\text{aa}}(\mathcal{X}^n) = \frac{1}{K^n} \sum_{(\mathbf{x}_i^n, y_i^n) \in \mathcal{X}^n} h\left([f(\mathbf{x}_i^n|\theta)\ y_i^n]|\phi\right) . \tag{2.6}$$

Algorithm 2 summarizes our adversarial alignment method. First, we perform the parameter update of critic $h(\cdot|\phi)$ using eq. 2.5. Similar to WGAN [14], we perform n_{critic} iterations to optimize h, before updating $f(\cdot|\theta)$ using eq. 2.6. Finally, the entire network is updated by a classification loss $\mathcal{L}_{\text{clf}}$ (defined in sec. 2.7).

2.7 Establishing a strong baseline

Before evaluating our alignment strategies in sec. 2.8, we first establish a strong baseline for comparison by following the recent literature. In particular, we build on the work of Chen *et al.* [3] but incorporate a different loss function and episodic early stopping on the pre-training stage.

2.7.1 Classification loss functions

Deng *et al.* [15] have shown that an additive angular margin ("arcmax" hereafter) outperforms other metric learning algorithms for face recognition. The arcmax has a metric learning property since it enforces a geodesic distance margin penalty on the normalized hypersphere, which we think can be beneficial for few-shot classification by helping keep class clusters compact and well-separated.

Let $\mathbf{z}$ be the representation of $\mathbf{x}$ in feature space. As per [15], we transform the logit as $\mathbf{w}_j^\top \mathbf{z} = \|\mathbf{w}_j\| \|\mathbf{z}\| \cos\varphi_j$, where φ_j is the angle between $\mathbf{z}$ and $\mathbf{w}_j$, the j-th column in the weight matrix $\mathbf{W}$. Each weight $\|\mathbf{w}_j\| = 1$ by l_2 normalization. Arcmax adds an angular margin m to the distributed examples on a hypersphere:

$$\mathcal{L}_{\text{clf}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(s\cos(\varphi_{y_i} + m))}{\exp(s\cos(\varphi_{y_i} + m)) + \sum_{\forall j \neq y_i} \exp(s\cos\varphi_j)} , \tag{2.7}$$

where s is the radius of the hypersphere on which $\mathbf{z}$ is distributed, N the number of examples, and m and s are hyperparameters (see sec. 2.8.1). The overall goal of the margin is to enforce inter-class discrepancy and intra-class compactness.

2.7.2 Episodic early stopping

A fixed number of epochs in the pre-training stage has been commonly used (e.g., [7, 9, 10, 3]), but this might hamper performance in the fine-tuning stage. Using validation error, we observe the necessity of early-stopping in pre-training phase (see supp. mat. for a validation error plot). We thus make the use of episodic early stopping using validation set at pre-training time, specifically by stopping the training when the mean accuracy over a window of recent epochs starts to decrease. The best model in the window is selected as the final result.

2.8 Experimental validation

In the following, we are conducting an experimental evaluation and comparison of the proposed associative alignment strategies for few-shot learning. First, we introduce the datasets used and evaluate the strong baseline from sec. 2.7.

2.8.1 Datasets and implementation details

Datasets We present experiments on four benchmarks: *mini*-ImageNet [10], tieredImageNet [16], and FC100 [17] for generic object recognition; and CUB-200-2011 (CUB) [18] for fine-grained image classification. *mini*-ImageNet is a subset of the ImageNet ILSVRC-12 dataset [2] containing 100 categories and 600 examples per class. We used the same splits as Ravi and Larochelle [8], where 64, 16, and 20 classes are used for the base, validation, and novel classes, respectively. As a larger benchmark, the tieredImageNet [16] is also a subset of ImageNet ILSVRC-12 dataset [2], this time with 351 base, 97 validation, and 160 novel classes respectively. Derived from CIFAR-100 [77], the FC100 dataset [17] contains 100 classes grouped into 20 superclasses to minimize class overlap. Base, validation and novel splits contain 60, 20, 20 classes belonging to 12, 5, and 5 superclasses, respectively. The CUB dataset [18] contains 11,788 images from 200 bird categories. We used the same splits as Hilliard *et al.* [78] using 100, 50, and 50 classes for the base, validation, and novel classes, respectively.

Network architectures We experiment with three backbones for the feature learner $f(\cdot|\theta)$: 1) a 4-layer convolutional network ("Conv4") with input image resolution of 84×84, similar to [7, 8, 9]; 2) a ResNet-18 [79] with input size of 224×224; and 3) a 28-layers Wide Residual Network ("WRN-28-10") [80] with input size of 80×80 in 3 steps of dimension reduction. We use a single hidden layer MLP of 1024 dimensions as the critic network $h(\cdot|\phi)$ (c.f. sec. 2.6.3).

Implementation details Recall from sec. 2.5 that training consists of two stages: 1) pre-training using base categories $\mathcal{X}^b$; and 2) fine-tuning on novel categories $\mathcal{X}^n$. For pre-training, we use the early stopping algorithm from sec. 2.7.2 with a window size of 50. Standard data augmentation approaches (i.e., color jitter, random crops, and left-right flips as in [3]) have been employed, and the Adam algorithm with a learning rate of 10^{-3} and batch size of 64 is used for both pre-training and fine-tuning. The arcmax loss (eq. 2.7) is configured with $s = 20$ and $m = 0.1$ which are set by cross validation. In the fine-tuning stage, episodes are defined by randomly selecting $N = 5$ classes from the novel categories $\mathcal{X}^n$. k examples for each category are subsequently sampled ($k = 1$ and $k = 5$ in our experiments). As in Chen *et al.* [3], no standard data augmentation was used in this stage. We used episodic cross-validation to find s and m with a fixed encoder. More specifically, (s, m) were found to be $(5, 0.1)$ for the Conv4 and $(5, 0.01)$ for the WRN-28-10 and ResNet-18 backbones. The learning rate for Adam was set to 10^{-3} and 10^{-5} for the centroid and adversarial alignments respectively. Similarly to [14], 5 iterations (inner loop of algorithm 2) were used to train the critic $h(\cdot|\phi)$. We fix the number

Table 2.1: Preliminary evaluation using *mini*-ImageNet and CUB, presenting 5-way classification accuracy using the Conv4 backbone, with ± indicating the 95% confidence intervals over 600 episodes. The best result is boldfaced, while the best result *prior to this work* is highlighted in blue. Throughout this paper, "–" indicates when a paper does not report results in the corresponding scenario.

		mini-ImageNet		CUB	
	Method	**1-shot**	**5-shot**	**1-shot**	**5-shot**
meta learning	Meta-LSTM [8]	43.44 ± 0.77	55.31 ± 0.71	–	–
	MatchingNet‡ [10]	43.56 ± 0.84	55.31 ± 0.73	60.52 ± 0.88	75.29 ± 0.75
	ProtoNet‡ [9]	49.42 ± 0.78	68.20 ± 0.66	51.31 ± 0.91	70.77 ± 0.69
	MAML‡ [23]	48.07 ± 1.75	63.15 ± 0.91	55.92 ± 0.95	72.09 ± 0.76
	RelationNet‡ [30]	50.44 ± 0.82	65.32 ± 0.70	62.45 ± 0.98	76.11 ± 0.69
tr. learning	softmax†	46.40 ± 0.72	64.37 ± 0.59	47.12 ± 0.74	64.16 ± 0.71
	softmax†◇	46.99 ± 0.73	65.33 ± 0.60	45.68 ± 0.86	66.94 ± 0.84
	cosmax†	50.92 ± 0.76	67.29 ± 0.59	60.53 ± 0.83	79.34 ± 0.61
	cosmax†◇	52.04 ± 0.82	68.47 ± 0.60	60.66 ± 1.04	79.79 ± 0.75
	our baseline (sec. 2.7)	51.90 ± 0.79	69.07 ± 0.62	60.85 ± 1.07	79.74 ± 0.64
align.	adversarial	52.13 ± 0.99	70.78 ± 0.60	**63.30** ± 0.94	**81.35** ± 0.67
	centroid	**53.14** ± 1.06	**71.45** ± 0.72	62.71 ± 0.88	80.48 ± 0.81

† our implementation ◇ with early stopping ‡ implementation from [3] for CUB

of related base categories as $B = 10$ (see supp. mat. for an ablation study on B). For this reason, we used a relatively large number of categories (50 classes out of the 64 available in *mini*-ImageNet).

2.8.2 *mini*-ImageNet and CUB with a shallow Conv4 backbone

We first evaluate the new baseline presented in sec. 2.7 and our associative alignment strategies using the Conv4 backbone on the *mini*-ImageNet (see supp. mat. for evaluations in higher number of ways) and CUB datasets, with corresponding results presented in table 2.1. We note that arcmax with early stopping improves on using cosmax and softmax with and without early stopping for both the 1- and 5-shot scenarios, on both the *mini*-ImageNet and CUB datasets. We followed the same dataset split configuration, network architecture, and implementation details given in [3] for our testing. Our centroid associative alignment outperforms the state of the art in all the experiments, with gains of 1.24% and 2.38% in 1- and 5-shot over our baseline on *mini*-ImageNet. For CUB, the adversarial alignment provides an additional gain of 0.6% and 0.87% over the centroid one.

2.8.3 *mini*-ImageNet and tieredimageNet with deep backbones

We now evaluate our proposed associative alignment on both the *mini*-ImageNet and tieredimageNet datasets using two deep backbones: ResNet-18 and WRN-28-10. Table 2.2 compares our proposed alignment methods with several approaches.

Table 2.2: *mini*-ImageNet and tieredImageNet results using ResNet-18 and WRN-28-10 backbones. ± denotes the 95% confidence intervals over 600 episodes.

		mini-ImageNet		tieredImageNet	
	Method	**1-shot**	**5-shot**	**1-shot**	**5-shot**
ResNet-18	TADAM [17]	58.50 ± 0.30	76.70 ± 0.30	–	–
	ProtoNet‡ [9]	54.16 ± 0.82	73.68 ± 0.65	61.23 ± 0.77	80.00 ± 0.55
	SNAIL [81]	55.71 ± 0.99	68.88 ± 0.92	–	–
	IDeMe-Net [39]	59.14 ± 0.86	74.63 ± 0.74	–	–
	Activation to Param. [82]	59.60 ± 0.41	73.74 ± 0.19	–	–
	MTL [83]	61.20 ± 1.80	75.50 ± 0.80	–	–
	TapNet [70]	61.65 ± 0.15	76.36 ± 0.10	63.08 ± 0.15	80.26 ± 0.12
	VariationalFSL [33]	61.23 ± 0.26	77.69 ± 0.17	–	–
	MetaOptNet* [73]	**62.64** ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
	our baseline (sec. 2.7)	58.07 ± 0.82	76.62 ± 0.58	65.08 ± 0.19	83.67 ± 0.51
	adversarial alignment	58.84 ± 0.77	77.92 ± 0.82	66.44 ± 0.61	85.12 ± 0.53
	centroid alignment	59.88 ± 0.67	**80.35** ± 0.73	**69.29** ± 0.56	**85.97** ± 0.49
WRN-28-10	LEO [25]	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.09	81.44 ± 0.12
	wDAE [43]	61.07 ± 0.15	76.75 ± 0.11	68.18 ± 0.16	83.09 ± 0.12
	CC+rot [42]	62.93 ± 0.45	79.87 ± 0.33	70.53 ± 0.51	84.98 ± 0.36
	Robust-dist++ [25]	63.28 ± 0.62	81.17 ± 0.43	–	–
	Transductive-ft [69]	65.73 ± 0.68	78.40 ± 0.52	73.34 ± 0.71	85.50 ± 0.50
	our baseline (sec. 2.7)	63.28 ±0.71	78.31 ±0.57	68.47 ±0.86	84.11 ±0.65
	adversarial alignment	64.79 ±0.93	82.02 ±0.88	73.87 ±0.76	84.95 ±0.59
	centroid alignment	**65.92** ± 0.60	**82.85** ± 0.55	**74.40** ± 0.68	**86.61** ±0.59

‡ Results are from [3] for *mini*-ImageNet and from [73] for tieredImageNet, * ResNet-12

***mini*-ImageNet** Our centroid associative alignment strategy achieves the best 1- and 5-shot classification tasks on both the ResNet-18 and WRN-28-10 backbones, with notable absolute accuracy improvements of 2.72% and 1.68% over MetaOptNet [73] and Robust-dist++ [74] respectively. The single case where a previous method achieves superior results is that of MetaOptNet, which outperforms our method by 2.76% in 1-shot. For the WRN-28-10 backbone, we achieve similar results to Transductive-ft [69] for 1-shot, but outperform their method by 4.45% in 5-shot. Note that unlike IDeMe-Net [39], SNAIL [81] and TADAM [17], which make use of extra modules, our method achieves significant improvements over these methods without any changes to the backbone.

tieredImageNet Table 2.2 also shows that our centroid associative alignment outperforms the compared methods on tieredImageNet in both 1- and 5-shot scenarios. Notably, our centroid alignment results in a gain of 3.3% and 4.41% over MetaOptNet [73] using the ResNet-18. Likewise, our centroid alignment gains 1.06% and 1.11% over the best of the compared methods using WRN-28-10.

2.8.4 FC100 and CUB with a ResNet-18 backbone

We present additional results on the FC100 and CUB datasets with a ResNet-18 backbone in table 2.3. In FC100, our centroid alignment gains 0.73% and 2.14% over MTL [83] in 1- and 5-shot respectively.

Table 2.3: Results on the FC100 and CUB dataset using ResNet-18 backbones. $\pm$ denotes the 95% confidence intervals over 600 episodes. The best result is boldfaced, while the best result *prior to this work* is highlighted in blue.

	FC100		CUB	
Method	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Robust-20 [74]	–	–	58.67 ± 0.65	75.62 ± 0.48
GNN-LFT [31]	–	–	51.51 ± 0.80	73.11 ± 0.68
RelationNet‡ [30]	–	–	67.59 ± 1.02	82.75 ± 0.58
ProtoNet‡ [9]	40.5 ± 0.6	55.3 ± 0.6	71.88 ± 0.91	87.42 ± 0.48
TADAM [17]	40.1 ± 0.4	56.1 ± 0.4	–	–
MetaOptNet† [73]	41.1 ± 0.6	55.5 ± 0.6	–	–
MTL [83]	45.1 ± 1.8	57.6 ± 0.9	–	–
Transductive-ft [69]	43.2 ± 0.6	57.6 ± 0.6	–	–
our baseline (sec. 2.7)	40.84 ± 0.71	57.02 ± 0.63	71.71 ± 0.86	85.74 ± 0.49
adversarial	43.44 ± 0.71	58.69 ± 0.56	70.80 ± 1.12	88.04 ± 0.54
centroid	**45.83** ± 0.48	**59.74** ± 0.56	**74.22** ± 1.09	**88.65** ± 0.55

‡ implementation from [3] for CUB, and from [73] for FC100

We also observe improvements in CUB with our associative alignment approaches, with the centroid alignment outperforming ProtoNet [9] by 2.3% in 1-shot and 1.2% in 5-shot. We outperform Robust-20 [74], an ensemble of 20 networks, by 4.03% and 4.15% on CUB.

2.8.5 Cross-domain evaluation

We also evaluate our alignment strategies in cross-domain image classification. Here, following [3], the base categories are drawn from *mini*-ImageNet, but the novel categories are from CUB. As shown in table 2.4, we gain 1.3% and 5.4% over the baseline in the 1- and 5-shot, respectively, with our proposed centroid alignment. Adversarial alignment falls below the baseline in 1-shot by -1.2%, but gains 5.9% in 5-shot. Overall, our centroid alignment method shows absolute accuracy improvements over the state of the art (i.e., cosmax [3]) of 3.8% and 6.0% in 1- and 5- shot respectively. We also outperform Robust-20 [74], an ensemble of 20 networks, by 4.65% for 5-shot on *mini*-ImageNet to CUB cross-domain.One could argue that the three bird categories (i.e., house finch, robin, and toucan) in *mini*-ImageNet bias the cross-domain evaluation. Re-training the approach by excluding these classes resulted in a similar performance as shown in table 2.4.

2.9 Discussion

This paper presents the idea of associative alignment for few-shot image classification, which allows for higher generalization performance by enabling the training of the entire network, still while avoiding overfitting. To do so, we design a procedure to detect related base categories for each novel class. Then, we proposed a centroid-based alignment strategy to keep the intra-class alignment while performing

Table 2.4: Cross-domain results from *mini*-ImageNet to CUB in 1-shot, 5-shot, 10-shot scenarios using a ResNet-18 backbone.

Method	1-shot	5-shot	10-shot
ProtoNet‡ [46]	–	62.02 ± 0.70	–
MAML‡ [23]	–	51.34 ± 0.72	–
RelationNet‡ [30]	–	57.71 ± 0.73	–
Diverse 20 [74]	–	66.17 ± 0.73	–
cosmax† [3]	43.06 ± 1.01	64.38 ± 0.86	67.56±0.77
our baseline (sec. 2.7)	45.60 ± 0.94	64.93 ± 0.95	68.95±0.78
adversarial	44.37 ± 0.94	70.80 ± 0.83	79.63 ±0.71
adversarial*	44.65 ± 0.88	71.48 ± 0.96	78.52 ±0.70
centroid	46.85 ± 0.75	70.37 ± 1.02	**79.98** ±0.80
centroid*	**47.25** ± 0.76	**72.37** ± 0.89	79.46 ±0.72

* without birds (house finch, robin, toucan) in base classes
† our implementation, with early stopping, ‡ implementation from [3]

updates for the classification task. We also explored an adversarial alignment strategy as an alternative. Our experiments demonstrate that our approach, specifically the centroid-based alignment, outperforms previous works in almost all scenarios. The current limitations of our work provide interesting future research directions. First, the alignment approach (sec. 2.6) might include irrelevant examples from the base categories, so using categorical semantic information could help filter out bad samples. An analysis showed that ∼12% of the samples become out-of-distribution (OOD) using a centroid nearest neighbour criteria on *mini*ImageNet in 5-way 1- and 5-shot using ResNet-18. Classification results were not affected significantly by discarding OOD examples at each iteration. Second, the multi-modality of certain base categories look inevitable and might degrade the generalization performance compared to the single-mode case assumed by our centroid alignment strategy. Investigating the use of a mixture family might therefore improve generalization performance. Finally, our algorithms compute the related base once and subsequently keep them fixed during an episode, not taking into account the changes applied to the latent space during the episodic training. Therefore, a more sophisticated dynamic sampling mechanism could be helpful in the finetuning stage.

Chapter 3

Mixture-based Feature Space Learning for Few-shot Image Classification

3.1 Résumé

Nous introduisons "Mixture-based Feature Space Learning" (MixtFSL) pour obtenir une représentation riche et robuste des caractéristiques dans le contexte de la classification d'images avec un petit jeu de données. Les travaux précédents ont proposé de modéliser chaque classe de base soit avec un seul point, soit avec un modèle de mélange en s'appuyant sur des algorithmes de regroupement hors ligne. À la différence de ces études, nous proposons une nouvelle façon de modéliser les classes de base avec des modèles de mélange en réalisant simultanément l'entrainement de l'extracteur de caractéristiques et l'apprentissage des paramètres du modèle de mélange en ligne. Il en résulte un espace de caractéristiques plus riche et plus discriminant qui peut être utilisé pour classer de nouveaux exemples à partir de très peu d'échantillons. Deux étapes principales sont proposées pour former le modèle MixtFSL. Premièrement, les mélanges multimodaux pour chaque classe de base et les paramètres de l'extracteur de caractéristiques sont appris en utilisant une combinaison de deux fonctions de perte. Ensuite, les modèles de réseau et de mélange résultants sont progressivement affinés par une procédure d'apprentissage de type leader-suiveur, qui utilise l'estimation actuelle comme réseau "cible". Ce réseau cible est utilisé pour effectuer une affectation cohérente des instances aux composants du mélange, ce qui augmente les performances et stabilise la formation. L'efficacité de notre approche d'apprentissage de l'espace des caractéristiques de bout en bout est démontrée par des expériences approfondies sur quatre ensembles de données standard et quatre ossatures (backbones). Lorsque nous combinons notre représentation robuste avec des approches récentes basées sur l'alignement, nous obtenons des résultats inédits dans le cadre inductif, avec une précision absolue pour la classification de 5 données de 82,45% sur miniImageNet, 88,20% avec tieredImageNet et 60,70% dans FC100 en utilisant l'ossature ResNet-12.

3.2 Abstract

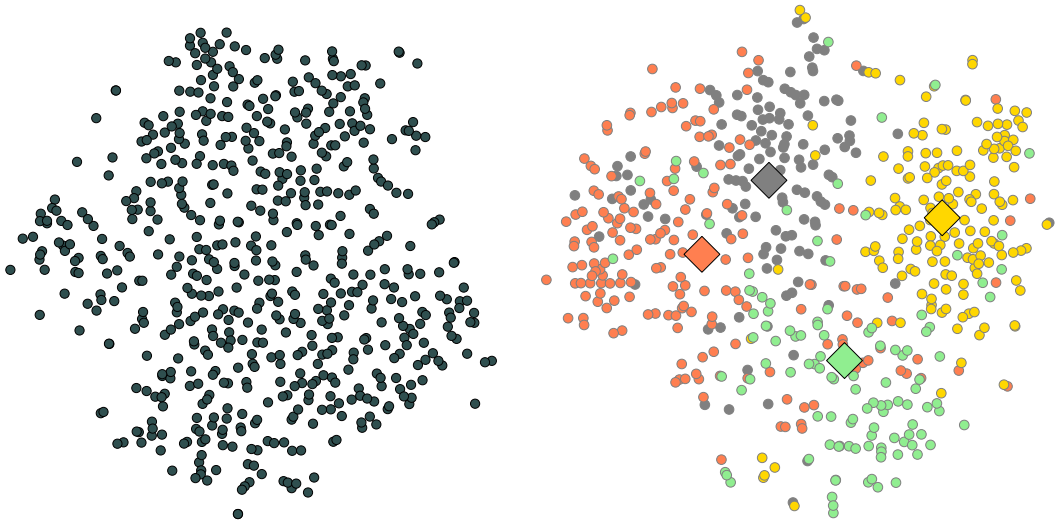
We introduce Mixture-based Feature Space Learning (MixtFSL) for obtaining a rich and robust feature representation in the context of few-shot image classification. Previous works have proposed to model each base class either with a single point or with a mixture model by relying on offline clustering algorithms. In contrast, we propose to model base classes with mixture models by simultaneously training the feature extractor and learning the mixture model parameters in an online manner. This results in a richer and more discriminative feature space which can be employed to classify novel examples from very few samples. Two main stages are proposed to train the MixtFSL model. First, the multimodal mixtures for each base class and the feature extractor parameters are learned using a combination of two loss functions. Second, the resulting network and mixture models are progressively refined through a leader-follower learning procedure, which uses the current estimate as a "target" network. This target network is used to make a consistent assignment of instances to mixture components, which increases performance and stabilizes training. The effectiveness of our end-to-end feature space learning approach is demonstrated with extensive experiments on four standard datasets and four backbones. Notably, we demonstrate that when we combine our robust representation with recent alignment-based approaches, we achieve new state-of-the-art results in the inductive setting, with an absolute accuracy for 5-shot classification of 82.45% on miniImageNet, 88.20% with tieredImageNet, and 60.70% in FC100 using the ResNet-12 backbone.

3.3 Introduction

The goal of few-shot image classification is to transfer knowledge gained on a set of "base" categories, containing a large number of training examples, to a set of distinct "novel" classes having very few examples [84, 85]. A hallmark of successful approaches [7, 9, 10] is their ability to learn rich and robust feature *representations* from base training images, which can generalize to novel samples.

A common assumption in few-shot learning is that classes can be represented with unimodal models. For example, Prototypical Networks [9] ("ProtoNet" henceforth) assumed each base class can be represented with a single prototype. Others, favoring standard transfer learning [3, 4, 6], use a classification layer which push each training sample towards a single vector. While this strategy has successfully been employed in "typical" image classification (e.g., ImageNet challenge [2]), it is somewhat counterbalanced because the learner is regularized by using validation examples that belong to the same training classes. Alas, this solution does not transfer to few-shot classification since the base, validation, and novel classes are disjoint. Indeed, Allen *et al.* [1] showed that relying on that unimodal assumption limits adaptability in few-shot image classification and is prone to underfitting from a data representation perspective.

To alleviate this limitation, Infinite Mixture Prototypes [1] (IMP) extends ProtoNet by representing each class with multiple centroids. This is accomplished by employing an offline clustering (extension of DP-means [19]) where the non-learnable centroids are recomputed at each iteration. This approach



(a) without MixtFSL (b) our MixtFSL

Figure 3.1: t-SNE [63] visualization of a single base class embedding (circles) (a) without, and (b) with our MixtFSL approach. MixtFSL learns a representation for base samples (circles) and associated mixture learned components (diamonds) that clusters a class into several modes (different colors). This more flexible representation helps in training robust classifiers from few samples in the novel domain compared to the monolithic representation of (a). Embeddings are extracted from a miniImageNet class using a ResNet-18.

however suffers from two main downsides. First, it does not allow capturing the global distribution of base classes since a small subset of the base samples are clustered at any one time—clustering over all base samples at each training iteration would be prohibitively expensive. Second, relying on DP-means in an offline, post hoc manner implies that feature learning and clustering are done independently.

In this paper, we propose "Mixture-based Feature Space Learning" (MixtFSL) to learn a multimodal representation for the base classes using a mixture of trainable components—learned vectors that are iteratively refined during training. The key idea is to learn both the *representation* (feature space) and the *mixture model* jointly in an online manner, which effectively unites these two tasks by allowing the gradient to flow between them. This results in a discriminative representation, which in turn yields superior performance when training on the novel classes from few examples.

We propose a two-stage approach to train our MixtFSL. In the first stage, the mixture components are initialized by the combination of two losses that ensure that: 1) samples are assigned to their nearest mixture component; while 2) enforcing components of a same class mixture to be far enough from each other, to prevent them from collapsing to a single point. In the second stage, the learnable mixture model is progressively refined through a leader-follower scheme, which uses the current estimate of the learner as a fixed "target" network, updated only on a few occasions during that phase, and a progressively declining temperature strategy. Our experiments demonstrate that this improves performance and stabilizes the training. During training, the number of components in the learned

mixture model is automatically adjusted from data. The resulting representation is flexible and better adapts to the multi-modal nature of images (fig. 3.1), which results in improved performance on the novel classes.

Our contributions are as follows. We introduce the idea of MixtFSL for few-shot image classification, which learns a flexible representation by modeling base classes as a mixture of learnable components. We present a robust two-stage scheme for training such a model. The training is done end-to-end in a fully differentiable fashion, without the need for an offline clustering method. We demonstrate, through an extensive experiments on four standard datasets and using four backbones, that our MixtFSL outperforms the state of the art in most of the cases tested. We show that our approach is flexible and can leverage other improvements in the literature (we experiment with associative alignment [6] and ODE [12]) to further boost performance. Finally, we show that our approach does not suffer from forgetting (the base classes).

3.4 Related work

Few-shot learning is now applied to problems such as image-to-image translation [86], object detection [87, 88], video classification [89], and 3D shape segmentation [90]. This paper instead focuses on the image classification problem [7, 9, 10], so the remainder of the discussion will focus on relevant works in this area. In addition, unlike transductive inference methods [69, 91, 92, 93, 94, 95, 96, 97] which uses the structural information of the entire novel set, our research focuses on inductive inference research area.

Meta learning In meta learning [7, 8, 9, 25, 69, 68, 67, 70, 98, 99], approaches imitate the few-shot scenario by repeatedly sampling similar scenarios (episodes) from the base classes during the pre-training phase. Here, distance-based approaches [9, 10, 13, 17, 26, 27, 28, 29, 30, 31, 32, 33, 71] aim at transferring the reduced intra-class variation from base to novel classes, while initialization-based approaches [7, 23, 24] are designed to carry the best starting model configuration for novel class training. Our MixtFSL benefits from the best of both worlds, by reducing the within-class distance with the learnable mixture component and increasing the adaptivity of the network obtained after initial training by representing each class with mixture components.

Standard transfer learning Batch form training makes use of a standard transfer learning *modus operandi* instead of episodic training. Although batch learning with a naive optimization criteria is prone to overfitting, several recent studies [3, 4, 5, 6, 22] have shown a metric-learning (margin-based) criteria can offer good performance. For example, Bin et al. [100] present a negative margin based feature space learning. Our proposed MixtFSL also uses transfer learning but innovates by simultaneously clustering base class features into multi-modal mixtures in an online manner.

Data augmentation Data augmentation [16, 37, 38, 39, 40, 41, 42, 44, 45, 46, 47, 48, 49, 50] for few-shot image classification aims at training a well-generalized algorithm. Here, the data can be

augmented using a generator function. For example, [38] proposed Feature Hallucination (FH) using an auxiliary generator. Later, [46] extends FH to generate new data using generative models. In contrast, our MixtFSL does not generate any data and achieves state-of-the-art. [6] makes use of "related base" samples in combination with an alignment technique to improve performance. We demonstrate (in sec. 3.8) that we can leverage this approach in our framework since our contribution is orthogonal.

Mixture modeling Similar to classical mixture-based works [101, 102] outside few-shot learning, infinite mixture model [56] explores Bayesian methods [57, 58] to infer the number of mixture components. Recently, IMP [1] relies on the DP-means [19] algorithm which is computed inside the episodic training loop in few-shot learning context. As in [56], our MixtFSL automatically learns the number of mixture components, but differs from [1] by learning the mixture model simultaneously with representation learning in an online manner, without the need for a separate, post hoc clustering algorithm. From the learnable component perspective, our MixtFSL is related to VQ-VAE [52, 53] which learns quantized feature vectors for image generation, and SwAV [54] for self-supervised learning. Here, we tackle supervised few-shot learning by using mixture modeling to increase the adaptivity of the learned representation. This also contrasts with variational few-shot learning [27, 33], which aims to reduce noise with variational estimates of the distribution. Our MixtFSL is also related to MM-Net [103] in that they both works store information during training. Unlike MM-Net, which contains read/write controllers plus a contextual learner to build an attention-based inference, our MixtFSL aims at modeling the multi-modality of the base classes with only a set of learned components.

3.5 Problem definition

In few-shot image classification, we assume there exists a "base" set $\mathcal{X}^b = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N^b}$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathcal{Y}^b$ are respectively the i-th input image and its corresponding class label. There is also a "novel" set $\mathcal{X}^n = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N^n}$, where $y_i \in \mathcal{Y}^n$, and a "validation" set $\mathcal{X}^v = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N^v}$, where $y_i \in \mathcal{Y}^v$. None of these sets overlap and $N^n \ll N^b$.

In this paper, we follow the standard transfer learning training strategy (as in, for example, [6, 3]). A network $\mathbf{z} = f(\mathbf{x}|\theta)$, parameterized by θ, is pre-trained to project input image $\mathbf{x}$ to a feature vector $\mathbf{z} \in \mathbb{R}^M$ using the base categories $\mathcal{X}^b$, validated on $\mathcal{X}^v$. The key idea behind our proposed MixtFSL model is to simultaneously train a learnable mixture model, along with $f(\cdot|\theta)$, in order to capture the distribution of each base class in $\mathcal{X}^b$. This mixture is guiding the representation learning for a better handling of multimodal class distributions, while allowing to extract information on the base class components that can be useful to stabilize the training. We denote the mixture model across all base classes as the set $\boldsymbol{\mathcal{P}} = \{(\mathcal{P}_k, y_k)\}_{k=1}^{N^b}$, where each $\mathcal{P}_k = \{\mathbf{u}_j\}_{j=1}^{N^k}$ is the set of all N^k components $\mathbf{u}_j \in \mathbb{R}^M$ assigned to the k-th base class. After training on the base categories, fine-tuning the classifier on the *novel* samples is very simple and follows [3]: the weights θ are fixed, and a single linear classification layer $\mathbf{W}$ is trained as in $c(\cdot|\mathbf{W}) \equiv \mathbf{W}^\top f(\cdot|\theta)$, followed by softmax. The key observation is that the mixture model, trained only on the base classes, makes the learned feature space more

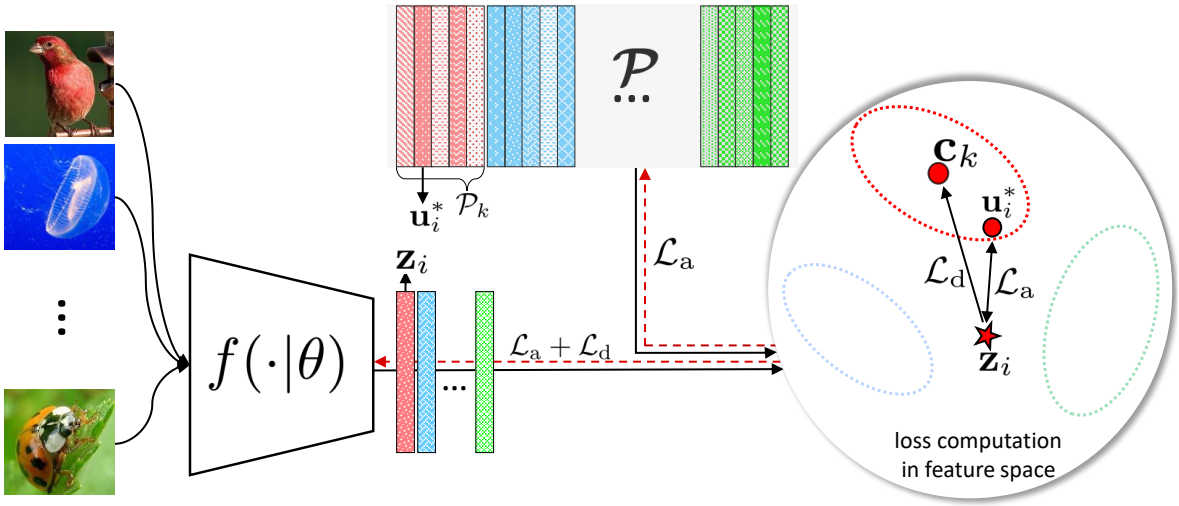


Figure 3.2: Initial training stage. The network $f(\cdot|\theta)$ embeds a batch (left) from the base classes to feature space. A feature vector $\mathbf{z}_i$ (middle) belonging to the k-th class is assigned to the most similar component $\mathbf{u}_i^*$ in class mixture $\mathcal{P}_k \in \boldsymbol{\mathcal{P}}$. Vectors are color-coded by class. Here, two losses interact for representation learning: $\mathcal{L}_\mathrm{a}$ which maximizes the similarity between $\mathbf{z}_i$ and $\mathbf{u}_i^*$; and $\mathcal{L}_\mathrm{d}$ keeps $\mathbf{z}_i$ close to the centroid $\mathbf{c}_k$ of all mixture components for class k. The backpropagated gradient is shown with red dashed lines. While $f(\cdot|\theta)$ is updated by $\mathcal{L}_\mathrm{it}$ (eq. 3.5), $\boldsymbol{\mathcal{P}}$ is updated by $\mathcal{L}_\mathrm{a}$ only to prevent collapsing of the components in $\mathcal{P}_k$ to a single point.

discriminative—only a simple classification layer can thus be trained on the novel classes.

3.6 Mixture-based Feature Space Learning

Training our MixtFSL on the base classes is done in two main stages: initial training and progressive following.

3.6.1 Initial training

The initial training of the feature extractor $f(\cdot|\theta)$ and the learnable mixture model $\boldsymbol{\mathcal{P}}$ from the base class set $\mathcal{X}^b$ is detailed in algorithm 3 and illustrated in fig. 3.2. In this stage, model parameters are updated using two losses: the "assignment" loss $\mathcal{L}_\mathrm{a}$, which updates both the feature extractor and the mixture model such that feature vectors are assigned to their nearest mixture component; and the "diversity" loss $\mathcal{L}_\mathrm{d}$, which updates the feature extractor to diversify the selection of components for a given class. Let us define the following angular margin-based softmax function [15], modified with a temperature variable τ:

$$p_\theta(v_j|\mathbf{z}_i, \boldsymbol{\mathcal{P}}) = \frac{e^{\cos((\angle(\mathbf{z}_i,\mathbf{u}_j)+m))/\tau}}{e^{\cos((\angle(\mathbf{z}_i,\mathbf{u}_j)+m))/\tau} + \sum\limits_{\mathbf{u}_l \in \{\boldsymbol{\mathcal{P}} \setminus \mathbf{u}_j\}} e^{\cos(\angle(\mathbf{z}_i,\mathbf{u}_l))/\tau}}, \tag{3.1}$$

Algorithm 3: Initial training.

Data: feature extractor $f(\cdot|\theta)$, mixture $\boldsymbol{\mathcal{P}}$, base dataset $\mathcal{X}^b$, validation dataset $\mathcal{X}^v$, maximum epoch α_0, patience α_1, and error evaluation function $E(\cdot)$
Result: Model $f(\cdot|\theta^{\text{best}})$ and mixture $\boldsymbol{\mathcal{P}}^{\text{best}}$ learned
$\theta^{\text{best}} \leftarrow \theta$; $\boldsymbol{\mathcal{P}}^{\text{best}} \leftarrow \boldsymbol{\mathcal{P}}$; $t \leftarrow 0$; $s \leftarrow 0$
while $s < \alpha_0$ *and* $t < \alpha_1$ **do**
  for $(\mathbf{x}_i, y_i) \in \mathcal{X}^b$ **do**
    Evaluate $\mathbf{z}_i \leftarrow f(\mathbf{x}_i|\theta)$, and $\mathbf{u}_i^*$ by eq. 3.2
    Update weights θ and mixture $\boldsymbol{\mathcal{P}}$ with $\mathcal{L}_{\text{it}}$ (eq. 3.5);
  end
  Evaluate $f(\cdot|\theta)$ on $\mathcal{X}^v$ with episodic training
  if $E(\theta, \boldsymbol{\mathcal{P}}|\mathcal{X}^v) < E(\theta^{\text{best}}, \boldsymbol{\mathcal{P}}^{\text{best}}|\mathcal{X}^v)$ **then**
    $\theta^{\text{best}} \leftarrow \theta$; $\boldsymbol{\mathcal{P}}^{\text{best}} \leftarrow \boldsymbol{\mathcal{P}}$; $t \leftarrow 0$
  else
    $t \leftarrow t + 1$
  end
  $s \leftarrow s + 1$
end

where, m is a margin; v_j is the pseudo-label associated to $\mathbf{u}_j$; and, $\angle(\mathbf{z}_i, \mathbf{u}_j) = \arccos\left(\mathbf{z}_i^\top \mathbf{u}_i / (||\mathbf{z}_i|| ||\mathbf{u}_j||)\right)$[1].

Given a training image $\mathbf{x}_i$ from base class $y_i = k$ and its associated feature vector $\mathbf{z}_i = f(\mathbf{x}_i|\theta)$, the closest component $\mathbf{u}_i^*$ is found amongst all elements of mixture $\mathcal{P}_k$ associated to the same class according to cosine similarity:

$$\mathbf{u}_i^* = \underset{\mathbf{u}_j \in \mathcal{P}_k}{\arg\max} \frac{\mathbf{z}_i \cdot \mathbf{u}_j}{\|\mathbf{z}_i\| \|\mathbf{u}_j\|}\,, \tag{3.2}$$

where $\cdot$ denotes the dot product. Based on this, the "assignment" loss function $\mathcal{L}_{\text{a}}$ updates both $f(\cdot|\theta)$ and $\boldsymbol{\mathcal{P}}$ such that $\mathbf{z}_i$ is assigned to its most similar component $\mathbf{u}_i^*$:

$$\mathcal{L}_{\text{a}} = -\frac{1}{N} \sum_{i=1}^{N} \log p_\theta(v_i^*|\mathbf{z}_i, \boldsymbol{\mathcal{P}})\,, \tag{3.3}$$

where N is the batch size and v_i^* is the one-hot pseudo-label corresponding to $\mathbf{u}_i^*$. The gradient of eq. 3.3 is back-propagated to both $f(\cdot|\theta)$ and the learned components $\boldsymbol{\mathcal{P}}$.

As verified later (sec. 3.7.3), training solely on the assignment loss $\mathcal{L}_{\text{a}}$ generally results in a single component $\mathbf{u}_i \in \mathcal{P}_k$ to be assigned to all training instances for class k, thereby effectively degrading the learned mixtures to a single mode. We compensate for this by adding a second loss function to encourage a diversity of components to be selected by enforcing $f(\cdot|\theta)$ to push the $\mathbf{z}_i$ values towards the centroid of the components corresponding to their associated labels y_i. For the centroid $\boldsymbol{c}_k = (1/|\mathcal{P}_k|) \sum_{\mathbf{u}_j \in \mathcal{P}_k} \mathbf{u}_j$ for base class k, and the set $\boldsymbol{\mathcal{C}} = \{\boldsymbol{c}_k\}_{k=1}^{N^b}$ of the centroids for base classes, we define the *diversity* loss as:

$$\mathcal{L}_{\text{d}} = -\frac{1}{N} \sum_{i=1}^{N} \log p_\theta(y_i|\mathbf{z}_i, \text{sg}[\boldsymbol{\mathcal{C}}])\,, \tag{3.4}$$

[1] As per [15], we avoid computing the arccos (which is undefined outside the $[-1, 1]$ interval) and directly compute the $\cos(\angle(\mathbf{z}_i, \mathbf{u}_j) + m)$.

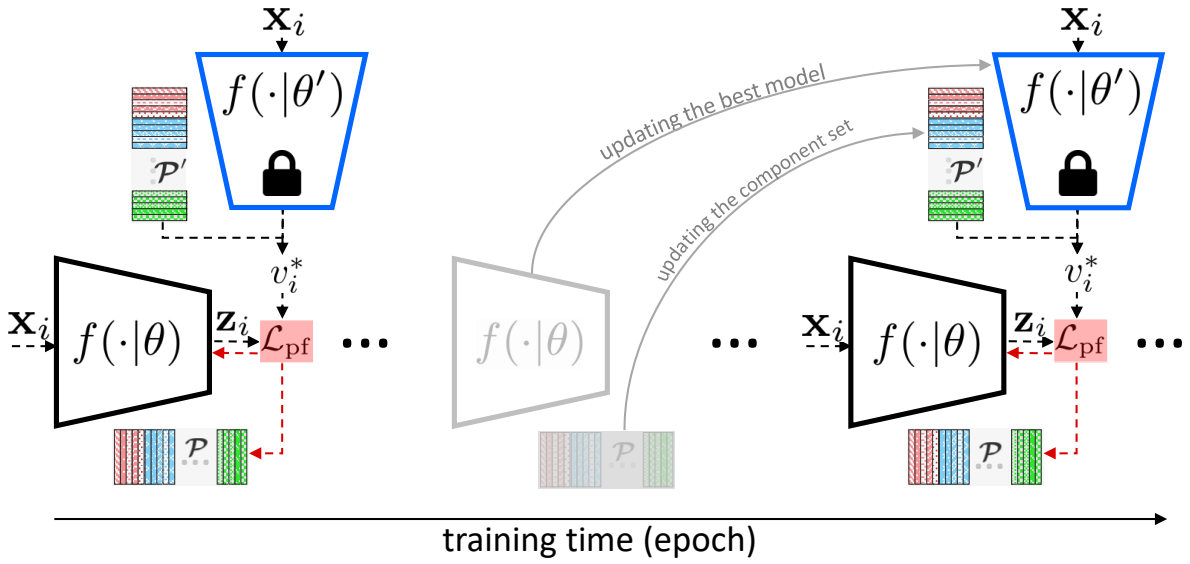


Figure 3.3: Progressive following training stage. $f(\cdot|\theta)$ is adapted using loss function $\mathcal{L}_{\text{pf}}$ (eq. 3.7) and supervised by a fixed copy of the best target model $f(\cdot|\theta')$ (in blue) and the corresponding mixture $\mathcal{P}'$ after the initial training stage. The gradient (dashed red line) is backpropagated only through $f(\cdot|\theta)$ and $\mathcal{P}$, while $f(\cdot|\theta')$ and $\mathcal{P}'$ are kept fixed. The target network and mixture $f(\cdot|\theta')$ and $\mathcal{P}'$ are replaced by the best validated $f(\cdot|\theta)$ and $\mathcal{P}$ after α_3 number of training steps with no improvement in validation. The temperature factor τ (eq. 3.1) decreases each time the target network is updated to create progressively more discriminative clusters.

where sg stands for stopgradient, which blocks backpropagation over the variables it protects. The sg operator in eq. 3.4 prevents the collapsing of all components of the k-th class $\mathcal{P}_k$ into a single point. Overall, the loss in this initial stage is the combination of eqs 3.3 and 3.4:

$$\mathcal{L}_{\text{it}} = \mathcal{L}_{\text{a}} + \mathcal{L}_{\text{d}} \,. \tag{3.5}$$

3.6.2 Progressive following

After the initial training of the feature extractor $f(\cdot|\theta)$ and mixture $\mathcal{P}$, an intense competition is likely to arise for the assignment of the nearest components to each instance $\mathbf{z}_i$. To illustrate this, suppose $\dot{\mathbf{u}}$ is assigned to $\mathbf{z}$ at iteration t. At the following iteration $t+1$, the simultaneous weight update to both $f(\cdot|\theta)$ and $\mathcal{P}$ could cause another $\ddot{\mathbf{u}}$, in the vicinity of $\dot{\mathbf{u}}$ and $\mathbf{z}$, to be assigned as the nearest component of $\mathbf{z}$. Given the margin-based softmax (eq. 3.1), $\mathbf{z}$ is pulled toward $\dot{\mathbf{u}}$ and pushed away from $\ddot{\mathbf{u}}$ at iteration t, and contradictorily steered in the opposite direction at the following iteration. As a result, this "pull-push" behavior stalls the improvement of $f(\cdot|\theta)$, preventing it from making further progress.

To tackle this problem, we propose a progressive following stage that aim to break the complex dynamic of simultaneously determining nearest components while training the representation $f(\cdot|\theta)$ and mixture $\mathcal{P}$. The approach is detailed in algorithm 4 and shown in fig. 3.3. Using the "prime" notation (θ' and $\mathcal{P}'$ to specify the best feature extractor parameters and mixture component so far, resp.), the approach

Algorithm 4: Progressive following.

Data: pre-trained $f(\cdot|\theta)$, pre-trained $\boldsymbol{\mathcal{P}}$, base set $\mathcal{X}^b$, validation set $\mathcal{X}^v$, patience α_2, number of repetitions α_3, temperature τ, decreasing ratio γ, and error evaluation function $E(\cdot)$
Result: Refined model $f(\cdot|\theta^{\text{best}})$ and mixture $\boldsymbol{\mathcal{P}}^{\text{best}}$
$\theta' \leftarrow \theta$; $\boldsymbol{\mathcal{P}}' \leftarrow \boldsymbol{\mathcal{P}}$; $\theta^{\text{best}} \leftarrow \theta$; $\boldsymbol{\mathcal{P}}^{\text{best}} \leftarrow \boldsymbol{\mathcal{P}}$; $s \leftarrow 0$
for $t = 1, 2, \ldots, \alpha_3$ **do**
  while $s < \alpha_2$ **do**
    for $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X}^b$ **do**
      Evaluate $\mathbf{z}_i \leftarrow f(\mathbf{x}_i|\theta')$, and $\mathbf{u}_i^{*\prime}$ by eq. 3.6
      Update weights θ and mixture $\boldsymbol{\mathcal{P}}$ by backward error propagation from $\mathcal{L}_{\text{pf}}$ (eq. 3.7)
    end
    if $E(\theta, \boldsymbol{\mathcal{P}}|\mathcal{X}^v) < E(\theta^{\text{best}}, \boldsymbol{\mathcal{P}}^{\text{best}}|\mathcal{X}^v)$ **then**
      $\theta^{\text{best}} \leftarrow \theta$; $\boldsymbol{\mathcal{P}}^{\text{best}} \leftarrow \boldsymbol{\mathcal{P}}$; $s \leftarrow 0$
    else
      $s \leftarrow s + 1$
    end
  end
  Update target $\theta' \leftarrow \theta^{\text{best}}$ and mixture $\boldsymbol{\mathcal{P}}' \leftarrow \boldsymbol{\mathcal{P}}^{\text{best}}$
  Decrease temperature τ of eq. 3.1 as $\tau \leftarrow \gamma\tau$
end

starts by taking a copy of $f(\cdot|\theta')$ and $\boldsymbol{\mathcal{P}}'$, and by using them to determine the nearest component of each training instance:

$$\mathbf{u}_i^{*\prime} = \underset{\mathbf{u}'_j \in \mathcal{P}'_k}{\arg\max} \frac{\mathbf{z}'_i \cdot \mathbf{u}'_j}{\|\mathbf{z}'_i\| \|\mathbf{u}'_j\|} \,, \tag{3.6}$$

where $\mathbf{z}'_i = f(\mathbf{x}_i|\theta')$. Since determining the labels does not depend on the learned parameters θ anymore, consistency in the assignment of nearest components is preserved, and the "push-pull" problem mentioned above is eliminated.

Since label assignments are fixed, the diversity loss (eq. 3.4) is not needed anymore. Therefore, we can reformulate the progressive assignment loss function as:

$$\mathcal{L}_{\text{pf}} = -\frac{1}{N} \sum_{i=1}^{N} \log p_\theta(v_i^{*\prime}|\mathbf{z}_i, \boldsymbol{\mathcal{P}}) \,, \tag{3.7}$$

where N is the batch size and $v_i^{*\prime}$ the pseudo-label associated to the nearest component $\mathbf{u}_i^{*\prime}$ found by eq. 3.6.

After α_2 updates to the representation with no decrease of the validation set error (function $E(\cdot)$ in algorithms 3 and 4), the best network $f(\cdot|\theta')$ and mixture $\boldsymbol{\mathcal{P}}'$ are then replaced with the new best ones found on validation set, the temperature τ is decreased by a factor $\gamma < 1$ to push the $\mathbf{z}$ more steeply towards their closest mixture component, and the entire procedure is repeated as shown in algorithm 4. After a maximum number of α_3 iterations is reached, the global best possible model θ^{best} and mixture $\mathcal{P}^{\text{best}}$ are obtained. Components that have no base class samples associated (i.e. never selected by eq. 3.6) are simply discarded. This effectively adapts the mixture models to each base class distribution.

In summary, the progressive following aims at solving the discussed pull-push behavior observed (see sec. 3.7.3). This stage applies a similar approach than in initial stage, with two significant differences: 1) the diversity loss $\mathcal{L}_{\text{d}}$ is removed; and 2) label assignments are provided by a copy of the best model so far $f(\cdot|\theta')$ to stabilize the training.

3.7 Experimental validation

The following section presents the experimental validations of our novel mixture-based feature space learning (MixtFSL). We begin by introducing the datasets, backbones and implementation details. We then present experiments on object recognition, fine-grained and cross-domain classification. Finally, an ablative analysis is presented to evaluate the impact of decisions made in the design of MixtFSL.

3.7.1 Datasets and implementation details

Datasets Object recognition is evaluated using the miniImageNet [10] and tieredImageNet [16], which are subsets of the ILSVRC-12 dataset [2]. miniImageNet contains 64/16/20 base/validation/novel classes respectively with 600 examples per class, and tieredImageNet [16] contains 351/97/160 base/validation/novel classes. For fine-grained classification, we employ CUB-200-2011 (CUB) [18] which contains 100/50/50 base/validation/novel classes. For cross-domain, we train on the base and validation classes of miniImageNet, and evaluate on the novel classes of CUB.

Backbones and implementation details We conduct experiments using four different backbones: 1) Conv4, 2) ResNet-18 [79], 3) ResNet-12 [79], and 4) 28-layer Wide-ResNet ("WRN") [80]. We used Adam [17] and SGD with a learning rate of 10^{-3} to train Conv4 and ResNets and WRN, respectively. In SGD case, we used Nesterov with an initial rate of 0.001, and the weight decay is fixed as 5e-4 and momentum as 0.9. In all cases, batch size is fixed to 128. The starting temperature variable τ and margin m (eq. 3.1 in sec. 3.6) were found using the validation set (see supp. material). Components in $\mathcal{P}$ are initialized with Xavier uniform [104] (gain $= 1$), and their number $N^k = 15$ (sec. 3.5), except for tieredImageNet where $N^k = 5$ since there is a much larger number of bases classes (351). A temperature factor of $\gamma = 0.8$ is used in the progressive following stage. The early stopping thresholds of algorithms 3 and 4 are set to $\alpha_0 = 400$, $\alpha_1 = 20$, $\alpha_2 = 15$ and $\alpha_3 = 3$.

3.7.2 Mixture-based feature space evaluations

We first evaluate our proposed MixtFSL model on all four datasets using a variety of backbones.

miniImageNet Table 3.1 compares our MixtFSL with several recent method on miniImageNet, with four backbones. MixtFSL provides accuracy improvements in all but three cases. In the most of these exceptions, the method with best accuracy is Neg-Margin [100], which is explored in more details in sec. 3.7.3. Of note, MixtFSL outperforms IMP [1] (sec. 3.3 and 3.4) by 3.22% and 2.57% on 1- and

Table 3.1: Evaluation on miniImageNet in 5-way. Bold/blue is best/second, and ± is the 95% confidence intervals in 600 episodes.

Method	Backbone	1-shot	5-shot
ProtoNet [9]	Conv4	49.42 ± 0.78	68.20 ± 0.66
MAML [23]	Conv4	48.07 ± 1.75	63.15 ± 0.91
RelationNet [30]	Conv4	50.44 ± 0.82	65.32 ± 0.70
Baseline++ [3]	Conv4	48.24 ± 0.75	66.43 ± 0.63
IMP [1]	Conv4	49.60 ± 0.80	68.10 ± 0.80
MemoryNetwork [103]	Conv4	**53.37** ± 0.48	66.97 ± 0.35
Arcmax [6]	Conv4	51.90 ±0.79	69.07 ± 0.59
Neg-Margin [100]	Conv4	52.84 ±0.76	70.41 ±0.66
MixtFSL (ours)	Conv4	52.82 ±0.63	**70.67** ±0.57
DNS [105]	RN-12	62.64 ±0.66	78.83 ±0.45
Var.FSL [33]	RN-12	61.23 ±0.26	77.69 ±0.17
MTL [83]	RN-12	61.20 ±1.80	75.50 ±0.80
SNAIL [81]	RN-12	55.71 ±0.99	68.88 ±0.92
AdaResNet [106]	RN-12	56.88 ±0.62	71.94 ±0.57
TADAM [17]	RN-12	58.50 ±0.30	76.70 ±0.30
MetaOptNet [73]	RN-12	62.64 ±0.61	78.63 ±0.46
Simple [22]	RN-12	62.02 ±0.63	79.64 ±0.44
TapNet [70]	RN-12	61.65 ±0.15	76.36 ±0.10
Neg-Margin [100]	RN-12	63.85 ±0.76	81.57 ±0.56
MixtFSL (ours)	RN-12	**63.98** ±0.79	**82.04** ±0.49
MAML‡ [7]	RN-18	49.61 ±0.92	65.72 ±0.77
RelationNet‡ [30]	RN-18	52.48 ±0.86	69.83 ±0.68
MatchingNet‡ [10]	RN-18	52.91 ±0.88	68.88 ±0.69
ProtoNet‡ [9]	RN-18	54.16 ±0.82	73.68 ±0.65
Arcmax [6]	RN-18	58.70 ±0.82	77.72 ±0.51
Neg-Margin [100]	RN-18	59.02 ±0.81	**78.80** ±0.54
MixtFSL (ours)	RN-18	**60.11** ±0.73	77.76 ±0.58
Act. to Param. [82]	RN-50	59.60 ±0.41	73.74 ±0.19
SIB-inductive§[107]	WRN	60.12	78.17
SIB+IFSL [108]	WRN	63.14 ±3.02	80.05 ±1.88
LEO [25]	WRN	61.76 ±0.08	77.59 ±0.12
wDAE [43]	WRN	61.07 ±0.15	76.75 ±0.11
CC+rot [42]	WRN	62.93 ±0.45	79.87 ±0.33
Robust dist++ [74]	WRN	63.28 ±0.62	81.17 ±0.43
Arcmax [6]	WRN	62.68 ±0.76	80.54 ±0.50
Neg-Margin [100]	WRN	61.72 ±0.90	**81.79** ±0.49
MixtFSL (ours)	WRN	**64.31** ±0.79	81.66 ±0.60

‡taken from [3] §confidence interval not provided

Table 3.2: Evaluation on tieredImageNet and FC100 in 5-way classification. Bold/blue is best/second best, and ± indicates the 95% confidence intervals over 600 episodes.

	Method	Backbone	1-shot	5-shot
tieredImageNet	DNS [105]	RN-12	66.22 ±0.75	82.79 ±0.48
	MetaOptNet [73]	RN-12	65.99 ±0.72	81.56 ±0.53
	Simple [22]	RN-12	69.74 ±0.72	84.41 ±0.55
	TapNet [70]	RN-12	63.08 ±0.15	80.26 ±0.12
	Arcmax* [6]	RN-12	68.02 ±0.61	83.99 ±0.62
	MixtFSL (ours)	RN-12	**70.97** ±1.03	**86.16** ±0.67
	Arcmax [6]	RN-18	65.08 ±0.19	83.67 ±0.51
	ProtoNet [9]	RN-18	61.23 ±0.77	80.00 ±0.55
	MixtFSL (ours)	RN-18	**68.61** ±0.91	**84.08** ±0.55
FC100	TADAM [17]	RN-12	40.1 ± 0.40	56.1 ± 0.40
	MetaOptNet [73]	RN-12	41.1 ± 0.60	55.5 ± 0.60
	ProtoNet† [9]	RN-12	37.5 ± 0.60	52.5 ± 0.60
	MTL [83]	RN-12	43.6 ± 1.80	55.4 ± 0.90
	MixtFSL (ours)	RN-12	**44.89** ±0.63	**60.70** ±0.67
	Arcmax [6]	RN-18	40.84 ± 0.71	57.02 ± 0.63
	MixtFSL (ours)	RN-18	**41.50** ±0.67	**58.39** ±0.62

*our implementation †taken from [73]

5-shot respectively, thereby validating the impact of jointly learning the feature representation together with the mixture model.

tieredImageNet and FC100 Table 3.2 presents similar comparisons, this time on tieredImageNet and FC100. On both datasets and in both 1- and 5-shot scenarios, our method yields state-of-the-art results. In particular, MixtFSL results in classification gains of 3.53% over Arcmax [6] in 1-shot using RN-18, and 1.75% over Simple [22] in 5-shot using ResNet-12 for tieredImageNet, and 1.29% and 4.60% over MTL [83] for FC100 in 1- and 5-shot, respectively.

CUB Table 3.3 evaluates our approach on CUB, both for fine-grained classification in 1- and 5-shot, and in cross-domain from miniImageNet to CUB for 5-shot using the ResNet-18. Here, previous work [100] outperforms MixtFSL in the 5-shot scenario. We hypothesize this is due to the fact that either CUB classes are more unimodal than miniImageNet or that less examples per-class are in the dataset, which could be mitigated with self-supervised methods.

3.7.3 Ablative analysis

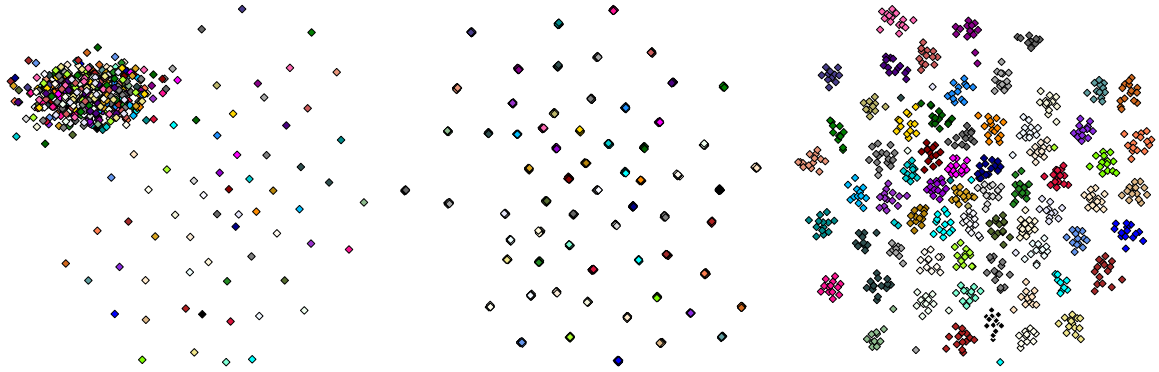
Here, we perform ablative experiments to evaluate the impact of two design decisions in our approach.

Initial training vs progressive following Fig. 3.4 illustrates the impact of loss functions qualitatively.

Table 3.3: Fine-grained and cross-domain from miniImageNet to CUB evaluation in 5-way using ResNet-18. Bold/blue is best/second, and ± is the 95% confidence intervals on 600 episodes.

	CUB		miniIN→CUB
Method	**1-shot**	**5-shot**	**5-shot**
GNN-LFT$^\diamond$ [31]	51.51 ±0.8	73.11 ±0.7	–
Robust-20 [74]	58.67 ±0.7	75.62 ±0.5	–
RelationNet‡ [30]	67.59 ±1.0	82.75 ±0.6	57.71 ±0.7
MAML‡ [7]	68.42 ±1.0	83.47 ±0.6	51.34 ±0.7
ProtoNet‡ [9]	71.88 ±0.9	86.64 ±0.5	62.02 ±0.7
Baseline++ [3]	67.02 ±0.9	83.58 ±0.5	64.38 ±0.9
Arcmax [6]	71.37 ±0.9	85.74 ±0.5	64.93 ±1.0
Neg-Margin [100]	72.66 ±0.9	**89.40** ±0.4	67.03 ±0.8
MixtFSL (ours)	**73.94** ±1.1	86.01 ±0.5	**68.77** ±0.9

‡taken from [108] $^\diamond$backbone is ResNet-10

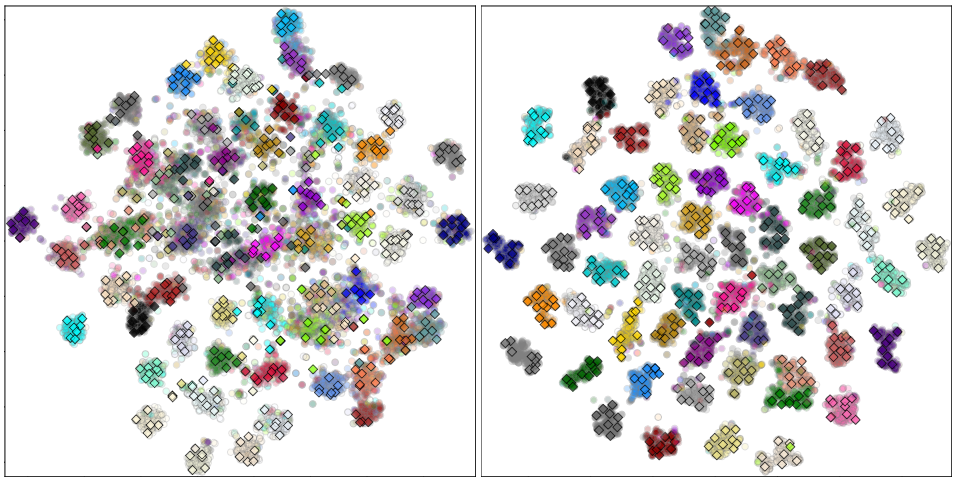


(a) without $\mathcal{L}_d$ (b) $\mathcal{L}_d$ without sg (c) $\mathcal{L}_d$ with sg

Figure 3.4: t-SNE of mixture components (RN-12, miniImageNet).

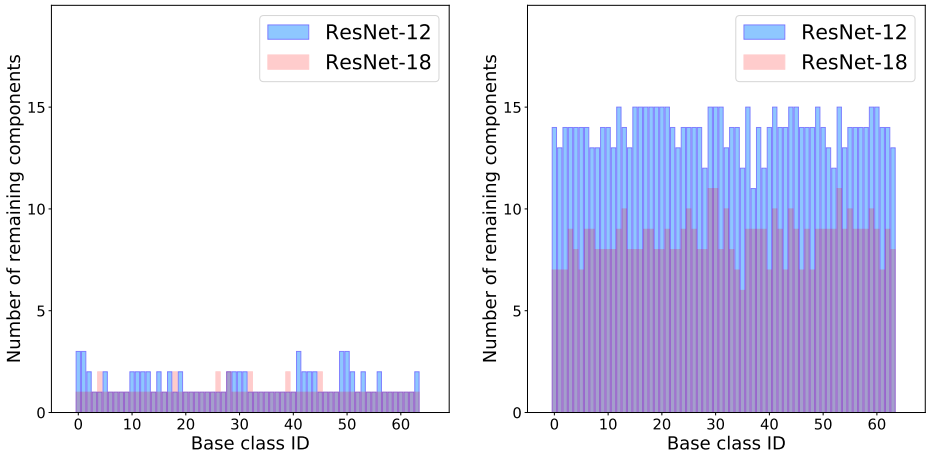
Using only $\mathcal{L}_\text{a}$ causes a single component to dominate while the others are pushed far away (big clump in fig. 3.4a) and is equivalent to the baseline (table 3.4, rows 1–2). Adding $\mathcal{L}_\text{d}$ *without* the sg operator minimizes the distance between the $\mathbf{z}_i$'s to the centroids, resulting in the collapse of all components in $\mathcal{P}_k$ into a single point (fig. 3.4b). sg prevents the components (through their centroids) from being updated (fig. 3.4c), which results in improved performance in the novel domain (t. 3.4, row 3). Finally, $\mathcal{L}_\text{pf}$ further improves performance while bringing stability to the training (t. 3.4, row 4). Beside, Fig. 3.5 presents a t-SNE [63] visualization of base examples and their associated mixture components. Compared to initial training, the network at the end of progressive following stage results in an informative feature space with the separated base classes.

Diversity loss $\mathcal{L}_\text{d}$ Fig. 3.6 presents the impact of our diversity loss $\mathcal{L}_\text{d}$ (eq. 3.4) by showing the number of remaining components after optimization (recall from sec. 3.6.2 that components assigned to no base sample are discarded after training). Without $\mathcal{L}_\text{d}$ (fig. 3.6a), most classes are represented by a single component. Activating $\mathcal{L}_\text{d}$ results in a large number of components having non-zero base



(a) after initial training (b) after progressive following

Figure 3.5: t-SNE [63] visualization of the learned feature embedding (circles) and mixture components (diamonds), after the (a) initial training and (b) progressive following stages. Results are obtained with the ResNet-12 and points are color-coded by base class.



(a) without $\mathcal{L}_{\mathrm{d}}$ (b) with $\mathcal{L}_{\mathrm{d}}$

Figure 3.6: Number of remaining components after training for each of the miniImageNet base classes (a) without and (b) with the diversity loss $\mathcal{L}_{\mathrm{d}}$ (eq. 3.4) using ResNet-12 and ResNet-18. The loss is critical to model the multimodality of base classes.

samples, thereby results in the desired mixture modeling (fig. 3.6b).

Margin in eq. 3.1 As in [6] and [100], our loss function (eq. 3.1) uses a margin-based softmax function modulated by a temperature variable τ. In particular, [100] suggested that a negative margin $m < 0$ improves accuracy. Here, we evaluate the impact of the margin m, and demonstrate in table 3.5 that MixtFSL does not appear to be significantly affected by its sign.

Table 3.4: Validation set accuracy of miniImageNet on 150 epochs.

Method	RN-12		RN-18	
	1-shot	5-shot	1-shot	5-shot
Baseline	56.55	72.68	55.38	72.81
Only $\mathcal{L}_{\mathrm{a}}$	56.52	72.78	55.55	72.67
Init. tr. ($\mathcal{L}_{\mathrm{a}} + \mathcal{L}_{\mathrm{d}}$)	57.88	73.94	56.18	69.43
Prog. fol. ($\mathcal{L}_{\mathrm{a}} + \mathcal{L}_{\mathrm{d}} + \mathcal{L}_{\mathrm{pf}}$)	58.60	76.09	57.91	73.00

Table 3.5: Margin ablation using miniImageNet in 5-way classification. Bold/blue is best/second best, and $\pm$ indicates the 95% confidence intervals over 600 episodes.

Method	Backbone	1-shot	5-shot
MixtFSL-Neg-Margin	RN-12	**63.98** $\pm$0.79	**82.04** $\pm$0.49
MixtFSL-Pos-Margin	RN-12	63.57 $\pm$0.00	81.70 $\pm$0.49
MixtFSL-Neg-Margin	RN-18	**60.11** $\pm$0.73	77.76 $\pm$0.58
MixtFSL-Pos-Margin	RN-18	59.71 $\pm$0.76	77.59 $\pm$0.58

3.8 Extensions

We present extensions of our approach that make use of two recent works: the associative alignment of Afrasiyabi *et al.* [6], and Ordinary Differential Equation (ODE) of Xu *et al.* [12]. In both cases, employing their strategies within our framework yields further improvements, demonstrating the flexibility of our MixtFSL.

3.8.1 Associative alignment [6]

Two changes are necessary to adapt our MixtFSL to exploit the "centroid alignment" of Afrasiyabi *et al.* [6]. First, we employ the learned mixture model $\mathcal{P}$ to find the related base classes. This is both faster and more robust than [6] who rely on the base samples themselves. Second, they used a classification layer $\mathbf{W}$ in $c(\mathbf{x}|\mathbf{W}) \equiv \mathbf{W}^\top f(\mathbf{x}|\theta)$ (followed by softmax). Here, we use two heads ($\mathbf{W}^b$ and $\mathbf{W}^n$), to handle base and novel classes separately.

Evaluation We evaluate our adapted alignment algorithm on the miniImageNet and tieredImageNet using the RN-18 and RN-12. Table 3.6 presents our MixtFSL and MixtFSL-alignment (MixtFSL-Align.) compared to [6] for the 1- and 5-shot (5-way) classification problems. Employing MixtFSL improves over the alignment method of [6] in all cases except in 5-shot (RN-18) on tieredImageNet, which yields slightly worse results. However, our MixtFSL results in gain up to 1.49% on miniImageNet and 1.88% on tieredImageNet (5-shot, RN-12). To ensure a fair comparison, we reimplemented the approach proposed in [6] using our framework.

Forgetting Aligning base and novel examples improves classification accuracy, but may come at the cost of forgetting the base classes. Here, we make a comparative evaluation of this "remembering" capacity between our approach and that of Afrasiyabi *et al.* [6]. To do so, we first reserve 25% of the

Table 3.6: Comparison of our MixtFSL with alignment (MixtFSL-Align) in 5-way classification. Here, bold is the best performance.

	Method	Backbone	1-shot	5-shot
miniIN	Cent. Align.* [6]	RN-12	63.44 ±0.67	80.96 ±0.61
	MixtFSL-Align. (ours)	RN-12	**64.38** ±0.73	**82.45** ±0.62
	Cent. Align.* [6]	RN-18	59.85 ±0.67	80.62 ±0.72
	MixtFSL-Align. (ours)	RN-18	**60.44** ±1.02	**81.76** ±0.74
tieredIN	Cent. Align.* [6]	RN-12	71.08 ±0.93	86.32 ±0.66
	MixtFSL-Align. (ours)	RN-12	**71.83** ±0.99	**88.20** ±0.55
	Cent. Align.* [6]	RN-18	69.18 ±0.86	**85.97** ±0.51
	MixtFSL-Align. (ours)	RN-18	**69.82** ±0.81	85.57 ±0.60

* our implementation

base examples from the dataset, and perform the entire training on the remaining 75%. After alignment, we then go back to the reserved classes and evaluate whether the trained models can still classify them accurately. Table 3.7 presents the results on miniImageNet. It appears that Afrasiyabi *et al.* [6] suffers from catastrophic forgetting with a loss of performance ranging from 22.1–33.5% in classification accuracy. Our approach, in contrast, effectively remembers the base classes with a loss of only 0.5%, approximately.

3.8.2 Combination with recent and concurrent works

Several recent and concurrent works [12, 109, 110, 111] present methods which achieves competitive—or even superior—performance to that of MixtFSL presented in table 3.1. They achieve this through improvements in neural network architectures: [109] adds a stack of 3 convolutional layers as a pre-backbone to train other modules (SElayer, CSEI and TSFM), [110] uses a pre-trained RN-12 to train a "Cross Non-local Network", and [111] adds an attention module with 1.64M parameters to the RN-12 backbone. Xu *et al.* [12] also modify the RN-12 and train an adapted Neural Ordinary Differential Equation (ODE), which consists of a dynamic meta-filter and adaptive alignment modules. The aim of the extra alignment module in [12] is to perform channel-wise adjustment besides the spatial-level adaptation. In contrast to these methods, we emphasize that as opposed to these works, all MixtFSL results presented throughout the paper have been obtained with standard backbones *without additional architectural changes*.

Since this work focuses on representation learning, our approach is thus orthogonal—and can be combined—to other methods which contain additional modules. To support this point, table 3.8 combines MixtFSL with the ODE approach of Xu *et al.* [12] (MixtFSL-ODE) and shows that the resulting combination results in a gain of 0.85% and 1.48% over [12] in 1- and 5-shot respectively.

Table 3.7: Evaluation of the capacity to remember base classes before and after alignment. Evaluation performed on miniImageNet in 5-way image classification. Numbers in () indicate the change in absolute classification accuracy compared to before alignment.

Method	**Backbone**	**1-shot**	**5-shot**
[6] before align.	RN-12	96.17	97.49
[6] after align.	RN-12	65.47 (-30.7)	75.37 (-22.12)
ours before align.	RN-12	96.83	98.06
ours after align.	RN-12	96.27 (-0.6)	98.11 (+0.1)
[6] before align.	RN-18	91.56	90.72
[6] after align.	RN-18	58.02 (-33.5)	62.97 (-27.8)
ours before align.	RN-18	97.46	98.16
ours after align.	RN-18	97.20 (-0.3)	97.65 (-0.5)

Table 3.8: Combining MixtFSL with the ODE approach of Xu *et al.* [12] (MixtFSL-ODE) in 5-way on miniImageNet using RN-12.

Method	**1-shot**	**5-shot**
ODE [12]	67.76 $\pm$0.46	82.71 $\pm$0.31
MixtFSL-ODE	**68.61** $\pm$0.73	**84.19** $\pm$0.44

3.9 Discussion

This paper presents the idea of Mixture-based Feature Space Learning (MixtFSL) for improved representation learning in few-shot image classification. It proposes to simultaneously learn a feature extractor along with a per-class mixture component in an online, two-phase fashion. This results in a more discriminative feature representation yielding to superior performance when applied to the few-shot image classification scenario. Experiments demonstrate that our approach achieves state-of-the-art results with no ancillary data used. In addition, combining our MixtFSL with [6] and [12] results in significant improvements over the state of the art for inductive few-shot image classification. A limitation of our MixtFSL is the use of a two-stage training, requiring a choreography of steps for achieving strong results while possibly increasing training time. A future line of work would be to revise it into a single stage training procedure to marry representation and mixture learning, with stable instance assignment to components, hopefully giving rise to a faster and simpler mixture model learning. Another limitation is observed with small datasets where the within-class diversity is low such that the need for mixtures is less acute (cf. CUB dataset in fig. 3.3). Again, with a single-stage training, dealing with such a unimodal dataset may be better, allowing to activate multimodal mixtures only as required.

Chapter 4

Matching Feature Sets for Few-shot Image Classification

4.1 Résumé

Dans la classification d'images, il est courant d'entraîner les réseaux profonds à extraire un seul vecteur de caractéristiques par image d'entrée. Les méthodes de classification avec peu d'exemples (Few-shot) suivent également cette tendance. Dans ce travail, nous nous écartons de cette méthode et proposons plutôt d'extraire des ensembles de vecteurs de caractéristiques pour chaque image. Nous avançons qu'une représentation basée sur des ensembles permet, par sa nature même, de construire une représentation plus riche des images à partir des classes de base, qui peut ensuite être mieux transférée aux classes de peu d'images. Pour ce faire, nous proposons d'adapter les extracteurs de caractéristiques existants pour produire des ensembles de vecteurs de caractéristiques à partir des images. Notre approche, baptisée SetFeat, intègre des mécanismes d'auto-attention peu profonds dans les architectures des encodeurs existants. Les modules d'attention sont légers et, de ce fait, notre méthode permet d'obtenir des encodeurs qui ont approximativement les mêmes nombres de paramètres que leurs versions originales. Au cours de l'apprentissage et de l'inférence, une métrique de correspondance d'ensemble à ensemble est utilisée pour effectuer la classification des images. L'efficacité de l'architecture et des métriques proposées est démontrée par des expériences approfondies sur des ensembles de données standard avec peu d'exemples, à savoir miniImageNet, tieredImageNet et CUB, dans des scénarios à 1 et 5 données. Dans tous les cas mis à part un, notre méthode surpasse les autres méthodes en usage.

4.2 Abstract

In image classification, it is common practice to train deep networks to extract a single feature vector per input image. Few-shot classification methods also mostly follow this trend. In this work, we depart from this established direction and instead propose to extract sets of feature vectors for each

image. We argue that a set-based representation intrinsically builds a richer representation of images from the base classes, which can subsequently better transfer to the few-shot classes. To do so, we propose to adapt existing feature extractors to instead produce sets of feature vectors from images. Our approach, dubbed SetFeat, embeds shallow self-attention mechanisms inside existing encoder architectures. The attention modules are lightweight, and as such our method results in encoders that have approximately the same number of parameters as their original versions. During training and inference, a set-to-set matching metric is used to perform image classification. The effectiveness of our proposed architecture and metrics is demonstrated via thorough experiments on standard few-shot datasets—namely miniImageNet, tieredImageNet, and CUB—in both the 1- and 5-shot scenarios. In all cases but one, our method outperforms the state-of-the-art.

4.3 Introduction

The task of few-shot image classification is to transfer knowledge gained on a set of "base" categories, assumed to be available in large quantities, to another set of "novel" classes of which we are given only very few examples. To solve this problem, a popular strategy is to employ a deep feature extractor which learns to convert an input image into a feature vector that is both discriminative and transferable to the novel classes. In this context, the common practice found in the literature is to train a deep network to extract, for a given input, a *single* feature vector from which classification decisions are made.

In this paper, we depart from this established strategy by proposing instead to represent images as *sets* of feature vectors. With this, we aim at learning a richer feature space that is both more discriminative and easier to transfer to the novel domain, by allowing the network to focus on different characteristics of the image and at different scales. The intuition motivating that approach is that decomposing the representation into independent components should allow the capture of several distinctive aspects of images that can then be combined to efficiently represent images of novel classes.

To do so, we take inspiration from Feature Pyramid Networks [55] which proposes to concatenate multi-scale feature maps from convolutional backbones. In contrast, however, we do not just poll the features themselves but rather embed shallow self-attention modules (called "mappers") at various scales in the network. This adapted network therefore learns to represent an image via a set of attention-based latent representations. The network is first pre-trained by injecting the signal of a classification loss at every mapper. Then, it is fine-tuned in a meta-training stage, which performs classification by computing the distance between a query (test) and a set of support (training) samples in a manner similar to Prototypical Networks [9]. Here, the main difference is that the distance between samples is computed using set-to-set metrics rather than traditional distance functions. To this end, we propose and experiment with three set-to-set metrics.

This paper presents the following contributions. First and foremost, it presents the idea of reasoning on *sets* and a set-based inference of feature vectors extracted from images. It shows that set representation

yields improved performance on few-shot image classification without increasing the total number of network parameters. Second, it presents a simple and intuitive way for modifying *existing* backbones to make them extract *sets* of feature vectors rather than single ones, and processing them in order to achieve decisions. To do so, it proposed to embed simple self-attention modules in between convolutional blocks of the network, with examples of adapted three popular backbones, namely Conv4-64, Conv4-256, and ResNet12. It also proposes set-to-set metrics for evaluation of differences between query and support set. Third, it presents extensive experiments on three popular few-shot datasets, namely miniImageNet, tieredImageNet and CUB. In almost all cases, the proposed approach outperforms the state-of-the-art. Notably, our method gains 1.83%, 1.42%, and 1.83% accuracy in 1-shot over the baselines in miniImageNet, tieredImageNet and CUB, respectively. To support reproducible research, the code for our approach is available in the supplementary material and will be open sourced upon publication.

4.4 Related work

Our work falls within the domain of inductive few-shot image classification [7, 8, 9, 10] and investigates a metric function to infer the class of a query given a set of support examples. In this setting, previous works have broadly considered the following three problems of determining: 1) the best training framework; 2) the best matching metric; and 3) how to use additional data when available. The following covers the most relevant works under these three research contexts, and the other related works beyond the few-shot learning research area.

Training framework Two main training frameworks have been explored so far, namely meta learning or standard transfer learning. On one hand, meta learning [7, 8, 9, 10], also named episodic training, repeatedly samples small subsets of base classes to train the network, thereby simulating few-shot "episodes" during training. For example, some methods (e.g. [7, 24, 112]) aim at training a model to classify the novel classes with a small number of gradient updates. On the other hand, standard transfer learning methods [3, 4, 5, 6, 22] usually rely on a generic batch training with a metric-based (such as margin-based) criteria. Recently, several works [11, 12, 13] have shown that combining both standard transfer learning in a pre-training stage, following by a second meta-training stage can offer good performance. We employ a similar two-stage training procedure in this paper. From a feature extraction aspect, our method bears resemblance to FPN [55] which is proposed for object detection; however, our set-feature extractor embeds shallow self-attention mechanisms on such features.

Metrics Metric-based approaches [9, 10, 13, 26, 71, 27, 28, 29, 17, 30, 31, 32, 33, 34] aim at improving how the similarity/distance is calculated for better performance at training and inference time. In this aspect, our work is related to Prototypical Networks [9] as it also seeks to reduce the distance between a query and the centroid of a set of support examples of the corresponding class, while differing by proposing the use of *set-to-set* distance metrics for computing distance over several feature vectors. Other highly related works include FEAT [11], CTX [113], TapNet [70] and

ConstellationNet [114], which apply attention embedding adaptation functions on the episodes before computing the distance between query and the prototypes of the support set. Unlike them, our method extracts a set of different feature vectors given a query and support set, over which a set-to-set metric is applied for computing the distances.

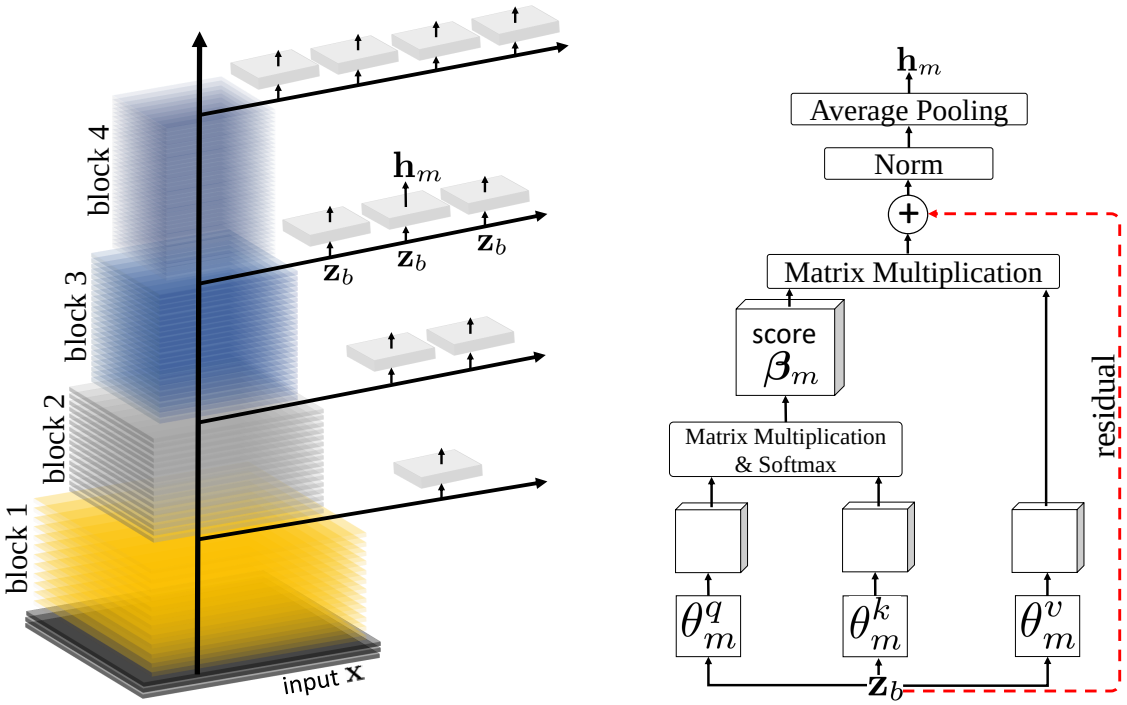
Extra data and transductive learning Relying on extra data [16, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50] is another strategy for building a well-generalized model. The augmented data can be in the form of hallucination with a data generator function [38, 46], using unlabelled data under semi-supervised [16, 115] or self-supervised [42, 116] frameworks, or aligning the novel classes to the base data [6]. In contrast, our approach does not require any additional data beyond the base classes. Moreover, it is inductive in that it does not need to exploit the structural information of the entire novel set, as transductive methods [69, 91, 92, 117] do.

Vision transformers and deep sets Our method employs shallow attention mappers that are inspired by the multi-head attention mechanism proposed in [20] and adapted to images by Dosovitskiy *et al.* [21]. In contrast to these works, our feature mappers work independently, are shallow (thus lightweight), and are not unified by FC-layers. We also employ several independent mappers at different depths/scales in the network. Also, while our feature mappers rely on convolutions as in [118, 119], our approach is focused on feature set matching. From a set-based perspective, Deep Sets [36] proposed a permutation invariant networks that operate on *input* sets. Our metrics rather aim at matching the feature set of support examples with the feature set of the query.

4.5 Preliminaries

In N-way K-shot (where K is small) image classification, we aim to predict the class of a given query example $\mathbf{x}_q$ from a support set $\mathcal{S}$ containing K training examples for each of the N different classes considered. Let $\mathcal{S}^n \in \mathcal{S}$ and $\mathcal{S}^n = \{(\mathbf{x}_i^n, y_i = n)\}_{i=1}^K$ be a set of example-label pairs, all pairs of that set $\mathcal{S}^n$ belonging to class n. In addition, let $f(\mathbf{x}|\theta^f)$ be a convolutional feature extractor composed of B blocks parameterized by $\theta^f = \{\theta_b^f\}_{b=1}^B$, where θ_b^f are the parameters of the b-th block. Here, a "block" broadly refers to a group of convolutional layers (with or without skip links), typically followed by a downscaling operation reducing the features spatial dimensions (e.g., pooling). The features after a given block b can be obtained from $\mathbf{z}_b \equiv f(\mathbf{x}|\{\theta_i^f\}_{i=1}^b)$.

In this work, we introduce a *set-feature* extractor, dubbed "SetFeat", which extracts a set of M feature vectors from images, rather than a single vector as it is typically done in the literature [10, 9, 7]. Formally, SetFeat produces the set $\mathcal{H} = \{\mathbf{h}_m\}_{m=1}^M$ of M feature vectors $\mathbf{h}_m$ through shallow self-attention mappers, and employs set-to-set matching metrics to establish the similarity between images in set-feature space. The following section presents our approach in detail.



(a) Set-feature extractor (SetFeat)

(b) Mapper $g(\cdot|\theta^g_m)$

Figure 4.1: The schematic overview of the proposed set-feature extractor (SetFeat) and detail of a single attention-based mapper: (a) given an input $\mathbf{x}$, SetFeat first extracts (convolutional) feature vectors $\mathbf{z}_b$ at each of its blocks, while at each block attention-based mappers (illustrated as small rectangles) convert $\mathbf{z}_b$ into a different embedding $\mathbf{h}_m$; (b) a single mapper m at block b extracts embedding $\mathbf{h}_m$ using an attention mechanism containing query θ^q_m and key θ^k_m to build attention scores $\boldsymbol{\beta}_m$, with self-attention inferred using value θ^v_m and score $\boldsymbol{\beta}_m$. This work focuses on backbones made of $B = 4$ blocks, consistent with popular few-shot image classification backbones such as Conv4 [10] and ResNet [79].

4.6 Set-based few-shot image classification

In this section, we first discuss our proposed set-feature extractor SetFeat, then dive into the details of our proposed set-to-set metrics. Finally, our proposed inference and training procedures are presented.

4.6.1 Set-feature extractor

The overall architecture of SetFeat is illustrated in 4.1. As mentioned in 4.5, its goal is to map an input image $\mathbf{x}$ to a feature set $\mathcal{H}$. To this end, and inspired by [20, 21], we embed segregated self-attention mappers $g(\cdot)$ throughout the network, as shown in 4.1a. We reiterate (3.4), however, that our mappers are different from multi-head attention-based models [20, 21] for two main reasons. First, each mapper in our approach is composed of a single attention head, thus we do not rely on fully connected layers to concatenate multi-head outputs. Our feature mappers are therefore separate from each other and each extract their own set of features. Second, our feature mappers are shallow (unit depth), with the learning mechanisms relying on the convolutional layers of the backbone.

The detail of the m-th feature mapper $g(\mathbf{z}_{b_m}|\theta^g_m)$, where b_m represents the block preceding the mapper, is illustrated in 4.1b. The learned representation $\mathbf{z}_{b_m} \in \mathbb{R}^{P\times D^p}$ is separated into P non-overlapping patches of D^p dimensions. In this work, we use patches of size 1×1, each patch is therefore a 1-D vector of D^p elements. Following Vaswani *et al.* [20], an attention map is first computed using two parameterized elements $q(\mathbf{z}_{b_m}|\theta^q_m)$ and $k(\mathbf{z}_{b_m}|\theta^k_m)$:

$$\boldsymbol{\beta}_m = \text{Softmax}\left(q(\mathbf{z}_{b_m}|\theta^q_m)k(\mathbf{z}_{b_m}|\theta^k_m)^\top/\sqrt{d_k}\right) , \tag{4.1}$$

where $\boldsymbol{\beta}_m \in \mathbb{R}^{P\times P}$ is the attention score over the patches of $\mathbf{z}_{b_m}$, and $\sqrt{d_k}$ is the scaling factor. Then, we compute the dot-product attention over the patches of $\boldsymbol{\beta}_m$ using $v(\mathbf{z}_{b_m}|\theta^v_m)$ in the following form:

$$\mathbf{a}_m = \boldsymbol{\beta}_m\, v(\mathbf{z}_{b_m}|\theta^v_m)\,, \tag{4.2}$$

where $\mathbf{a}_m \in \mathbb{R}^{P\times D^a}$ consists of P patches of D^a dimensions and D^a is the dimension of z_b. If the backbone feature extractor is ResNet [79] (see 4.7.1), we add a residual to the computed attention ($\mathbf{a}_m + \mathbf{z}_{b_m}$). In this case, if the dimensions mismatch ($D^a \neq D^p$), we use 1×1 convolution of unit stride and kernel size similar to downsampling. Finally, the feature vector $\mathbf{h}_m$ is computed by taking the mean of over the patches (over the P dimension).

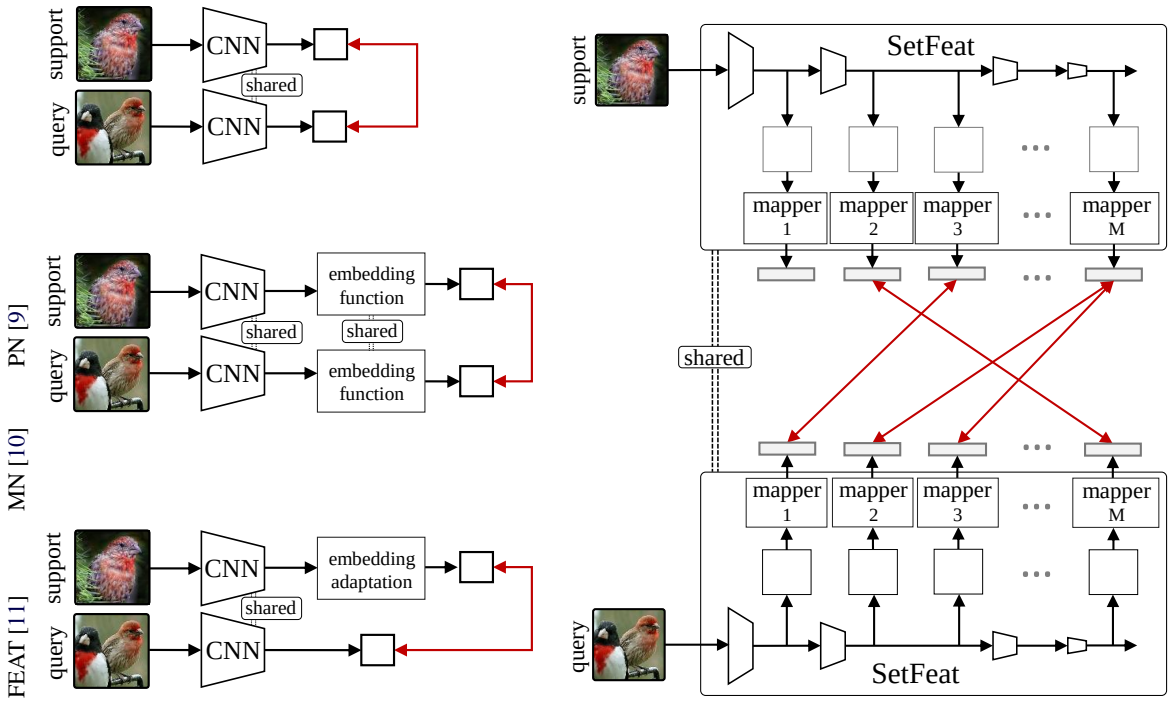
4.6.2 Set-to-set matching metrics

Having covered how SetFeat extracts a feature set for each input instance to process, we now proceed to how it leverages this set for image classification. In this context, we need to compare the feature set of the query with the feature sets corresponding to each instance of the support set of each class, to infer the class of the query. More specifically, in order to proceed with a distance-based approach as we do with prototypical networks, we need a set-to-set metric allowing the measure of distance over sets. We now present three distinct set-to-set metrics $d_{\text{set}}(\mathbf{x}_q, \mathcal{S}^n)$, which measure the distance between multiple feature sets, where $\mathbf{x}_q$ is the query, and $\mathcal{S}^n$ is a support set for class n (4.5). We employ the shorthand $\mathbf{h}_m(\mathbf{x}) \equiv g_m(\mathbf{z}_{b_m}|\theta^g_m)$ to refer to a feature extracted by mapper m. In addition, we also define $\bar{\mathbf{h}}_m(\mathcal{S}) \equiv \frac{1}{|\mathcal{S}|}\sum_{\mathbf{x}\in\mathcal{S}} \mathbf{h}_m(\mathbf{x})$ as the centroid of features extracted by mapper m on the support set $\mathcal{S}$. The following set metrics are built upon a generic distance function $d(\cdot,\cdot)$. In practice, we employ the negative cosine similarity function, e.g., $d(\cdot,\cdot) = -\cos(\cdot,\cdot)$.

Match-sum aggregates the distance between matching mappers for the query and supports:

$$d_{\text{ms}}(\mathbf{x}_q, \mathcal{S}^n) = \sum_{i=1}^{M} d\left(\mathbf{h}_i(\mathbf{x}_q), \bar{\mathbf{h}}_i(\mathcal{S}^n)\right) . \tag{4.3}$$

We use this metric as a baseline, as it parallels a common strategy of building representations simply by concatenating several feature vectors and invoking a standard metric on the flattened feature space.



(a) Three baseline methods

(b) Set matching with sum-min

Figure 4.2: Illustration of 1-shot image classification using (a) three existing methods and (b) our approach with the sum-min metric. (a) Given a query and support, existing methods either directly match the query to support (ProtoNet (PN) [9]), apply a single embedding function over both support and query (MatchingNetwork (MN) [10]), or perform embedding adaptation on the support before matching it with the query (FEAT [11]). (b) Our SetFeat method extracts sets of features for both of the support and query, which are then processed by the self-attention mappers. The set metric is then computed over the embeddings.

Min-min uses the minimum distance across all possible pairs of elements from the query and support set centroids:

$$d_{\text{mm}}(\mathbf{x}_q, \mathcal{S}^n) = \min_{i=1}^{M} \min_{j=1}^{M} d\left(\mathbf{h}_i(\mathbf{x}_q), \bar{\mathbf{h}}_j(\mathcal{S}^n)\right) . \tag{4.4}$$

Such a metric leverages directly the set structure of features.

Sum-min departs from the min-min metric by aggregating with a sum the minimum distances between the mappers computed on query and support set centroids:

$$d_{\text{sm}}(\mathbf{x}_q, \mathcal{S}^n) = \sum_{i=1}^{M} \min_{j=1}^{M} d\left(\mathbf{h}_i(\mathbf{x}_q), \bar{\mathbf{h}}_j(\mathcal{S}^n)\right) . \tag{4.5}$$

A schematic illustration of the sum-min metric is shown in 4.2, which also illustrates its difference with respect to three baseline few-shot models. Our method is different from FEAT [11] (and MN [10]) in two main ways. First, we define sets over features extracted from each example while FEAT/MN do so over the support set directly. In an extreme 1-shot case, the FEAT "set" degenerates to a single

Algorithm 5: SetFeat meta-training and validation.

Data: Network parameterized by $\theta = \{\theta^f, \theta^g\}$ made of a backbone of B convolution blocks ($\theta^f = \{\theta^f_b\}_{b=1}^B$) and M mappers ($\theta^g = \{\theta^g_m\}_{m=1}^M$); episodic train dataset $\mathcal{X}_{\text{train}}$ containing episodes of support set $\mathcal{S}$ and a query example $\mathbf{x}_q$; validation dataset $\mathcal{X}_{\text{valid}}$; maximum number of epochs $t^{\max}$; 0-1 loss function ℓ_{0-1} used to measure the validation accuracy.

Result: Best model defined as $\theta_{\text{best}} = \{\theta^f_{\text{best}}, \theta^g_{\text{best}}\}$

$E^{\text{best}}_{\text{valid}} \leftarrow \infty$

for $t = 1, \ldots, t^{\max}$ **do**

  for $(\mathbf{x}_q, \mathcal{S}) \in \mathcal{X}_{\text{train}}$ **do**

    $\ell^t \leftarrow -\log p(y_q|\mathbf{x}_q, \mathcal{S})$ using eq. 4.6

    Update network θ with backpropagation of loss ℓ^t

  end

  $\hat{y}_q \leftarrow \arg\min_{\mathcal{S}^n \in \mathcal{S}} d_{\text{set}}(\mathbf{x}_q, \mathcal{S}^n), \ \ \forall (\mathbf{x}_q, \mathcal{S}) \in \mathcal{X}_{\text{valid}}$

  $E_{\text{valid}} \leftarrow \frac{1}{|\mathcal{X}_{\text{valid}}|} \sum_{(\mathbf{x}_q, \mathcal{S}) \in \mathcal{X}_{\text{valid}}} \ell_{0-1}(\hat{y}_q, y_q)$

  if $E_{\text{valid}} < E^{\text{best}}_{\text{valid}}$ **then**

    $E^{\text{best}}_{\text{valid}} \leftarrow E_{\text{valid}}$

    $\theta_{\text{best}} \leftarrow \theta$

  end

end

element (1 support). Beneficial for few-shot, our work always keeps sets of many elements, regardless of the support set cardinality. Second, our method employs the parameterized mappers for set feature extraction. Here, we adjust the backbone (unlike FEAT and MN) so that adding mappers results in the same total number of parameters. Third, our method employs non-parametric set-to-set metrics, used for inference.

4.6.3 Inference

Given one of our metrics $d_{\text{set}} \in \{d_{\text{ms}}, d_{\text{mm}}, d_{\text{sm}}\}$ defined in the previous section, we follow the approach of Prototypical Networks [9] with SetFeat and model the probability of a query example $\mathbf{x}_q$ belonging to class $y = n$, where $n \in \{1, \ldots, C\}$ (N-way), using a softmax function:

$$p(y = n|\mathbf{x}_q, \mathcal{S}) = \frac{\exp(-d_{\text{set}}(\mathbf{x}_q, \mathcal{S}^n))}{\sum_{\mathcal{S}^i \in \mathcal{S}} \exp(-d_{\text{set}}(\mathbf{x}_q, \mathcal{S}^i))}\,, \tag{4.6}$$

with $\mathcal{S}$ as the (few-shot) support set.

4.6.4 Training procedure

We follow recent literature [11, 12, 13] and leverage a two-stage procedure to train SetFeat using one of our proposed set-to-set metrics. The first stage performs standard pre-training where a random batch $\mathcal{X}_{\text{batch}}$ of instances $\mathbf{x}$ from base classes are drawn from the training set. Here, we append fully-connected (FC) layers $\mathbf{o}_m$ to convert each of the mapper features $\mathbf{h}_m$ into logits in order to achieve classification over the C classes. From that, cross-entropy loss is used to train each mapper independently:

$$\ell_{\text{pre}} = -\sum_{\mathbf{x}_i \in \mathcal{X}_{\text{batch}}} \sum_{m=1}^{M} \log \frac{\exp(o_{m,y_i}(\mathbf{h}_{m,i}))}{\sum_{c=1}^{C} \exp(o_{m,c}(\mathbf{h}_{m,i}))}\,, \tag{4.7}$$

where $o_{m,c}$ is the FC layer output of mapper m for class c, $\mathbf{h}_{m,i}$ is the feature set of mapper m for instance $\mathbf{x}_i$, and y_i is the target output corresponding to instance $\mathbf{x}_i$.

The second stage discards the FC layers that were added in the first stage, and employs episodic training [10, 9] which simulates the few-shot scenario on the base training dataset. This stage is presented in 5. Specifically, we randomly sample N-way K-shot and Q-queries, then we compute the probability scores for each query using eq. 4.6. Finally, we update the parameters of the network after computing the cross-entropy loss.

4.7 Evaluation

This section first covers the details of our experiments with SetFeat, which are based on conventional backbones employed in the few-shot image classification literature. This is followed by description of the datasets and implementation details are described next. Finally, we present the evaluations of SetFeat with our set-matching metrics using four backbones with three datasets.

4.7.1 Backbones

We adopt the following three popular backbones, each composed of four blocks: (a) Conv4-64 [10], which consists of 4 convolution layers with 64/64/64/64 filters for a total of 0.113M parameters, (b) Conv4-512 [10]: 96/128/256/512 for 1.591M parameters, and (c) ResNet12 [17, 79]: 64/160/320/640 for 12.424M parameters. In all experiments below, we embed a total of 10 self-attention mappers throughout each backbone by following this per-block pattern: 1 mapper after block 1, then 2, 3 and 4 mappers for the three subsequent blocks. We experiment with other choices of mapper configurations in sec. 4.8.1.

Since our attention-based feature mappers require additional parameters, we correspondingly reduce the number of kernels in the backbone feature extractors to ensure that the performance gains are not simply due to the over-parameterization. Specifically, our SetFeat4-512, the counterpart of Conv4-512, uses a reduced set of 96/128/160/200 convolution kernels for a total of 1.583M parameters (compared to 1.591M for Conv4-512). SetFeat12, counterpart of ResNet12, consists of 128/150/180/512 kernels for 12.349M parameters (comp. 12.424M for ResNet12). For Conv4-64, reducing the amount of parameters collapses the training (as noted in [11, 111, 120]) since it contains very few parameters already. Our SetFeat4-64 therefore has more parameters (0.238M vs 0.113M for Conv4-64), but in sec. 4.8.4 we artificially augment the number of parameters for Conv4-64 and show our approach still outperforms it.

Convolutional attention [118] is used in SetFeat4-512 and SetFeat12. Particularly, we used single depth convolution and batch normalization to parameterize key, query and value in each mapper. The output dimension of the feature mappers is set to the number of channels in the last layer of the feature extractor — having all mappers producing feature vectors of the same dimension is a necessary

condition for our proposed metrics. For SetFeat4, FC-layers are used to compute the attention in order to limit the number of additional parameters as much as possible. The supplementary material includes the details of our implementation.

4.7.2 Datasets and implementation details

We conduct experiments on miniImageNet [10] (100/50/50 train/validation/test classes), tieredImageNet [16] (351/97/160), and CUB [18] (100/50/50). The first three are considered for object recognition, while the latter is used for fine-grained classification.

To pretrain SetFeat4, we used Adam [17] with a learning rate of 0.001 and weight decay of 5×10^{-4}. Batch size is fixed to 64. For SetFeat12, we used Nesterov momentum with an initial learning rate of 0.1, momentum of 0.9 and weight decay of 5×10^{-4}. We follow [11, 12, 13] for generic normalization and data augmentation. In the meta-training stage, SGD is used for all architectures. Validation sets are used to tune the schedule of the optimizer.

4.7.3 Quantitative and comparative evaluations

miniImageNet Table 4.1 presents evaluations of SetFeat with our set-to-set metrics on the miniImageNet dataset. First, we observe that our sum-min metric outperforms both the other proposed metrics and the state-of-the-art except in the 5-shot with SetFeat12. In particular, SetFeat4-64 (sum-min) results in an accuracy gain of 1.83% and 1.4% over MELR[111] in 1- and 5-shot, respectively.

tieredImageNet Table 4.2 presents the tieredImageNet evaluation of our SetFeat12 with our proposed metrics. Our sum-min metric results in 1.42% and 0.51 % improvement over the baseline Distill [34] in 1- and 5-shot. Please note that baselines such as Distill [34], MELR [111], and FEAT [11] contain more parameters compared to the original ResNet12 and our SetFeat12.

CUB Table 4.3 illustrates the fine-grained classification evaluation of our approach, compared to Conv4-64 and ResNet18. We observe that SetFeat4-64 (min-min) again surpasses all baselines by providing gains of 1.83% and 2.04% over MELR [111] in 1- and 5-shot respectively. When comparing with ResNet18, we further reduce the number of convolution kernels to 128/150/196/480 (dubbed SetFeat12*) to better match the number of parameters (11.466M for SetFeat12* vs 11.511M for ResNet18). Our approach again defines a new state-of-the-art performance in this scenario.

4.8 Ablation

In this section, we further analyze SetFeat to explore alternative design decisions and gain a better understanding as to why our set-based representation achieves better classification accuracy.

Table 4.1: Evaluation on miniImageNet in 5-way. Bold/blue is best/second, and ± is the 95% confidence intervals in 600 episodes.

	Method	Backbone	1-shot	5-shot
	ProtoNet [9]	Conv4-64	49.42±0.78	68.20±0.66
	MAML [23]		48.07±1.75	63.15±0.91
	RelationNet [30]		50.44±0.82	65.32±0.70
	Baseline++ [3]		48.24±0.75	66.43±0.63
	IMP [1]		49.60±0.80	68.10±0.80
	MemoryNet [103]		53.37±0.48	66.97±0.35
	Neg-Margin [100]		52.84±0.76	70.41±0.66
	MixtFSL [121]		52.82±0.63	70.67±0.57
	FEAT [11]		55.15±0.20	71.61±0.16
	MELR [111]		55.35±0.43	72.27±0.35
	BOIL [122]		49.61±0.16	66.45±0.37
Ours	Match-sum	SF4-64	55.74±0.65	72.18±0.70
Ours	Min-min		56.22±0.89	72.70±0.65
Ours	Sum-min		**57.18**±0.89	**73.67**±0.71
	ProtoNet† [9]	Conv4-512	53.52±0.43	73.34±0.36
	MAML [7]		49.33±0.60	65.17±0.49
	Relation Net [30]		50.86±0.57	67.32±0.44
	PN+rot [4]		56.02±0.46	74.00±0.35
	CC+rot [42]		56.27±0.43	74.30±0.33
	MELR [111]		57.54±0.44	74.37±0.34
Ours	Match-sum	SF4-512	56.50±0.85	72.69±0.68
Ours	Min-min		58.57±0.87	73.46±0.68
Ours	Sum-min		**59.10**±0.87	**74.97**±0.66
	AdaResNet [106]	ResNet12	56.88±0.62	71.94±0.57
	TADAM [17]		58.50±0.30	76.70±0.30
	MetaOptNet [73]		62.64±0.61	78.63±0.46
	Neg-Margin [100]		63.85±0.76	81.57±0.56
	MixtFSL [121]		63.98±0.79	82.04±0.49
	Meta-Baseline [123]		63.17±0.23	79.26±0.17
	Distill [22]		64.82±0.60	82.14±0.43
	DeepEMD [13]		65.91±0.82	82.41±0.56
	DMF [12]		67.76±0.46	82.71±0.31
	MELR [111]		67.40±0.43	**83.40**±0.28
	ProtoNet§ [9]		62.39	80.53
	FEAT§ [11]		66.78	82.05
Ours	Match-sum	SF-12 -	67.41±0.64	81.79±0.55
Ours	Min-min		67.88±0.55	82.07±0.61
Ours	Sum-min		**68.32**±0.62	82.71±0.46

§confidence interval not provided † taken from [111]

Table 4.2: TieredImageNet evaluation. Bold/red is best/second best, and ± indicates the 95% conf. intervals over 600 episodes of 5-way.

	Method	Backbone	1-shot	5-shot
	OptNet [73]	- ResNet12	65.99±0.72	81.56±0.53
	MTL [83]		65.62±1.80	80.61±0.90
	DNS [105]		66.22±0.75	82.79±0.48
	Simple [22]		69.74±0.72	84.41±0.55
	TapNet [70]		63.08±0.15	80.26±0.12
	ProtoNet† [9]		68.23± 0.23	84.03±0.16
	FEAT [11]		70.80±0.23	84.79±0.16
	MixtFSL [121]		70.97±1.03	86.16±0.67
	Distill [22]		71.52±0.69	86.03±0.49
	DeepEMD [13]		71.16±0.87	86.03±0.58
	DMF [12]		71.89±0.52	85.96±0.35
	MELR [111]		72.14±0.51	87.01±0.35
	Distill [34]		72.21±0.90	87.08±0.58
– Ours –	Match-sum	- SF12 -	71.22±0.86	85.43±0.55
	Min-min		71.75±0.90	86.40±0.56
	Sum-min		**73.63**±0.88	**87.59**±0.57

†taken from [73]

4.8.1 Probing the activation of mappers

Let us now investigate whether all mappers are actually useful by analyzing the behavior under the sum-min metric. For this, fig. 4.3 illustrates the percentage of time where a specific mapper (y-axis) provides the minimum prototype-query distance for each validation class (x-axis) in the miniImageNet dataset. This illustrates that low-level mappers are often active like the high-level ones, but all mappers are consistently being used across all validation classes, thereby validating that our proposed set-based representation is effective and working as expected.

In addition, fig. 4.4 shows t-SNE [63] visualizations of 640 embedded examples from miniImageNet, CUB, and tieredImageNet datasets using our set-feature extractor. Note how the distributions of mapper embeddings are generally disjoint and do not collapse to overlapping points, which shows intuitively that mappers extract different features.

4.8.2 Mapper configurations

We now experiment with different ways of embedding ten mappers throughout the backbone levels. We compare: 1) putting all mappers on the last layer (0-0-0-10); 2) a single mapper per block (1-1-1-1); 3) distributing mappers more equally (2-2-3-3); and 4) employing a progressive growth strategy (1-2-3-4) (this last one being used in the main evaluation in 4.7). Table 4.4 compares these four strategies on both SetFeat4-64 and SetFeat4-512 on the validation set of miniImageNet. We observe that placing mappers throughout the network yields better results than putting them all at the end.

Table 4.3: Fine-grained evaluation using CUB in 5-way. ± is the 95% confidence intervals on 600 episodes.

	Method	Backbone	1-shot	5-shot
	MatchingNet[10]	— Conv4-64 —	61.16±0.89	72.86±0.70
	ProtoNet[9]		64.42±0.48	81.82±0.35
	MAML[7]		55.92±0.95	72.09±0.76
	RelationNet[30]		62.45±0.98	76.11±0.69
	FEAT[11]		68.87±0.22	82.90±0.15
	MELR[111]		70.26±0.50	85.01±0.32
– Ours –	Match-sum	SF4-64	67.35±0.93	83.82±0.61
	Min-min		70.15±0.93	84.94±0.64
	Sum-min		**72.09**±0.92	**87.05**±0.58
	Robust-20 [74]	— ResNet18 —	58.67±0.7	75.62±0.5
	RelationNet‡ [30]		67.59±1.0	82.75±0.6
	MAML‡ [7]		68.42±1.0	83.47±0.6
	ProtoNet‡ [9]		71.88±0.9	86.64±0.5
	Baseline++ [3]		67.02±0.9	83.58±0.5
	MixtFSL [121]		73.94±1.1	86.01±0.5
	Neg-Margin [100]		72.66±0.9	89.40±0.4
– Ours –	Match-sum	- SF12*-	77.95± 0.83	88.93± 0.49
	Min-min		78.51±0.82	89.73±0.47
	Sum-min		**79.60**±0.80	**90.48**± 0.44

‡taken from [108]

The two other options perform similarly. We also observe that (2-2-3-3) only beats (1-2-3-4) using shallower network SetFeat4-64 in 5-shot. Otherwise, progressive growth either reaches or surpasses the other combinations. Note, that going from 0-0-0-10 to 1-2-3-4 or 2-2-3-3 improves performance while using the same number of mappers, which confirms that multi-scale indeed helps. Additionally, removing our set-based representation by concatenating all mappers outputs and treating the result as a single (multi-scale) feature vector ("concat" in tab. 4.4) completely cancels any performance gain. Therefore, we conclude that it is our sets of multi-scale features that explains the performance improvement, where our proposed set-based representation plays a key role.

4.8.3 Visualizing mappers saliency

We now visualize in fig. 4.5 the impact of learning a set of features by visualizing the saliency map of each mapper, and by comparing them with the saliency maps of the single-feature approach of Chen et al. [3]. We compute the smoothed saliency maps [124] by single back-propagation through a classification layer. It can be seen that our approach devotes attention to many more parts of the images than when a single feature vector is learned. For example, note how a single dog is highlighted (fourth row of fig. 4.5), whereas our mappers jointly fire on all three. Please consult the supplementary materials for more examples.

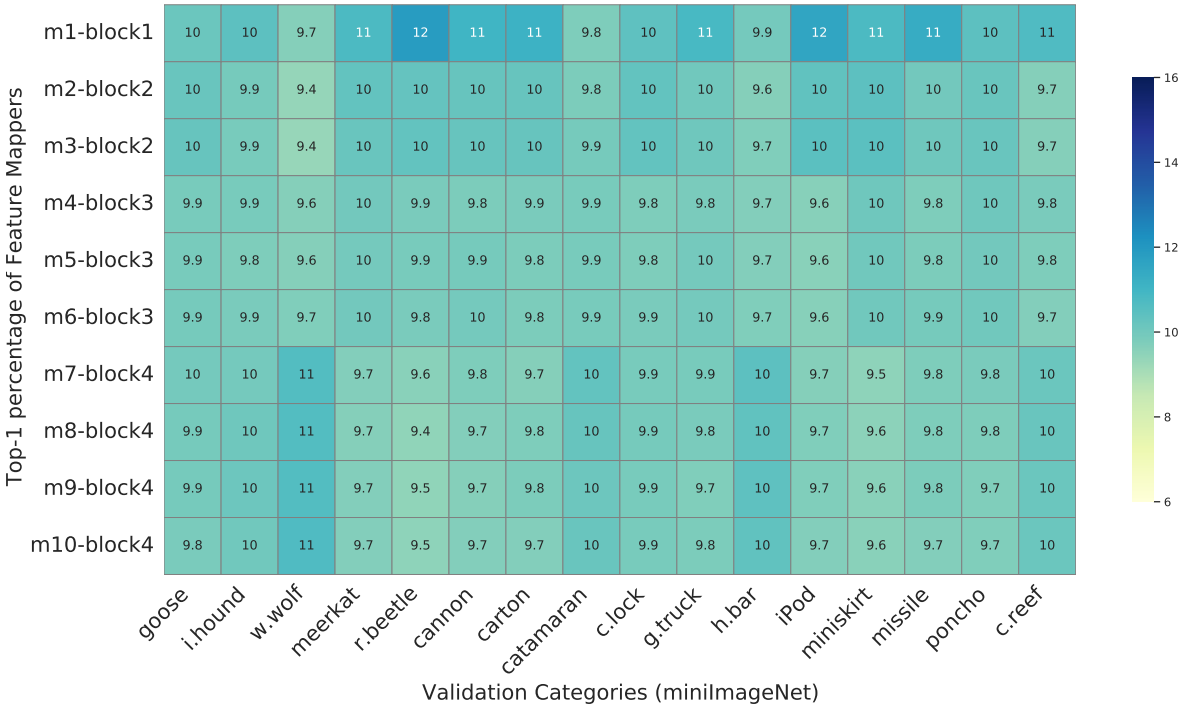
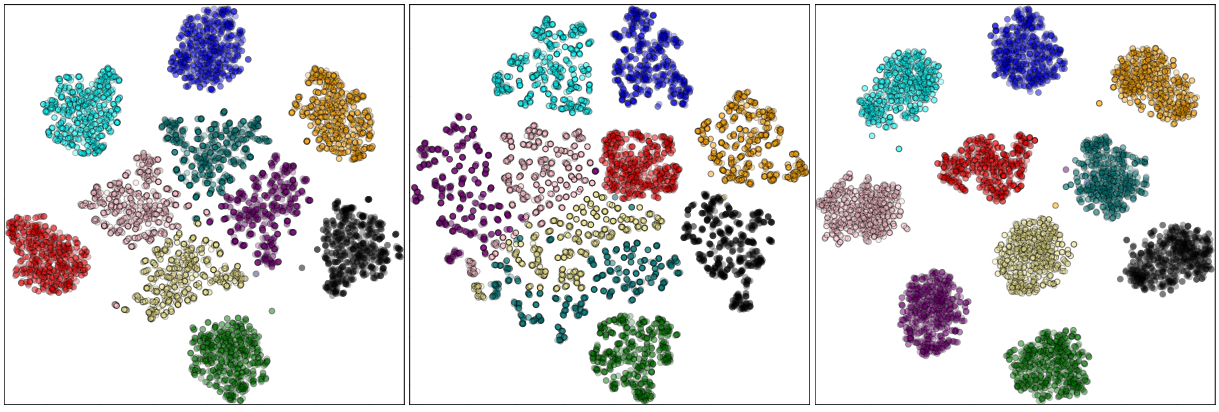
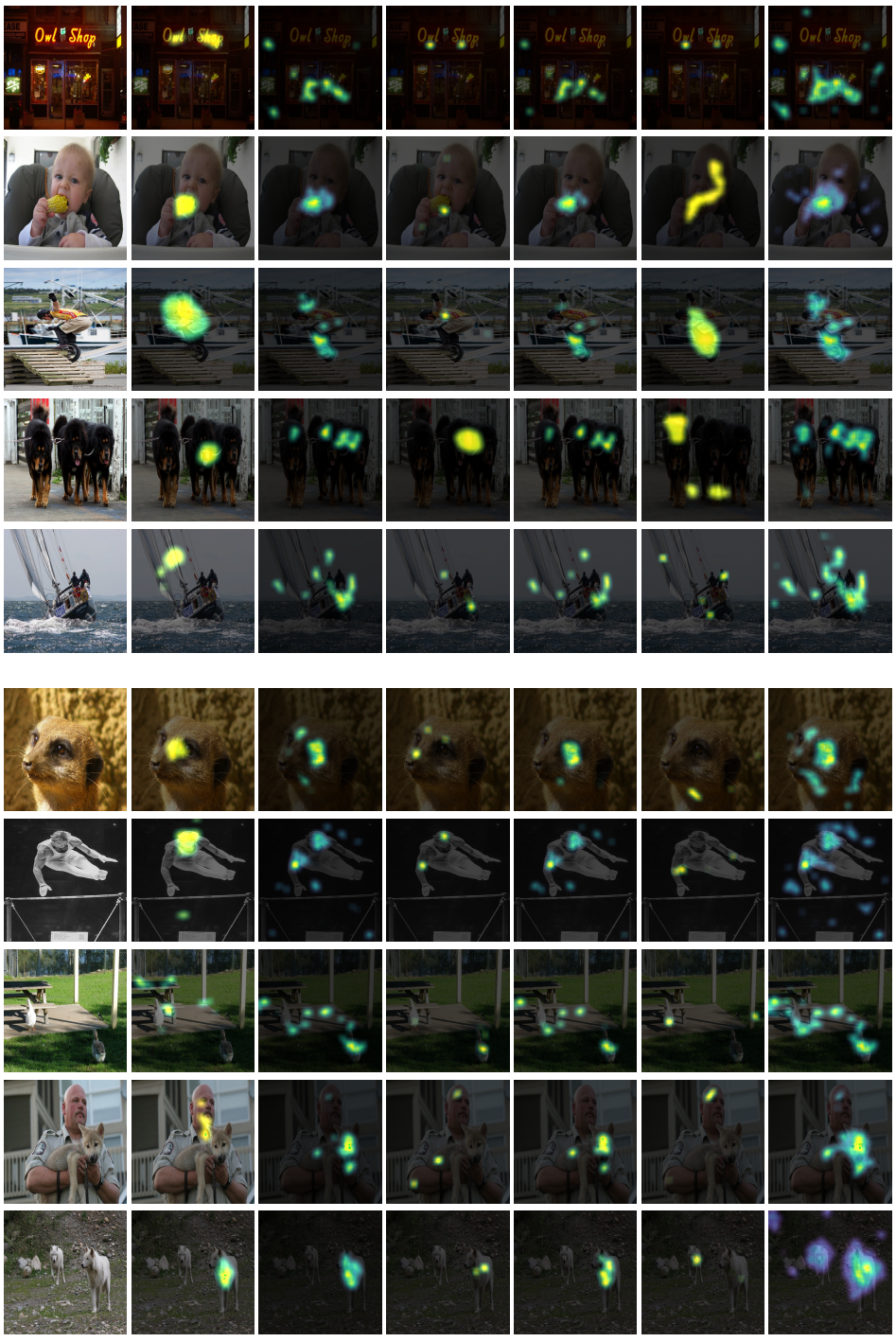


Figure 4.3: The percentage time each of the mappers (y-axis) is selected for each of the 16 validation categories (x-axis) of the miniImageNet dataset. The result is obtained by SetFeat12 and averaged over 600 episodes of 5-way 1-shot. While the earlier mappers are more often active, all mappers are consistently useful.



(a) miniImageNet (b) CUB (c) tieredImageNet

Figure 4.4: Visualizing mappers with t-SNE [63] on 640 randomly-sampled from validation set for (a) miniImageNet with SetFeat12, (b) CUB with SetFeat12* (4.7.3) and (c) tieredImageNet with SetFeat12. Points are color-coded according to the mapper.



(a) input (b) baseline (c) ours

Figure 4.5: Comparison on gradient saliency maps after training SetFeat12 on miniImageNet dataset. From left, we look at the (a) input original image, (b) baseline [3], and (c) subset of five feature vectors extracted by our set-feature extractor SetFeat12. The figure presents five examples of the training data in the first rows and five examples from the validation set of miniImageNet in the last five rows. See the supplementary materials for more examples.

Table 4.4: Ablation of different mapper-level combinations using miniImageNet. The results are validation accuracy with min-sum.

	SetFeat4-64		SetFeat4-512	
Mappers	1-shot	5-shot	1-shot	5-shot
ProtoNet*	53.51	71.57	–	–
0-0-0-1	53.55	71.51	–	–
1-2-3-4 (concat)	53.56	71.82	–	–
1-1-1-1	51.11	69.41	53.57	71.60
0-0-0-10	52.90	69.49	55.36	71.59
2-2-3-3	54.73	71.98	56.29	74.74
1-2-3-4	54.71	71.35	58.74	75.30

Table 4.5: Ablation of our SetFeat with miniImageNet and CUB on 600 episodes with augmented Conv4-64 and SetFeat4-64 in 5-way.

	miniImageNet		CUB	
Method	1-shot	5-shot	1-shot	5-shot
ProtoNet [9]	49.42	68.20	68.23	84.03
ProtoNet* [9]	49.98	69.53	69.11	85.27
Sum-min (ours)	57.18	73.67	73.50	87.61

* our implementation with augmented Conv4-64

4.8.4 Over-parameterization of SetFeat4-64

Sec. 4.7.1 mentioned that the number of kernels in backbone feature extractors was reduced in such a way that adding our proposed attention-based mappers did not significantly change the total number of parameters in the network—but unfortunately doing so for Conv4-64 resulted in poor generalization as each of its four blocks is only composed of a single layer with 64 kernels. Here, we instead *augment* Conv4-64 and add parameters with three FC layers (of 512, 160, 64 dimensions) after the convolutional blocks. This reaches 0.239M parameters, which matches the 0.238M parameters of SetFeat4-64. Results are presented in table 4.5. Although the augmented Conv4-64 improves over the baseline Conv4-64, the improvements are significantly below those obtained by SetFeat4-64, showing that the additional parameters alone do not explain the performance gap.

4.8.5 Top-m analysis

The min-min and sum-min metrics are two ends of the spectrum: min-min takes the minimum distance across all mappers, while sum-min computes the sum over all the mappers. Here, we sort the mappers according to distance and sum the top-m as an ablation shown in table 4.6. In general, we observe that the classification results progressively improve as we move towards sum-min, which uses all of the mappers.

Table 4.6: Ablation of top-m mapper in the min-sum metric using SetFeat4 and SetFeat12* on CUB. The results are validation set accuracy in 600 episodes.

	SetFeat4		**SetFeat12***	
Method	1-shot	5-shot	1-shot	5-shot
top-1 (min-min)	70.15	84.94	78.51	89.73
top-2	70.84	85.30	77.92	89.87
top-4	70.34	85.95	78.37	89.78
top-8	71.47	86.88	79.56	90.03
top-10 (sum-min)	**72.09**	**87.05**	**79.60**	**90.48**

4.9 Discussion

This paper proposes to extract and match *sets* of feature vectors for few-shot image classification. This contrasts with the use of a monolithic single-vector representation, which is a popular strategy in that context. To produce these sets, we embed shallow attention-based mappers at different stages of conventional convolutional backbones. These mappers aim at extracting distinct sets of features, capturing different properties of the images seen. In this aspect, random initialization is a factor which diversifies the mappers of the same block. However, the main factor is the non-linearity in the sum-min and min-min: the inner minimum distance causes a non-linearity that forces the selection of a given mapper, which creates diversity. Match-sum, our worst metric, only benefits from random initialization. We then rely on set-to-set matching metrics for inferring the class of a given query from the support set examples, following the usual approach for inference with prototypical networks. Experiments with four different adaptations of two main backbones demonstrate the effectiveness of our approach by achieving state-of-the-art results in miniImageNet, tieredImageNet, and CUB datasets. For fair comparison, the parameters of all the adapted backbones are reduced according to the number of parameters added by the mappers.

Limitations Even though a comparison with different mapper configurations has been provided in ablation section, we have evaluated our method using a fixed set of $M = 10$ mappers. Using more mappers ($M > 10$) has been considered, but was dismissed from experimental evaluation. Indeed, given that the experiments aim at making a fair comparison regarding the size of the networks, increasing the number of mappers would require reducing the number of backbone kernels, which in turn could cause underfitting at the backbone level due to the under-parameterization of the convolution blocks. As future work, we see great potential on analyzing the effect of increasing the number of mappers, possibly with larger backbones. Another topic requiring further investigations would be to vary the weighting of each mapper through more flexible set-to-set matching metrics. Although the min-sum and min-min metric non-linearly match the feature sets (through the $\min$ operation), investigating the weighted sum-min would be an interesting future work direction. For example, employing Deep Set [36] before computing the min-sum metric would be a potential direction to see the effect of weighted set-to-set mapping. Finally, we are also particularly enthusiastic regarding the

adaptation of our method to self-supervised methods, since the set of features provide more choices for the comparison of different variations of single images.

Conclusion

In this thesis, we propose **representation learning for few-shot image classification** tasks by employing deep learning. Few-shot learning aims to transfer knowledge from base classes with more examples to the novel classes with only a few labeled examples. In particular, this thesis presents three contributions to increase the model capacity of the conventional few-shot methods (as explained in Chapter 2), learn a robust multi-modal feature space in a mixture model learning way (as illustrated in Chapter 3), and effectively transfer learning with matching feature sets (as covered in Chapter 4). These three contributions are have been accepted by the top venues of computer science.

Generally, freezing a neural network model in few-shot learning looks inevitable after pre-training on base classes because any changes with a few novel examples would result in overfitting. Since fixing the model would limit the model capacity, we aim to increase the model capacity in generalizing to the novel classes as the first contribution. To do so, we offer to align the most related base classes to the novel classes. In particular, we **introduce associative alignment for few-shot image classification**, which matches the novel examples to the detected highly related base examples. This enables a practical and well-generalized fine-tuning stage. In this aspect, we first present a straightforward algorithm (after pre-training on base classes) to detect related base classes to the novel classes. Then, we proposed centroid alignment and adversarial alignment as two alignment strategies. The centroid alignment strategy reduces the intra-class variations by minimizing the distance between the novel classes samples and the centroid of their related base classes. However, the adversarial alignment uses an auxiliary critic network to match the distribution of the novel classes to their related bases without changing the intra-class structures. In addition to associative alignment, we also adopt a margin base loss function which benefits from its metric learning property for transfer learning. We also present an early stopping algorithm for the few-shot learning. Our extensive evaluation with four few-shot learning datasets (miniImageNet, tieredImageNet, FC100, and CUB) and three popular backbones (Conv4, ResNet-18, and WRN-28-10) demonstrates that associative alignment, specifically the centroid-based alignment, reaches new state-of-the-art few-shot classification accuracy. Besides the associative alignment, we also adopt a margin-based loss function and a novel early-stopping algorithm for few-shot image classification. The proposed idea of associative alignment [6] has been accepted to the European Conference on Computer Vision (ECCV) 2020 as a spotlight (5% acceptance rate). It has also been accepted to Montreal AI Symposium 2020.

While the traditional baseline models are based on uni-model assumption, Allen et al. [1] present the advantage of multi-modal assumption and propose infinite mixture prototypes (IMP) for few-shot learning. Unfortunately, IMP suffers from some limitations. Notably, they used a classical clustering algorithm called DP-means to cluster the embedded batch of base classes temporarily. To overcome the limitations of IMP, we propose **an end-to-end and fully differentiable mixture model method to increase the model adaptability** for few-shot image classification for the second contribution, namely "Mixture-based Feature Space Learning" (MixtFSL). Our proposed MixFSL consists of two training phases: initial training and progressive following. The initial training stage explores the mixture representation using a set of learnable components and two-loss functions, namely assignment loss and diversity loss. Particularly, the assignment loss minimizes the distance between examples and their class corresponding nearest learnable component. Alas, applying only assignment criteria results in backing to unimodal representation learning. Therefore, we have diversity criteria to keep all learnable components active. However, applying assignment and diversity criteria unstabilizes the training at the end of the initial training phase. MixtFSL shifts to the progressive following stage using a single loss function and a leader-follower training scheme to resolve this. Specifically, in the progressive following stage, we propose to use a fixed copy of the best-validated network to provide robust labeling for the follower network to capture the mixture representation. The experimental evaluation of MixtFSL with three popular backbones (Conv4, ResNet-12, and ResNet-18) generally results in a new state-of-the-art accuracy in four few-shot learning datasets: miniImageNet, tieredImageNet, FC100, and CUB. In addition, we combined MixtFSL with associative alignment (the first contribution) to form MixtFSL-align. The evaluation for MixtFSL-align with miniImagNet using ResNet-12 and ResNet-18 resulted in significant improvements over the baseline method under the inductive setting. Our MixtFSL [121] has been accepted and presented at the International Conference on Computer Vision (ICCV) 2021, and it also has been accepted to Montreal AI Symposium 2021 as an oral presentation.

As the thesis's third and last contribution, we propose a **set-based representation learning to increase the representation capacity** for few-shot image classification. Though single feature-based representation learning might not be problematic in generic image classification where new examples are from the same pre-trained classes, relying on a single feature would limit the representation learning power in transferring knowledge to the novel classes with few examples. In order to generalize the current single feature-based representation learning to set feature-based representation learning, we present a set feature extractor (SetFeat). SetFeat aims to learn independent features at different convolutional neural networks (ConvNets) blocks through attention-based learning modules. In this respect, we propose a light-weighted parametrized self-attention-based function called mappers. Specifically, we employ a set of proposed mappers to extract a set of features from a different stage of ConvNets. We enhance two single feature-based popular backbones Conv4 and ResNet-12, to build SetFeat4 and SetFeat-12, respectively. For a fair comparison in the evaluations, we reduce the number of convolution layers in original convolutional layers by the number of augmented attentional mapper's parameters. For example, ResNet-12 and SetFeat-12 both have almost the same learning parameters. However, our proposed SetFeat extracts a set of features for both the support set and query examples. Therefore,

novel set-based metrics are required since we have a set-to-set problem. In this regard, to compare the feature sets of support and query examples, we propose set-to-set metrics, namely match-sum, min-min, and sum-min. We conducted extensive experiments and ablation using three datasets (miniImageNet, tieredImageNet, and CUB) to compare our SetFeat with popular backbones (Conv4, ResNet-12, and ResNet-18) used in few-shot learning literature. Our experiment revealed that SetFeat with our sum-min metric achieves new state-of-the-art results. The idea of set-based representation learning titles as "Matching feature sets for few-shot image classification" [125] has been accepted to International Conference on Computer Vision and Pattern Recognition (CVPR) 2022.

While our first two offered methods focus on different parts of the training procedure, matching feature sets (our third contribution) method extends learning to all possible in-hand data. Specifically, MixtFSL only focuses on constructive learning using the base classes, but our associative alignment investigates effective fine-tuning. Notably, our associative alignment approach uses base and novel classes, while MixtFSL only uses base categories. However, our matching feature sets generalize the representation learning to the image space level. Therefore, matching feature sets not only aims at powerful pre-training but also beneficial learning from novel categories.

After discussing our three proposed associative alignment, MixtFSL, and matching feature sets approaches, let's see how these contributions are related. First, our associative alignment and MixtFSL are built by employing a standard transfer learning paradigm, unlike the matching feature set, which is a meta-learning (episodic training) approach. Second, all of our approaches are kind of initialization-based approaches which prepare effective representation learning. Third, our associative centroid alignment and matching feature set approaches are metric learning-based approaches since both investigate a new form of prototypical representation learning. Fourth, the matching feature set is a generalization version of MixtFSL in terms of multimodal representation learning, specifically with our presented match-sum and min-sum metric.

Apart from the relation of the three discussed approaches, we can also unify these methods to benefit from different perspectives. For example, we can combine our MixtFSL with associative alignment. Here, while MixtFSL can boost the model adaptability in the pre-training, our associative alignment can improve the model capacity during the learning with novel categories. As presented in chapter 3, we unify MixtFSL and associative alignment ideas (called MixtFSL+Align). Our evaluations resulted in classification accuracy gain using MixtFSL+Align over using one of them. As another example, we can integrate our set-based approach with associative alignment. We hypothesize that merging the ideas of matching feature sets and associative alignment can outperform MixtFSL+Align.

Although the evaluations of our presented three contributions result in the superiority of the associative alignment, MixtFSL, and matching feature sets, there are still some limitations that open possible future works that are worth if we could investigate them. Followings are the limitations with possible future works of our thesis:

1. As the limitation of the first contribution, our associative alignment approach (presented in Chapter 2) can introduce out-of-distribution (OOD) instances. In this aspect, centroid alignment has an advantage over adversarial alignment due to its metric learning property by pushing the novel examples to the centroid of their related base classes. However, the centroid alignment can potentially reduce the multi-modality to a single modality and reduce the adaptability of the network. Therefore, a promising future work direction would propose a mixture model that can be developed to prevent the reduction of associative alignment to a single model compared to centroid alignment without having the OOD problem of adversarial alignment. Our associative alignment is also based on the non-domain shift assumption. Specifically, we assumed that some base classes are related and associated with the novel classes. This assumption might not be a big problem in the current few-shot learning task since all of the SOTA approaches consider the in-domain assumption while performing transfer learning from the base to the novel classes. However, considering a significant domain shift between the base and novel categories, our associative alignment algorithm would end up with some results specifically in the 1-shot task. Therefore, another promising research path would be extending the idea of associative alignment to cross-domain few-shot learning. Finally, our associative alignment depends on the number of samples. Finding the most related base classes using only one novel example of the novel categories is not a trivial task.

2. As the limitation of the second contribution, our MixtFSL (presented in Chapter 3) contains a complex two-stage training algorithm. Though we try our best to simplify the training procedure of MixtFSL, the overall training algorithm is difficult, which would make it hard to use to tackle the real-world problem. For the future line of work, the two-stage training algorithm can be revised to a single-stage representation learning algorithm to fasten the extraction of the mixture model representation. As the second limitation, the advantage of MixtFSL might not be significant to the fine-grained tasks in which multimodal property stays in the dataset. For example, our evaluation with CUB (dataset of different bird classes) revealed that MixtFSL could not reach the baseline method in fine-grained for tiny dataset CUB. This possibly happens due to the multimodal nature of the CUB dataset since it only contains different bird classes. Another future work requires a novel mixture-based representation learning for the unimodal dataset. Finally, our MixtFSL can not be extended to the novel classes since learning a mixture model of a few examples is extremely difficult. However, one possible excellent investigation would be a theoretical analysis of mixture model learning during the novel class generalization. For instance, after illustrator of the guarantee for accuracy gain, one can work on learning mixture model by having, for example, 5-shot. We are not sure that extending MixtFSL to novel class representation learning is an easy task to tackle without theoretical investigation.

3. As the limitation of our third contribution, the matching set feature (presented in Chapter 4) is limited to using only a fixed number of the mappers. The proposed set-to-set matching metrics treat each feature in equal weight. Using more mappers has been considered, but we focus on the

Figure 4.6: Selected summary of results from chapter 2 and chapter 3 for MiniImagNet using Conv4 and tieredImageNet with ResNet12. While the left column presents 1-shot evaluation setup, the right column presents the 5-shot results. Our methods MixtFSL+alignment presents by unifying the idea of chapters two and three. Our results are coded in yellow and red.

fixed number due to the fair comparison with the same number of parameters between SetFeat and the conventional backbones. However, one interesting possible future work would be analyzing the number of mappers. Additionally, we used non-parametric set-to-set metric functions. In this aspect, a topic requiring further work would be investigating the weight parameterized metric and, for example, adapting deep set [36] to propose a weighted sum-min set-to-set matching. Another limitation of the presented SetFeat is that we only include experimental evaluations with standard backbones such as ResNet-12. One prominent future work for our set feature representation learning would be extending SetFeat to use full attention-based transformer type architecture instead of a hybrid (ConvNet+Transformer) architecture. We think having only a transformer-type approach would result in better generalization accuracy.

4. Finally, we think our MixtFSL (chapter 3) and set feature (chapter 4) approaches can be considered in other tasks behind the few-shot image classification. For example, MixtFSL and SetFeat can be applied to generic image classification, where the unseen query example is from one of the base classes. Our preliminary results with MixtFSL (discussed in chapter 3) illustrates the advantages of MixtFSL in generic image classification. However, a systematic study of both methods is promising as a future study.

Apart from the limitation and possible future work sketches, our presented *representation learning for*

few-shot image classification reaches to new state-of-the-art results under few-shot image classification context. Lets to summarize and highlight the obtained results. Plots in fig. 4.6 present the result in 1-shot and 5-shot with MiniImageNet and tieredImageNet dataset using Conv4 and ResNet12, respectively. MixtFSL+align. presents by unifying the ideas of associative alignment (chapter 2) and MixtFSL (chapter 3). As the plot shows, our SetFeat improves the literature methods except in 5-shot tieredImageNet, where our MixtFSL+align. is the best.

In conclusion, we hope this thesis's objectives can help machine learning algorithms take a step toward human-level learning with little supervision.

Appendix A

Supplementary Results on Associative Alignment

In this supplementary material, the following items are provided:

1. Validation error plot (sec. A.1);
2. Ablation study on B (sec. A.2);
3. Visualization (sec. A.3);
4. More ways (sec. A.4);
5. Comparison to no alignment (sec. A.5);
6. Sensitivity to wrongly-related classes (sec. A.6)
7. Ablation on the margin (sec. A.7)

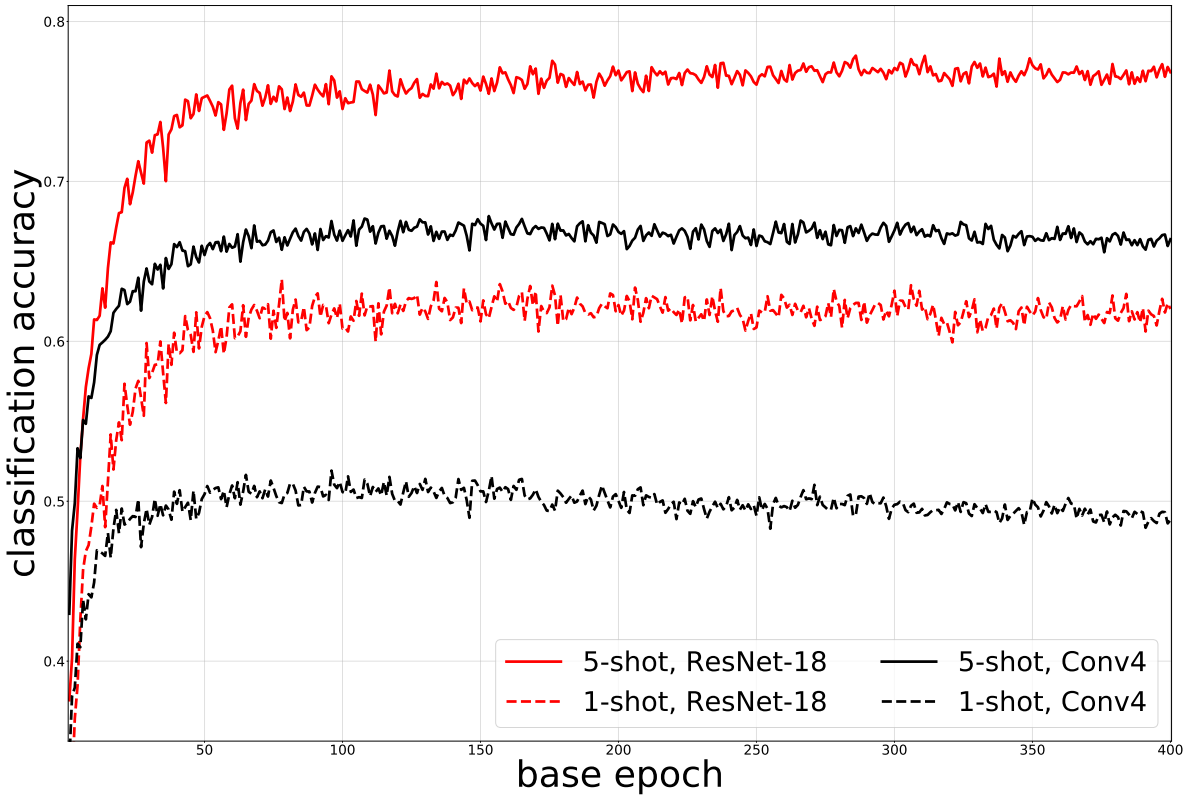


Figure A.1: Validation error after fine-tuning as a function of the number of pre-training base epochs on *mini*-ImageNet with the cosmax loss. Pre-training for a fixed number of iterations (here 400 as in [3]) may lead to overfitting the feature extraction on the base set. Each curve represents the average of 50 episodes.

A.1 Validation error plot

Fig. A.1 plots validation error after fine-tuning vs. the number of pre-training epochs. The "cosmax" function is used, with the entire network pre-trained on $\mathcal{X}^b$, and only the classification weights $\mathbf{W}$ fine-tuned on $\mathcal{X}^n$, as in [3]. The decrease in accuracy over the epochs (after 150 epoch for 1-shot) shows that pre-training should not be conducted for a fixed number of epochs.

A.2 Ablation study on B

Table A.1 presents an ablation study for B, the number of related base classes selected for each novel class. We perform the study on few-shot image classification on the *mini*-ImageNet dataset using ResNet-18 backbone. Overall, better results are obtained with a larger value of B, except for the adversarial alignment method in the 5-shot scenario.

Table A.1: Effect of three different number of related bases B on few-shot classification results on *mini*-ImageNet using ResNet-18 backbones. $\pm$ denotes the 95% confidence intervals over 600 episodes.

B	1-shot	5-shot
arcm.	58.07 $\pm$ 0.82	76.62 $\pm$ 0.58
1	55.76 $\pm$ 1.20	**79.34** $\pm$ 0.69
5	58.20 $\pm$ 1.14	78.65 $\pm$ 0.94
10	**58.84** $\pm$ 0.77	77.92 $\pm$ 0.82
12	58.79 $\pm$ 0.81	77.56 $\pm$ 0.85

(a) Adversarial alignment

B	1-shot	5-shot
arcm.	58.07 $\pm$ 0.82	76.62 $\pm$ 0.58
1	58.04 $\pm$ 0.98	77.54 $\pm$ 0.73
5	58.97 $\pm$ 1.06	79.14 $\pm$ 0.91
10	**59.88** $\pm$ 0.67	**80.23** $\pm$ 0.73
12	60.04 $\pm$ 0.77	80.18 $\pm$ 0.79

(b) Centroid alignment

A.3 Visualization of the alignment methods

Fig. A.2 presents a 2D visualization of our adversarial and centroid alignment methods using t-SNE [63] on *mini*ImageNet (see sec. 6.1 for the dataset description) dataset in 5-shot 5-way scenario. While both methods achieve similar results with $B = 1$, the centroid method results yields more discriminative class separation compared to the adversarial method with $B = 10$. The multi-modalities of certain

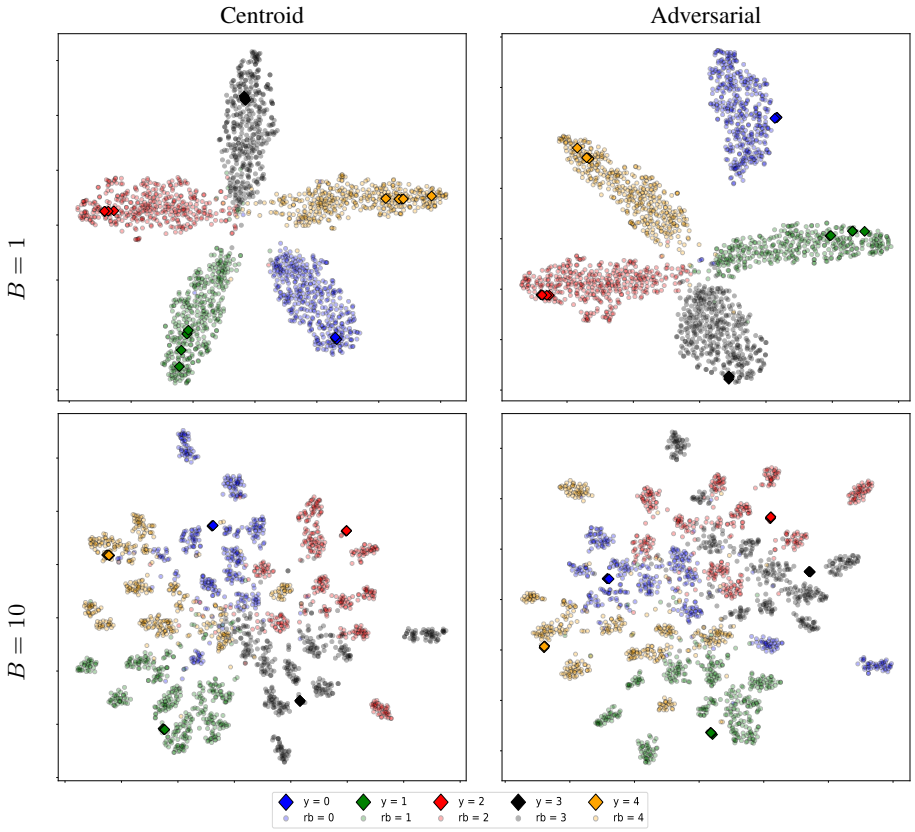


Figure A.2: Aligning novel and related base classes. Columns present centroid and adversarial distribution matching while the rows compare picking $B = 1$ and $B = 10$ related base classes for each novel class. We use t-SNE [63] to visualize the 512-dimensional feature space of ResNet-18 in 2D. Results are for 5-shot in a 5-way setting.

base categories look inevitable and might indeed degrade the generalization performance compared to the single-mode case assumed by our centroid alignment strategy. We compute the percentage of classes for which our centroid alignment approach: 1) improves, 2) does not change, or 3) deteriorates performance compared to our strong baseline (using a fixed threshold of 1% on classification accuracy). In the 5-shot scenario using ResNet-18 on *mini*-ImageNet, our centroid alignment approach results in improvements for 69.8% of the classes (with 13.9% not changing, and 16.3% deteriorates).

A.4 More-way

We experiment with N-way, 5-shot experiment (for N = 5, 10, 20) to examine the effect of associative alignment on more-way using *mini*-ImageNet. As Table A.2 presents, our associative alignment gains on the compared meta-learning and standard transfer learning methods. Specifically, we outperform the best of the compared method by 6.67%, 4.47%, 3.82% in 5-, 10-, and 20-way respectively. Note that we used 10, 5, 3 number of related base classes (B) 5-way, 10-way and 20-way respectively which corresponds to 60 classes out of all 64 base categories in *mini*-ImageNet.

Table A.2: N-way 5-shot classification results on *mini*-ImageNet using ResNet-18 backbone. $\pm$ denotes the 95% confidence intervals over 600 episodes. The best results prior this work is highlighted in blue, and the best results are presented in boldfaced.

	Method	**5-way**	**10-way**	**20-way**
meta-l.	MatchingNet‡ [10]	68.88 $\pm$ 0.69	52.27 $\pm$ 0.46	36.78 $\pm$ 0.25
	ProtoNet‡ [9]	73.68 $\pm$ 0.65	59.22 $\pm$ 0.44	44.96 $\pm$ 0.26
	RelationNet‡ [30]	69.83 $\pm$ 0.68	53.88 $\pm$ 0.48	39.17 $\pm$ 0.25
transfer-l.	softmax [3]	74.27 $\pm$ 0.63	55.00 $\pm$ 0.46	42.03 $\pm$ 0.25
	cosmax [3]	75.68 $\pm$ 0.63	63.40 $\pm$ 0.44	50.85 $\pm$ 0.25
	our baseline (sec. 5.1)	76.62 $\pm$ 0.58	62.95 $\pm$ 0.83	51.92 $\pm$ 1.02
	B	10	5	3
align.	adversarial	77.92 $\pm$ 0.82	64.87 $\pm$ 0.96	52.46 $\pm$ 0.99
	centroid	**80.35** $\pm$ 0.73	**68.17** $\pm$ 0.79	**54.67** $\pm$ 1.02

‡ implementation from [3]

A.5 Comparison to no alignment

Table A.3 illustrates the effect of training the network using both novel and their related classes, but without the alignment losses. The results are shown in the "no alignment" row in table A.3 below. Excluding the alignment loss slightly improves the accuracy compared to baseline by 0.82% and 0.24% in 1-shot and 5-shot using Conv4, respectively; however, it falls below the baseline by -2.13% and -2.34% in 1-shot and 5-shot using ResNet-18, respectively. In addition, except for the adversarial alignment in 1-shot using Conv4, both of the alignment strategies result in accuracy improvement in all of the scenarios, which shows the necessity of an alignment strategy.

Table A.3: Evaluating the necessity of alignment loss. Few-shot classification results on *mini*-ImageNet using both Conv4 and ResNet-18 backbones. $\pm$ denotes the 95% confidence intervals over 600 episodes.

		Conv4		ResNet-18	
		1-shot	5-shot	1-shot	5-shot
	baseline	51.90 $\pm$ 0.79	69.07 $\pm$ 0.62	58.07 $\pm$ 0.82	76.62 $\pm$ 0.58
	no alignment	52.72 $\pm$ 0.79	69.31 $\pm$ 0.69	55.94 $\pm$ 0.88	74.28 $\pm$ 0.83
alignment	adversarial	52.13 $\pm$ 0.99	70.78 $\pm$ 0.60	58.84 $\pm$ 0.77	77.92 $\pm$ 0.82
alignment	centroid	**53.14** $\pm$ 1.06	**71.45** $\pm$ 0.72	**59.88** $\pm$ 0.67	**80.35** $\pm$ 0.73

A.6 Sensitivity to wrongly-related classes

We also evaluate the sensitivity of the algorithm to the percentage of wrongly-related classes by replacing an increasing number of related base classes (selected by our algorithm) with random base classes instead (while keeping the total number of related base classes fixed to B=10). Results with the centroid alignment on *mini*-ImageNet and ResNet-18 are shown in table A.4.

Small changes to the selected classes have little impact on performance showing the stability of our approach. Replacing 5 randomly-selected base classes with random ones still results in improved performance in the 5-shot scenario. Even if heuristic, our related base class selection algorithm results in much improved performance compared to the 0/10 case.

Table A.4: Evaluating the sensitivity to wrongly-related classes. Few-shot classification results on *mini*-ImageNet using ResNet-18 backbone. $\pm$ denotes the 95% confidence intervals over 600 episodes.

selected / random	1-shot	5-shot
[paper] 10 / 0	**59.98** $\pm$ 0.7	**80.35** $\pm$ 0.7
9 / 1	59.74 $\pm$ 0.7	80.07 $\pm$ 0.9
8 / 2	59.77 $\pm$ 0.6	78.69 $\pm$ 0.8
5 / 5	58.36 $\pm$ 0.7	77.35 $\pm$ 0.8
0 / 10	56.72 $\pm$ 1.2	76.19 $\pm$ 0.8
[paper] baseline	58.07 $\pm$ 0.8	76.62 $\pm$ 0.6

A.7 Ablation on the margin m

We used episodic cross-validation to find the margin (m). In our experiments, we found that m needs to be adjusted according to the architectures rather than the datasets, which is likely due to its relation to the network learning capacity. An ablation for m on the *mini*-ImageNet validation set for the 5-way scenario is presented in table A.5.

Table A.5: ablation for margin (m) on the *mini*-ImageNet using ResNet-18 and Conv4 backbones. $\pm$ denotes the 95% confidence intervals over 600 episodes.

	Conv4		ResNet-18	
m	1-shot	5-shot	1-shot	5-shot
0.9	48.6	66.9	58.1	77.0
0.1	52.3	68.9	58.3	76.6
0.01	52.0	67.5	60.0	77.6

Appendix B

Supplementary Results on Mixture based Feature Space Learning

In this supplementary material, the following items are provided:

1. Ablation on the number of components N^k in the mixture model $\mathcal{P}$ (sec. B.1)
2. Dynamic of the training (sec. B.2);
3. More ways ablation (sec. B.3);
4. Ablation of the margin m (sec. B.4);
5. Ablation of the temperature τ (sec. B.5);
6. Visualization: from MixtFSL to MixtFSL-Alignment (sec. B.6);

B.1 Ablation on the number of components N^k in the mixture model $\mathcal{P}$

Although our proposed MixtFSL automatically infers the number of per-class mixture components from data, we also ablate the initial size of mixture model N^k for each class to evaluate whether it has an impact on the final results. Table B.1 presents 1- and 5-shot classification results on miniImageNet using ResNet-12 and ResNet-18 by initializing N^k to 5, 10, 15, and 20 components per class.

Initializing $N^k = 5$ results in lower classification accuracy compared to the higher N^k. We think this is possible due to the insufficient capacity of small mixture model $\mathcal{P}$ size. However, as long as N^k is sufficiently large (10, 15, 20), our approach is robust to this parameter and results do not change significantly as a function of N^k. Note that N^k cannot be set to an arbitrary high number due to memory limitations.

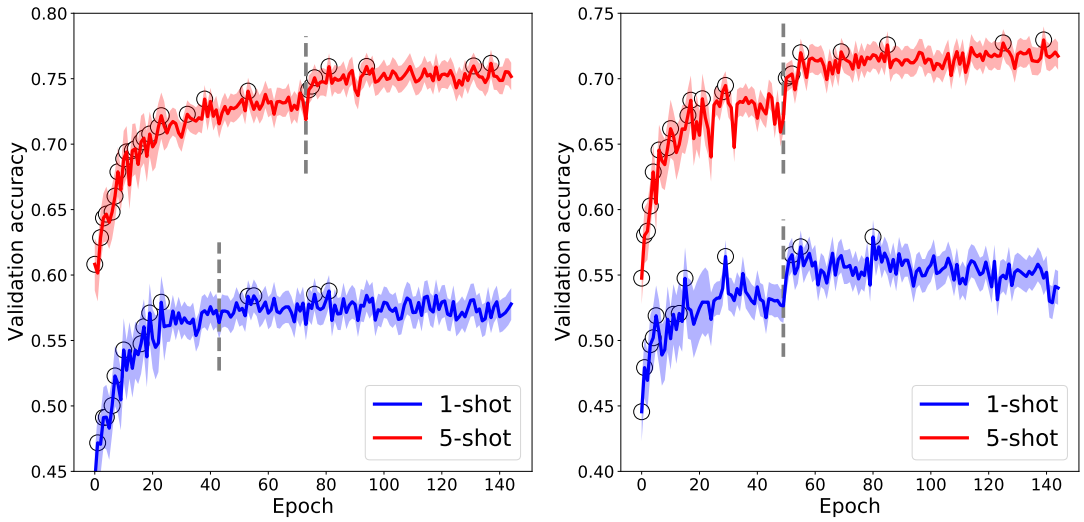
Table B.1: Classification results on mini-ImageNet using ResNet-12 and ResNet-18 backbones as a function of the initial value for the number of components per class N^k. $\pm$ denotes the 95% confidence intervals over 300 episodes.

N^k	1-shot	5-shot
5	62.29 $\pm$1.08	78.85 $\pm$0.61
10	64.01 $\pm$0.79	81.87 $\pm$0.49
15	63.98 $\pm$0.79	82.04 $\pm$0.49
20	63.91 $\pm$0.80	82.05 $\pm$0.49

(a) ResNet-12

N^k	1-shot	5-shot
5	58.57 $\pm$1.09	76.44 $\pm$0.61
10	60.15 $\pm$0.80	77.71 $\pm$0.61
15	60.11 $\pm$0.73	77.76 $\pm$0.58
20	58.99 $\pm$0.81	77.77 $\pm$0.58

(b) ResNet-18



(a) ResNet-12

(b) ResNet-18

Figure B.1: Validation accuracy of the first 150 epochs using ResNet-12 and ResNet-18 on miniImageNet. 1- and 5-shot scenarios are plotted using blue and red colors with their confidence intervals over 300 testing episodes of the validation set, respectively. The dashed vertical line is starting point of progressive following stage. The circles are the points when we update the best model.

B.2 Dynamic of the training

Fig. B.1 evaluates the necessity of the two training stages (sec. 4 from the main paper) by showing the (episodic) validation accuracy during 150 epochs. The vertical dashed line indicates the transition between training stages. In most cases, the progressive following stage results in a validation accuracy gain.

B.3 More ways ablation

Table B.2 presents more-way 5-shot comparison of our MixtFSL on miniImageNet using ResNet-18 and ResNet-12. Our MixtFSL gains 1.14% and 1.23% over the Pos-Margin [6] in 5-way and 20-way, respectively. Besides, MixtFSL gains 0.78% over Baseline++ [3] in 10-way.

We could not find "more-ways" results with the ResNet-12 backbone in the literature, but we provide our results here for potential future literature comparisons.

Table B.2: N-way 5-shot classification results on mini-ImageNet using ResNet-18 and ResNet-12 backbones. $\pm$ denotes the 95% confidence intervals over 600 episodes. The best results prior this work is highlighted in blue, and the best results are presented in boldfaced.

Method	Backbone	**5-way**	**10-way**	**20-way**
MatchingNet‡ [10]	RN-18	68.88 ±0.69	52.27 ±0.46	36.78 ±0.25
ProtoNet‡ [9]	RN-18	73.68 ±0.65	59.22 ±0.44	44.96 ±0.26
RelationNet‡ [30]	RN-18	69.83 ±0.68	53.88 ±0.48	39.17 ±0.25
Baseline [3]	RN-18	74.27 ±0.63	55.00 ±0.46	42.03 ±0.25
Baseline++ [3]	RN-18	75.68 ±0.63	63.40 ±0.44	50.85 ±0.25
Pos-Margin [6]	RN-18	76.62 ±0.58	62.95 ±0.83	51.92 ±1.02
MixtFSL (ours)	RN-18	**77.76** ±0.58	**64.18** ±0.76	**53.15** ±0.71
MixtFSL (ours)	RN-12	82.04 ±0.49	68.26 ±0.71	55.41 ±0.71

‡ implementation from [3]

Table B.3: Margin evaluation using miniImageNet in 5-way classification. Bold/blue is best/second best, and ± indicates the 95% confidence intervals over 600 episodes.

Method	Backbone	1-shot	5-shot
Neg-Margin* [100]	Conv4	51.81 ±0.81	69.24 ±0.59
ArcMax* [6]	Conv4	51.95 ±0.80	69.05 ± 0.58
MixtFSL-Neg-Margin	Conv4	52.76 ±0.67	**70.67** ±0.57
MixtFSL-Pos-Margin	Conv4	**52.82** ±0.63	70.30 ±0.59
Neg-Margin* [100]	RN-12	61.90 ±0.74	78.86 ±0.53
ArcMax* [6]	RN-12	61.86 ±0.71	78.55 ±0.55
MixtFSL-Neg-Margin	RN-12	**63.98** ±0.79	**82.04** ±0.49
MixtFSL-Pos-Margin	RN-12	63.57 ±0.00	81.70 ±0.49
Neg-Margin* [100]	RN-18	59.15 ±0.81	**78.41** ±0.54
ArcMax* [6]	RN-18	58.42 ±0.84	77.72 ±0.51
MixtFSL-Neg-Margin	RN-18	**60.11** ±0.73	77.76 ±0.58
MixtFSL-Pos-Margin	RN-18	59.71 ±0.76	77.59 ±0.58
Neg-Margin* [100]	WRN	62.27 ±0.90	80.52 ±0.49
ArcMax* [6]	WRN	62.68 ±0.76	80.54 ±0.50
MixtFSL-Neg-Margin	WRN	63.18 ±1.02	81.66 ±0.60
MixtFSL-Pos-Margin	WRN	**64.31** ±0.79	**81.63** ±0.56

* our implementation

B.4 Ablation of the margin

As table B.3 shows, a negative margin provides slightly better results than using a positive one, thus replicating the findings from Liu *et al.* [100], albeit with a more modest improvement than reported in their paper. We theorize that the differences between our results (in table B.3) and theirs are due to slight differences in training setup (e.g., learning rate scheduling, same optimizer for base and novel classes). Nevertheless, the impact of the margin on our proposed MixtFSL approach is similar. We also note that in all cases except 5-shot on ResNet-18, our proposed MixtFSL yields significant improvements. Notably, MixtFSL provides classification improvements of 2.08% and 3.18% in 1-shot and 5-shot using ResNet-12.

The margin m in eq.1 (sec. 4.1) is ablated in Table B.4 using the validation set of the miniImagNet dataset using ResNet-12 and ResNet-18. We experiment with both $m = 0.01$ to match Afrasiyabi *et al.* [6], and $m = -0.02$ to match Bin *et al.*l [100].

Table B.4: Margin m ablation on the miniImageNet using ResNet-12 and ResNet-18 backbones.

	ResNet-12		ResNet-18	
m	1-shot	5-shot	1-shot	5-shot
-0.02	61.85	80.38	60.57	79.04
+0.01	60.97	77.43	60.27	78.12

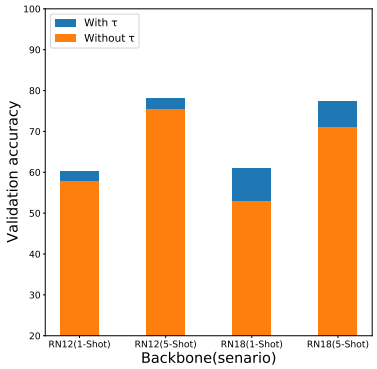


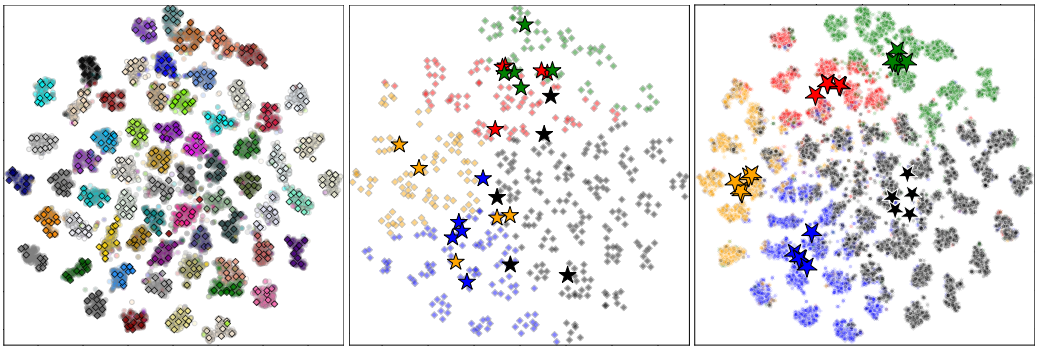
Figure B.2: Effect of temperature τ on MixtFSL using ResNet-12 and -18 in 1- and 5-shot scenarios in miniImageNet's validation set. The orange bars are the classification results without temperature variable ($\tau = 1$), and the blue colored bars are the amount of classification gain by training the backbone with temperature variable ($\tau = 0.05$).

B.5 Ablation of the temperature τ

We ablate the effect of having a temperature variable τ in the initial training stage using the validation set. As fig. B.2 presents, the validation set accuracy increases with the use of τ variable across the RN-12 and RN-18. Here, "without τ" corresponds to setting $\tau = 1$, and "with τ" to $\tau = 0.05$ (found on the validation set).

B.6 Visualization: from MixtFSL to MixtFSL-Alignment

Fig. B.3 summarizes the visualization of embedding space from our mixture-based feature space learning (MixtFSL) to its centroid alignment extension (sec. 6.1 from the main paper). Fig. B.3-(a) is a visualization of 200 base examples per class (circles) and the learned class mixture components (diamonds) after the progressive following training stage. Fig. B.3-(b) presents the t-SNE visualization of novel class examples (stars) and related base detection (diamonds of the same color) using our proposed MixtFSL. Fig. B.3-(c) presents the visualization of fine-tuning the centroid alignment of [6]. Here, the novel examples align to the center of their related bases.



(a) after progressive following (b) related-modes detection (c) after alignment

Figure B.3: t-SNE [63] applied to the ResNet-12 base feature embedding. (a) learned base categories feature embedding (circles) and mixture components (diamonds) after the progressive following stages. (b) using 5-way (coded by color) novel example shown by stars to detect their related base classes with the learned mixture components shown by diamonds. (b) aligning the novel examples to the center of their related base classes without forgetting the base classes. Points are color-coded by related base and novel examples.

Appendix C

Supplementary Results on Matching Feature Sets

In this supplementary material, the following items are provided:

1. Ablation with more ways and cross-domain results from miniImageNet $\mapsto$ CUB (C.1);
2. Visualizing mappers saliency (C.2);
3. Class structure in cluster (C.3);
4. Hausdorff distance ablation (C.4);

Table C.1: Specifications of miniImageNet, tieredImageNet and CUB.

Dataset	Number of examples	Source	train/val/test	Reference
MiniImageNet	60,000	ImageNet† [2]	64/16/20	[10]
TieredImageNet	779,165	ImageNet† [2]	351/97/160	[16]
CUB	11,788	CUB-200-2011* [18]	100/50/50	[3]

† https://www.image-net.org/ * http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

Table C.2: Number of parameters for various backbones, compared with our SetFeat implementations (in blue). Blocks column illustrates the number of parameters in all the convolution layers. Mappers column shows the number of parameters in 10 employed mappers in SetFeat.

Backbone	Blocks	Mappers	Total
Conv4-64	0.113 M	–	0.113 M
SetFeat4-512	0.113 M	0.124 M	0.238 M
Conv4-512	1.591 M	–	1.591 M
SetFeat4-512	0.587 M	0.996 M	1.583 M
ResNet18	11.511 M	–	11.511 M
SetFeat12*	6.977 M	4.489 M	11.466 M
ResNet12	12.424 M	–	12.424 M
SetFeat12	7.447 M	4.902 M	12.349 M

C.1 Ablation with more ways and cross-domain results from miniImageNet ↦ CUB

sec. C.3 shows 5-way, 10-way, and 20-way comparisonds of SetFeat12* and SetFeat12 with ResNet18 and ResNet12, respectively. As illustrated in reftab:backboneparameters and mentioned in sec. 5.3 of the main paper, SetFeat12* (11.466M parameters) is the counterpart of ResNet18 (11.511M parameters).

sec. C.3 shows that SetFeat with the sum-min metric (eq. (5) from the main paper) achieves state-of-the-art results in 5-shot for all of 5-, 10- and 20-way classification. Notably, SetFeat12* and SetFeat12 gain 6.18% and 2.84% over MixtFSL [121] in 5-way, respectively. Additionally, last column of sec. C.3 shows cross domain adaptation, where we pre-train our model on miniImageNet and test on the CUB dataset. Here, our SetFeat12* obtains the second best and is 0.92% below MixtFSL [121].

Table C.3: N-way 5-shot classification results on miniImageNet using ResNet and SetFeat. $\pm$ denotes the 95% confidence intervals over 600 episodes. The best results prior to this work is highlighted in red, and the best results are presented in boldface.

		miniImageNet			**miniIN→CUB**
Method	B.b.	**5-way**	**10-way**	**20-way**	**5-way**
MN‡ [10]	ResNet18	68.88 ±0.69	52.27 ±0.46	36.78 ±0.25	–
Neg-Margin‡ [100]	ResNet18	–	–	–	67.03 ±0.80
ProtoNet‡ [9]	ResNet18	73.68 ±0.65	59.22 ±0.44	44.96 ±0.26	62.02 ±0.70
Rel.Net‡ [30]	ResNet18	69.83 ±0.68	53.88 ±0.48	39.17 ±0.25	57.71 ±0.70
Baseline [3]	ResNet18	74.27 ±0.63	55.00 ±0.46	42.03 ±0.25	65.57 ±0.25
Baseline++ [3]	ResNet18	75.68 ±0.63	63.40 ±0.44	50.85 ±0.25	64.38 ±0.90
Pos-Margin [121]	ResNet18	76.62 ±0.58	62.95 ±0.83	51.92 ±1.02	64.93 ±1.00
MixtFSL [121]	ResNet18	77.76 ±0.58	64.18 ±0.76	53.15 ±0.71	**68.77** ±0.90
Sum-min (ours)	SF12*	**81.22**±0.45	**70.36**±0.46	**57.36**±0.36	67.85±0.70
MixtFSL [121]	RN12	82.04 ±0.49	68.26 ±0.71	55.41 ±0.71	–
Sum-min (ours)	SF12	**82.71**±0.46	**71.10**±0.46	**57.97**±0.36	–

‡ implementation from [3]
SF12 refers to SetFeat12 and RN12 refers to ResNet12

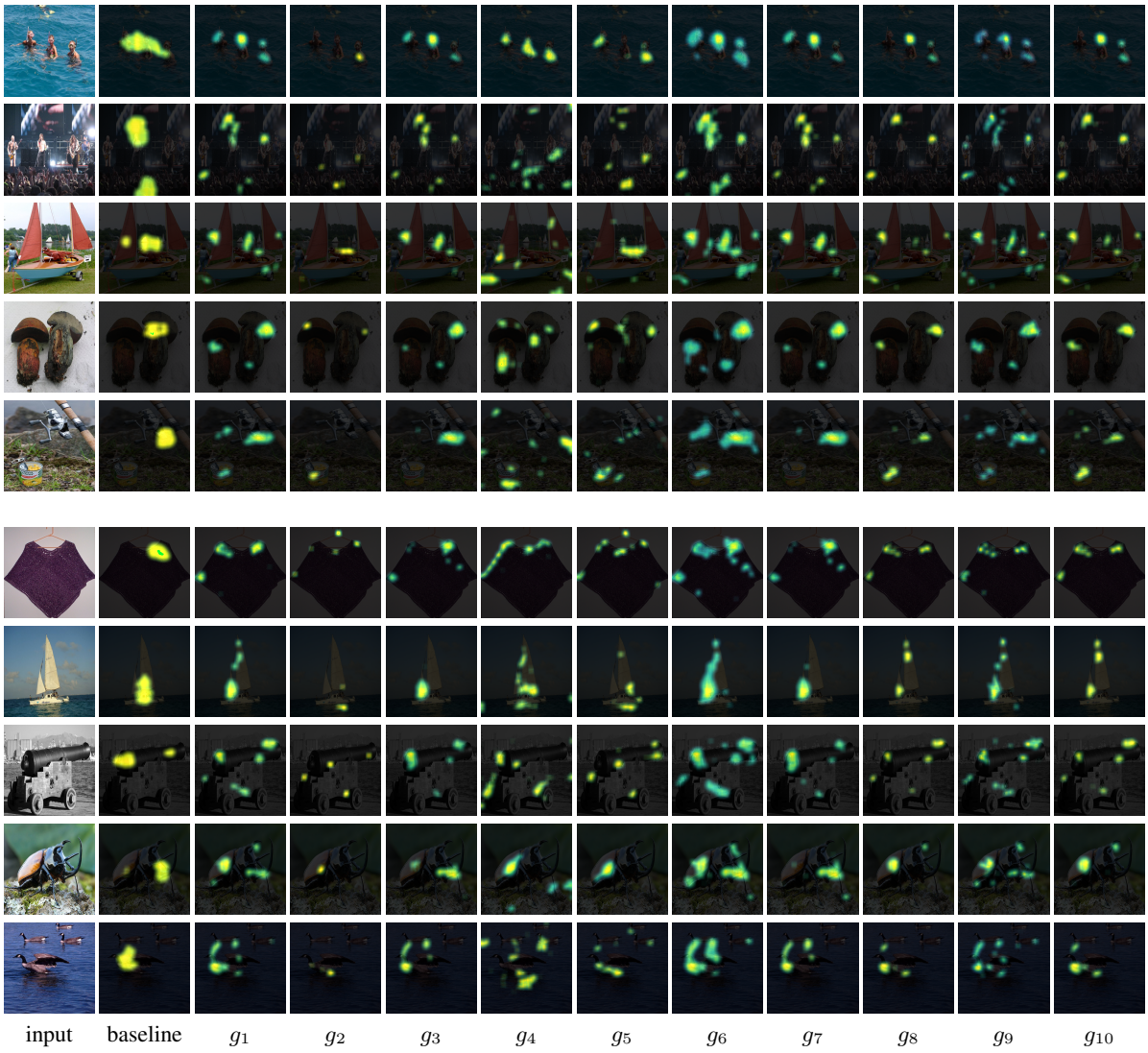


Figure C.1: Gradient saliency maps after training SetFeat4-64 on miniImageNet. From left: input image, baseline [3] trained with Conv4-64, and 10 different mappers from our SetFeat4-64 (g_i is the i-th mapper). The first five rows show examples from the training dataset, and the last five are from the validation set of miniImageNet.

C.2 Visualizing mappers saliency

Figure. C.1 and figure C.2 compare the gradient saliency maps of SetFeat12 and SetFeat4-64 using our sum-min metric with ResNet12 and Conv4-64 using "baseline" from [39]. Here SetFeat4-64 uses an FC-layer to compute mappers, while SetFeat12 uses a convolutional layer to do so. As shown in the figures, different mappers focus on different regions of the input image.

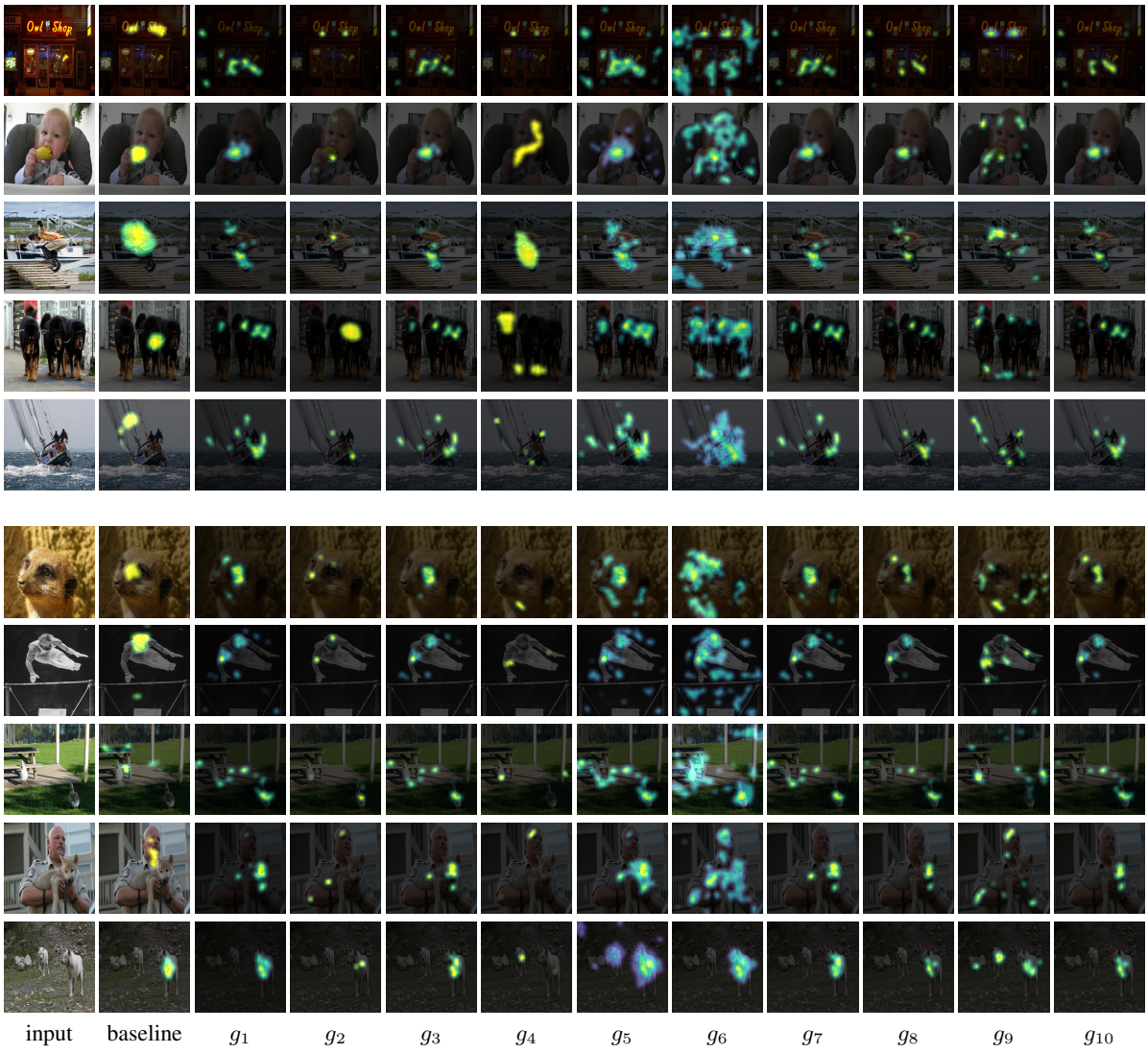


Figure C.2: Gradient saliency maps after training SetFeat12 on miniImageNet. From left: input image, baseline [3] trained with ResNet12, and 10 different mappers from our SetFeat12 (g_i is the i-th mapper). The first five rows show examples from the training dataset, and the last five are from the validation set of miniImageNet.

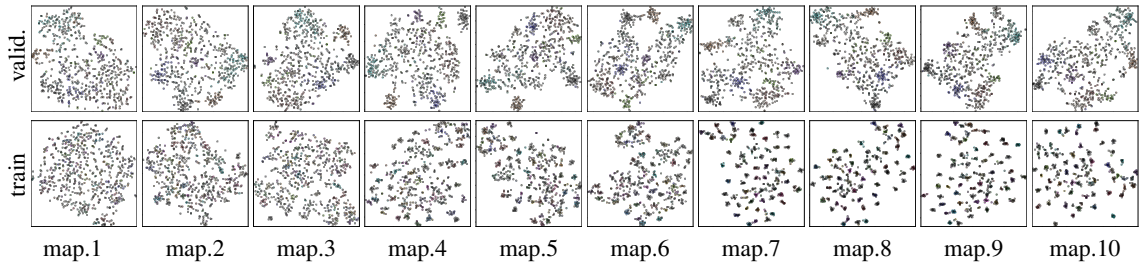


Figure C.3: tSNE of miniIN's 640 samples of 64 train and 16 valid. classes (color-coded) by SF12 mappers separately (columns).

C.3 Class structure in cluster

Fig. C.3 shows that tSNE for each mapper independently exhibits the expected class structure for both validation (top row) and train (bottom row) sets. Since tSNE applied over all mappers jointly on the validation set in fig. 4 of chapter 4, the largest variation (across mappers) is captured.

C.4 Hausdorff distance ablation

Our matching feature set work can be extended to other set distances. Tab. C.4 presents our method with Hausdorff (in blue) compared to our Sum-min for both miniIN and CUB.

Table C.4: MiniIN (from Table 1) and CUB (from Table 3) by SF4-64 plus blue.

	config.	1-shot	5-shot
miIN	Sum-min	**57.18**	**73.67**
	Hausdorff	56.07	72.32
CUB	Sum-min	**72.09**	**87.05**
	Hausdorff	70.20	84.85

Bibliography

[1] Kelsey R Allen, Evan Shelhamer, Hanul Shin, and Joshua B Tenenbaum. Infinite mixture prototypes for few-shot learning. *International Conference on Machine Learning (ICML)*, 2019. iv, 4, 6, 18, 19, 40, 43, 48, 49, 66, 75

[2] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 2015. 1, 34, 40, 48, 96

[3] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations (ICLR)*, 2019. 1, 3, 4, 6, 10, 26, 27, 28, 29, 33, 34, 35, 36, 37, 38, 40, 42, 43, 49, 51, 58, 66, 68, 70, 81, 84, 91, 96, 97, 98, 99

[4] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 10, 26, 28, 40, 42, 58, 66

[5] Hang Qi, Matthew Brown, and David G. Lowe. Low-shot learning with imprinted weights. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 10, 26, 28, 42, 58

[6] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 10, 40, 42, 43, 49, 50, 51, 52, 53, 54, 55, 58, 59, 74, 91, 92, 94

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017. 1, 4, 7, 10, 11, 12, 26, 27, 33, 34, 40, 42, 49, 51, 58, 59, 66, 68

[8] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2016. 1, 10, 26, 27, 34, 35, 42, 58

[9] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Neural Information Processing Systems (NeurIPS)*, 2017. 1, 3, 5, 6, 7, 10, 12, 13, 18, 26, 27, 28, 31, 33, 34, 35, 36, 37, 40, 42, 49, 50, 51, 57, 58, 59, 62, 63, 64, 66, 67, 68, 71, 84, 91, 97

[10] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Neural Information Processing Systems (NeurIPS)*, 2016. 1, 3, 4, 6, 7, 9, 10, 12, 13, 14, 26, 28, 33, 34, 35, 40, 42, 48, 49, 58, 59, 60, 62, 64, 65, 68, 84, 91, 96, 97

[11] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 10, 12, 14, 15, 58, 62, 63, 64, 65, 66, 67, 68

[12] Chengming Xu, Yanwei Fu, Chen Liu, Chengjie Wang, Jilin Li, Feiyue Huang, Li Zhang, and Xiangyang Xue. Learning dynamic alignment via meta-filter for few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3, 10, 42, 53, 54, 55, 58, 63, 65, 66, 67

[13] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. DeepEMD: Few-shot image classification with differentiable Earth Mover's Distance and structured classifiers. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 10, 12, 42, 58, 63, 65, 66, 67

[14] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 2017. 5, 27, 32, 33, 34

[15] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5, 27, 33, 44, 45

[16] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *International Conference on Learning Representations (ICLR)*, 2018. 6, 7, 9, 15, 16, 28, 34, 42, 48, 59, 65, 96

[17] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: Task dependent adaptive metric for improved few-shot learning. In *Neural Information Processing Systems (NeurIPS)*, 2018. 6, 7, 12, 28, 34, 36, 37, 42, 48, 49, 50, 58, 64, 65, 66

[18] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD birds-200-2011 dataset, 2011. 6, 7, 9, 34, 48, 65, 96

[19] Brian Kulis and Michael I Jordan. Revisiting k-means: New algorithms via Bayesian nonparametrics. In *International Conference on Machine Learning (ICML)*, 2012. 6, 40, 43

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems (NeurIPS)*, 2017. 8, 14, 15, 22, 23, 59, 60, 61

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2020. 8, 18, 23, 24, 59, 60

[22] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Few-shot image classification: a good embedding is all you need. In *European Conference on Computer Vision (ECCV)*, 2020. 10, 42, 49, 50, 58, 66, 67

[23] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Neural Information Processing Systems (NeurIPS)*, 2018. 11, 27, 35, 38, 42, 49, 66

[24] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Neural Information Processing Systems (NeurIPS)*, 2018. 11, 27, 42, 58

[25] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *International Conference on Learning Representations (ICLR)*, 2018. 11, 12, 27, 36, 42, 49

[26] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations (ICLR)*, 2019. 12, 28, 42, 58

[27] Junsik Kim, Tae-Hyun Oh, Seokju Lee, Fei Pan, and In So Kweon. Variational prototyping-encoder: One-shot learning with prototypical images. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 12, 28, 42, 43, 58

[28] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 12, 28, 42, 58

[29] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 12, 28, 42, 58

[30] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to Compare: Relation network for few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 12, 28, 35, 37, 38, 42, 49, 51, 58, 66, 68, 84, 91, 97

[31] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *International Conference on Learning Representations (ICLR)*, 2020. 12, 28, 37, 42, 51, 58

[32] Davis Wertheimer and Bharath Hariharan. Few-shot learning with localization in realistic settings. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 12, 28, 42, 58

[33] Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. Variational few-shot learning. In *International Conference on Computer Vision (ICCV)*, 2019. 12, 28, 36, 42, 43, 49, 58

[34] Mamshad Nayeem Rizve, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 12, 58, 65, 67

[35] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997. 13, 14, 15

[36] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep Sets. In *Neural Information Processing Systems (NeurIPS)*, 2017. 14, 15, 59, 72, 78

[37] Akshay Mehrotra and Ambedkar Dukkipati. Generative adversarial residual pairwise networks for one shot learning. *arXiv preprint arXiv:1703.08033*, 2017. 15, 28, 42, 59

[38] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *International Conference on Computer Vision (ICCV)*, 2017. 15, 16, 28, 42, 43, 59

[39] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 15, 28, 29, 30, 36, 42, 59, 98

[40] Wen-Hsuan Chu, Yu-Jhe Li, Jing-Cheng Chang, and Yu-Chiang Frank Wang. Spot and learn: A maximum-entropy patch sampler for few-shot image classification. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 15, 28, 42, 59

[41] Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. In *Neural Information Processing Systems (NeurIPS)*, 2018. 15, 28, 42, 59

[42] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *International Conference on Computer Vision (ICCV)*, 2019. 15, 16, 28, 36, 42, 49, 59, 66

[43] Spyros Gidaris and Nikos Komodakis. Generating classification weights with GNN denoising autoencoders for few-shot learning. *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 15, 28, 36, 49, 59

[44] Bin Liu, Zhirong Wu, Han Hu, and Stephen Lin. Deep metric transfer for label propagation with limited annotated data. In *International Conference on Computer Vision (ICCV) Workshops*, 2019. 15, 16, 17, 28, 42, 59

[45] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *Neural Information Processing Systems (NeurIPS)*, 2018. 15, 28, 42, 59

[46] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 15, 16, 28, 38, 42, 43, 59

[47] Yu-Xiong Wang and Martial Hebert. Learning from small sample sets by combining unsupervised meta-training with CNNs. In *Neural Information Processing Systems (NeurIPS)*, 2016. 15, 16, 28, 42, 59

[48] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. MIXUP: Beyond empirical risk minimization. *International Conference on Learning Representations (ICLR)*, 2018. 15, 28, 42, 59

[49] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 15, 28, 42, 59

[50] Manli Zhang, Jianhong Zhang, Zhiwu Lu, Tao Xiang, Mingyu Ding, and Songfang Huang. IEPT: Instance-level and episode-level pretext tasks for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2020. 15, 42, 59

[51] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *Neural Information Processing Systems (NeurIPS)*, 2019. 16, 28

[52] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Neural Information Processing Systems (NeurIPS)*, 2019. 18, 19, 43

[53] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Neural Information Processing Systems (NeurIPS)*, 2017. 18, 19, 20, 43

[54] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Neural Information Processing Systems (NeurIPS)*, 2020. 18, 20, 21, 43

[55] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 18, 21, 22, 57, 58

[56] Nils Lid Hjort, Chris Holmes, Peter Müller, and Stephen G Walker. *Bayesian nonparametrics*. Cambridge University Press, 2010. 18, 43

[57] Carl Edward Rasmussen. The infinite Gaussian mixture model. In *Neural Information Processing Systems (NeurIPS)*, 2000. 18, 43

[58] Mike West and Michael D Escobar. *Hierarchical priors and mixture models, with application in regression and density estimation*. Institute of Statistics and Decision Sciences, Duke University, 1993. 18, 43

[59] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *arXiv preprint arXiv:1312.6114*, 2013. 19

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015. 21

[61] Ross Girshick. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*, 2015. 21

[62] 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 (ECCV)*, 2016. 21

[63] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008. 26, 41, 51, 52, 67, 69, 83, 94

[64] Fang Zhao, Jian Zhao, Shuicheng Yan, and Jiashi Feng. Dynamic conditional networks for few-shot learning. In *European Conference on Computer Vision (ECCV)*, 2018. 27

[65] Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *European Conference on Computer Vision (ECCV)*, 2018. 27

[66] Liang-Yan Gui, Yu-Xiong Wang, Deva Ramanan, and Jose M. F. Moura. Few-shot human motion prediction via meta-learning. In *European Conference on Computer Vision (ECCV)*, 2018. 27

[67] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *International Conference on Computer Vision (ICCV)*, 2019. 27, 42

[68] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 2002. 27, 42

[69] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *International Conference on Learning Representations (ICLR)*, 2020. 27, 36, 37, 42, 59

[70] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. TapNet: Neural network augmented with task-adaptive projection for few-shot learning. *International Conference on Machine Learning (ICML)*, 2019. 27, 36, 42, 49, 50, 58, 67

[71] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *International Conference on Learning Representations (ICLR)*, 2018. 28, 42, 58

[72] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision (ECCV)*, 2016. 28

[73] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 28, 36, 37, 49, 50, 66, 67

[74] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *International Conference on Computer Vision (ICCV)*, 2019. 28, 36, 37, 38, 49, 51, 68

[75] Joseph J Lim, Russ R Salakhutdinov, and Antonio Torralba. Transfer learning by borrowing examples for multiclass object detection. In *Neural Information Processing Systems (NeurIPS)*, 2011. 28, 29

[76] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Learning class prototypes via structure alignment for zero-shot recognition. In *European Conference on Computer Vision (ECCV)*, 2018. 29

[77] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and Cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 2009. 34

[78] Nathan Hilliard, Lawrence Phillips, Scott Howland, Artëm Yankov, Courtney D Corley, and Nathan O Hodas. Few-shot learning with metric-agnostic conditional embeddings. *arXiv preprint arXiv:1802.04376*, 2018. 34

[79] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 34, 48, 60, 61, 64

[80] Zagoruyko Sergey and Komodakis Nikos. Wide residual networks. In *British Machine Vision Conference (BMVC)*, 2016. 34, 48

[81] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *International Conference on Learning Representations (ICLR)*, 2018. 36, 49

[82] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L. Yuille. Few-shot image recognition by predicting parameters from activations. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 36, 49

[83] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 36, 37, 49, 50, 67

[84] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2006. 40

[85] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015. 40

[86] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 42

[87] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 42

[88] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M. Hospedales, and Tao Xiang. Incremental few-shot object detection. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 42

[89] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 42

[90] Yaxing Wang, Salman Khan, Abel Gonzalez-Garcia, Joost van de Weijer, and Fahad Shahbaz Khan. Semi-supervised learning for few-shot image-to-image translation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 42

[91] Ruibing Hou, Hong Chang, Bingpeng MA, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *Neural Information Processing Systems (NeurIPS)*, 2019. 42, 59

[92] Malik Boudiaf, Ziko Imtiaz Masud, Jérôme Rony, José Dolz, Pablo Piantanida, and Ismail Ben Ayed. Transductive information maximization for few-shot learning. In *Neural Information Processing Systems (NeurIPS)*, 2020. 42, 59

[93] Liu Jinlu, Song Liang, and Qin Yongqiang. Prototype rectification for few-shot learning. In *European Conference on Computer Vision (ECCV)*, 2020. 42

[94] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 42

[95] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2019. 42

[96] Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian regularized few-shot learning. In *International Conference on Machine Learning (ICML)*, 2020. 42

[97] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *International Conference on Computer Vision (ICCV)*, 2019. 42

[98] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. On modulating the gradient for meta-learning. In *European Conference on Computer Vision (ECCV)*, 2020. 42

[99] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. When does self-supervision improve few-shot learning? In *European Conference on Computer Vision (ECCV)*, 2020. 42

[100] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *European Conference on Computer Vision (ECCV)*, 2020. 42, 48, 49, 50, 51, 52, 66, 68, 92, 97

[101] Basura Fernando, Elisa Fromont, Damien Muselet, and Marc Sebban. Supervised learning of Gaussian mixture models for visual vocabulary generation. *Pattern Recognition*, 2012. 43

[102] J-L Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov Chains. *Transactions on Speech and Audio Processing*, 1994. 43

[103] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. Memory matching networks for one-shot image recognition. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 43, 49, 66

[104] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010. 48

[105] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 49, 50, 67

[106] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning (ICML)*, 2018. 49, 66

[107] Shell Xu Hu, Pablo G Moreno, Yang Xiao, Xi Shen, Guillaume Obozinski, Neil D Lawrence, and Andreas Damianou. Empirical Bayes transductive meta-learning with synthetic gradients. In *International Conference on Learning Representations (ICLR)*, 2020. 49

[108] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *Neural Information Processing Systems (NeurIPS)*, 2020. 49, 51, 68

[109] Junjie Li, Zilei Wang, and Xiaoming Hu. Learning intact features by erasing-inpainting for few-shot classification. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 35, 2021. 54

[110] Jiabao Zhao, Yifan Yang, Xin Lin, Jing Yang, and Liang He. Looking wider for better adaptive representation in few-shot learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021. 54

[111] Nanyi Fei, Zhiwu Lu, Tao Xiang, and Songfang Huang. MELR: Meta-learning via modeling episode-level relationships for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2021. 54, 64, 65, 66, 67, 68

[112] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *International Conference on Learning Representations (ICLR)*, 2019. 58

[113] Carl Doersch, Ankush Gupta, and Andrew Zisserman. CrossTransformers: spatially-aware few-shot transfer. In *Neural Information Processing Systems (NeurIPS)*, 2021. 58

[114] Weijian Xu, Huaijin Wang, Zhuowen Tu, et al. Attentional constellation nets for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2020. 59

[115] Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. TransMatch: A transfer-learning scheme for semi-supervised few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 59

[116] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. When does self-supervision improve few-shot learning? In *European Conference on Computer Vision (ECCV)*, 2020. 59

[117] Baoquan Zhang, Xutao Li, Yunming Ye, Zhichao Huang, and Lisai Zhang. Prototype completion with primitive knowledge for few-shot learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 59

[118] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CVT: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 59, 64

[119] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: A vision transformer in convnet's clothing for faster inference. In *International Conference on Computer Vision (ICCV)*, 2021. 59

[120] Ziyang Wu, Yuwei Li, Lihua Guo, and Kui Jia. PARN: Position-aware relation networks for few-shot learning. In *International Conference on Computer Vision (ICCV)*, 2019. 64

[121] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Mixture-based feature space learning for few-shot image classification. In *International Conference on Computer Vision (ICCV)*, 2021. 66, 67, 68, 75, 97

[122] Jaehoon Oh, Hyungjun Yoo, ChangHwan Kim, and Se-Young Yun. BOIL: Towards representation change for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2021. 66

[123] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-Baseline: Exploring simple meta-learning for few-shot learning. In *International Conference on Computer Vision (ICCV)*, 2021. 66

[124] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations (ICLR) Workshop*. Citeseer, 2014. 68

[125] Arman Afrasiyabi, Hugo Larochelle, Jean-François Lalonde, and Christian Gagné. Matching feature sets for few-shot image classification. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 76