Algorithms and feature preprocessing for transductive few-shot image classification

Yuqing Hu

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THESE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPERIEURE MINES-TELECOM ATLANTIQUE
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Signal, Image, Vision*

Par

Yuqing HU

Algorithms and Feature Preprocessing for Transductive Few-Shot Image Classification

Thèse présentée et soutenue à Rennes, le 6 décembre 2022
Unité de recherche : Lab-STICC, CNRS UMR 6285
Thèse N° : 2022IMTA0315

Rapporteurs avant soutenance :

Céline HUDELOT	Professeur, CentraleSupélec
Julyan ARBEL	Chargé de recherche, Inria Grenoble

Composition du Jury :

Président :	Sébastien LEFEVRE	Professeur, Université Bretagne Sud
Examinateurs :	Céline HUDELOT	Professeur, CentraleSupélec
	Julyan ARBEL	Chargé de recherche, Inria Grenoble
	Yannis AVRITHIS	Directeur de recherche, Athena Research Center
	Stéphane PATEUX	Ingénieur de recherche, Orange
Dir. de thèse :	Vincent GRIPON	Directeur de recherche, IMT Atlantique

Les travaux présentés dans cette thèse ont fait l'objet d'une convention CIFRE avec la société Orange SA. Je tiens à remercier particulièrement M. Stéphane PATEUX sans qui cette thèse n'aurait surement jamais vu le jour.

Résumé (French Summary)

Ce manuscrit contient la description de mon travail de thèse portant sur les algorithmes et les procédures de prétraitements des vecteurs caractéristiques pour la classification d'images à partir de peu de données étiquetées. Il s'agit d'un problème important pour la communauté de l'apprentissage automatique : depuis plusieurs années, les algorithmes proposés pour la vision par ordinateur sont parvenus à atteindre des niveaux de performance permettant d'envisager leur déploiement dans de nombreux contextes applicatifs [KSH17 ; Ian+16 ; Sze+15 ; He+16]. Toutefois, ce niveau de performance requiert typiquement d'énormes quantités de données étiquetées, lesquelles ne sont pas forcément disponibles dans certaines applications. Parvenir à maintenir un excellent niveau de performance tout en n'ayant accès qu'à un très faible volume d'étiquettes et de données est donc un verrou central de la discipline. Le document est organisé en six chapitres que nous résumons dans les paragraphes suivants.

Le **chapitre 1** est une introduction au contexte général de l'apprentissage automatique pour la vision et de la problématique de l'apprentissage avec peu de données étiquetées. Commençons par rappeler que l'apprentissage automatique est une discipline de l'informatique s'intéressant à doter les machines de compétences par des mécanismes d'apprentissage sur des données, par opposition à l'informatique "classique" dans laquelle une solution explicite au problème posé doit d'abord être conçue avant d'être programmée. Un problème canonique de la discipline est celui de la classification, où l'objectif est d'inférer une fonction f, typiquement définie sur un espace tensoriel le plus souvent muni d'une distribution de probabilités et à valeur dans un espace fini, à partir d'un nombre fini d'exemples $(\mathbf{x}, f(\mathbf{x}))$. Posé ainsi, le problème de classification est mal posé, et tout l'art des algorithmes proposés par la communauté de l'apprentissage automatique consiste à trouver des façons d'introduire des a priori sur la fonction f à trouver pour espérer y répondre.

Une autre façon de formaliser une telle fonction f est de remarquer qu'il s'agit d'un partitionnement de l'espace d'entrée, chaque partie étant typiquement appelée une "classe". Une façon simple de trouver une fonction f^V compatible avec les exemples fournis consiste donc à partitionner l'espace en cellules de Voronoï [AK00] définies à partir des $\mathbf{x}$ fournis. Chaque cellule prendra alors pour image celle du $\mathbf{x}$ la définissant. Cette façon de faire peut mener à des résultats intéressants en pratique, mais pose des soucis lorsque les données $\mathbf{x}$ sont tirées selon une distribution potentiellement bruitée, amenant à une fragmentation de la partition et à une fonction f^V potentiellement éloignée de f.

Il est commun de considérer comme modèle simplifié dans le cadre de l'apprentissage automatique le fait que chaque partie de la fonction f correspond à une distribution Gaussienne. Si ces Gaussiennes sont isotropes et de même écart-type, alors on peut adapter la technique précédemment expliquée en moyennant les observations $\mathbf{x}$ correspondant à une même partie avant de calculer les cellules de Voronoï correspondantes, amenant à de meilleurs performances en pratique face à ce type de distributions. L'algorithme correspondant s'appelle le classifieur du plus proche centroide (CPPC).

Lorsque les données sont plus complexes, par exemple si les distributions considérées pour chaque partie différent davantage que par leur simples moyennes, une autre solution possible est d'utiliser une régression logistique [Cox58 ; Cra02], une forme particulière de modèle linéaire généralisé cherchant à attribuer des probabilités d'appartenir à chaque classe pour chaque exemple $\mathbf{x}$ fourni. Une routine d'optimisation est alors mise en place pour trouver les paramètres qui collent le mieux aux données. Celle-ci consiste le plus souvent en une adaptation de l'algorithme de descente de gradient.

Ces modèles et algorithmes ont l'avantage de donner l'espoir de trouver de bons résultats y compris lorsque les données fournies sont peu nombreuses, notamment car les modèles sous-jacents dépendent typiquement de peu de paramètres. Mais dans les problèmes pratiques en vision, les distributions des classes sont le plus souvent très complexes et échappent totalement à ces modèles simplistes. C'est là qu'intervient l'apprentissage profond.

L'apprentissage profond consiste à apprendre des représentations (nous les appellerons des "vecteurs caractéristiques") permettant de transporter les données initiales, donc dans notre cas des images, dans un nouvel espace où elles suivent des distributions plus simples, adaptées aux modèles décrits précédemment. Ces représentations peuvent donc ensuite être utilisées en combinaison avec une régression logistique par exemple. Dans sa réalisation la plus commune, l'objectif est de rendre les vecteurs caractéristiques de chaque classe linéairement séparables.

Ces algorithmes mettent le plus souvent en place un assemblage, potentiellement très complexe, d'opérateurs simples appelés des "couches". Une couche consiste typiquement en la composition d'une fonction non-linéaire appliquée sur des tenseurs coordonnée par coordonnée avec une application tensorielle affine. La première contient typiquement très peu de paramètres, alors que la seconde un très grand nombre. En jouant sur l'assemblage des couches, et en trouvant les bons paramètres, il est possible d'atteindre de très bonnes performances sur des tâches de classification dans un grand nombre de domaines d'application. Dans le cadre de la vision par ordinateur, on utilise très souvent des réseaux convolutifs (RC) [LB+95], lesquels utilisent principalement des opérateurs de convolution pour les parties linéaires des couches. Les RC, hormis quelques considérations d'effets de bord et des détails sur la gestion de la résolution d'image, peuvent être invariants par translation, ce qui est souvent une propriété souhaitable pour les tâches de classification d'image. Un exemple d'architecture de RC très utilisé dans la littérature (et dans nos travaux) est celui du ResNet [He+16 ; Ye+20], lequel est décrit en détails dans le chapitre 1.

Ces architectures RC (ou des équivalents pour d'autres domaines) ont permis

d'atteindre des performances remarquables dans un grand nombre de problèmes de l'apprentissage automatique. Toutefois, atteindre les meilleures performances requiert souvent d'utiliser beaucoup de puissance de calcul, comme le montre la figure 1. Et cette dépendance à une très forte quantité de calculs s'explique par le fait que les modèles correspondants contiennent une quantité immense de paramètres, laquelle peut se mesurer en milliards.

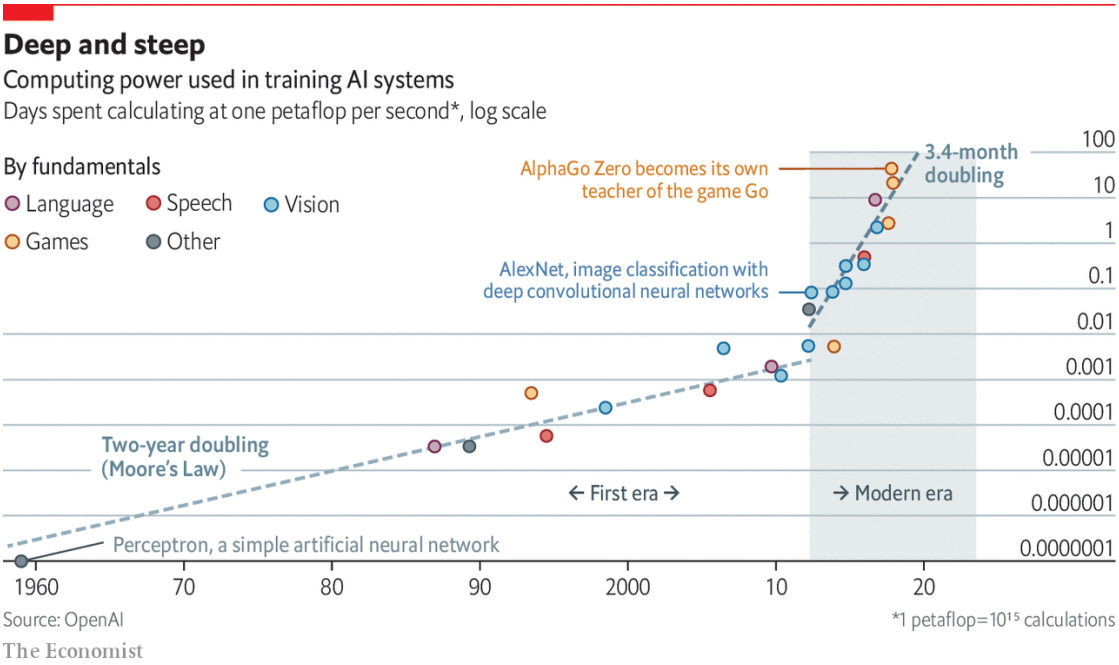


Figure 1: Évolution de la complexité de calcul requise pour les performances SOTA en IA. Source: `https://www.economist.com/technology-quarterly/2020/06/11/the-cost-of-training-machines-is-becoming-a-problem`.

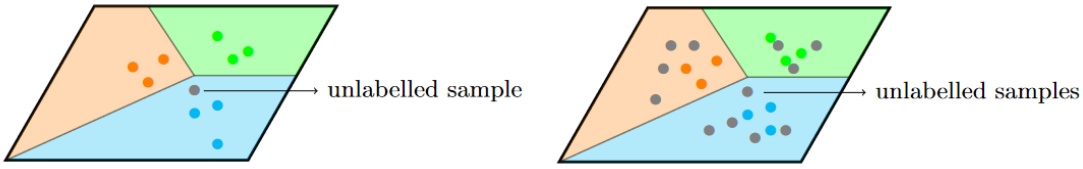
Évidemment, il n'est pas possible d'espérer trouver de bonnes valeurs pour ces milliards de paramètres avec seulement quelques données d'entraînement $\mathbf{x}$. C'est pourquoi le domaine de l'apprentissage avec peu de données étiquetées consiste à trouver des façons d'utiliser d'autres jeux de données, potentiellement beaucoup plus fournis, pour aider à trouver de bons paramètres dans l'architecture considérée, même si le problème considéré par ces autres jeux de données est potentiellement très différent de celui d'intérêt. On appelle ce procédé un "transfert d'apprentissage".

Il convient de distinguer deux cas bien différents d'apprentissage avec peu d'exemples.

- Dans le cas *inductif*, l'objectif est d'apprendre avec très peu de couples $(\mathbf{x}, f(\mathbf{x}))$;
- Dans le cas *transductif*, l'objectif est d'apprendre avec très peu de couples $(\mathbf{x}, f(\mathbf{x}))$ et de prédire sur $\{\mathbf{x}'\}$, ces derniers étant disponibles dès le départ et pouvant donc être exploités.

Le cas inductif correspond typiquement à des scénarios ou l'acquisition des données est le réactif limitant. Le cas transductif est rencontré lorsque l'annotation de ces données est le principal soucis, ce qui arrive dans des cas applicatifs où l'annotation

est particulièrement coûteuse ou que les événements d'intérêts sont rares. Dans le cadre de mes études, c'est bien le cas transductif auquel je me suis intéressé.

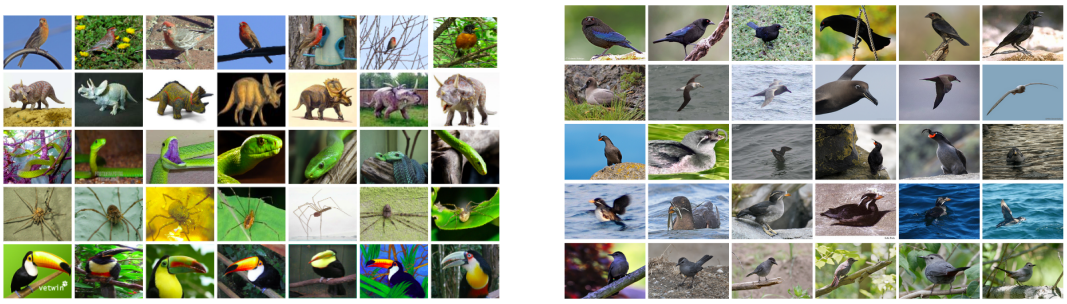


(a) Inductive settings **(b)** Transductive settings

Figure 2: Illustration du setting inductif et transductif.

Le **chapitre 2** décrit avec de plus amples détails la façon dont l'état de l'art aborde cette question de l'apprentissage automatique dans un contexte de classification avec peu de données étiquetées (donc classification transductive).

Nous prenons le temps de décrire avec précision la façon dont les auteurs se comparent par l'intermédiaire de bancs d'essais standardisés. Ces derniers consistent le plus souvent à fixer trois jeux de données : un jeu de données générique pouvant servir à préentraîner des RC ou d'autres modèles et contenant beaucoup de données étiquetées, un jeu de données de validation, distinct par ses classes du jeu de données générique, et permettant de tester la capacité des modèles préentraînés à s'adapter à de nouvelles classes, et un jeu de données de test, à partir duquel des milliers de problèmes artificiels d'apprentissage avec peu de données étiquetées sont générés aléatoirement, pour obtenir une performance moyenne de la méthode considérée.



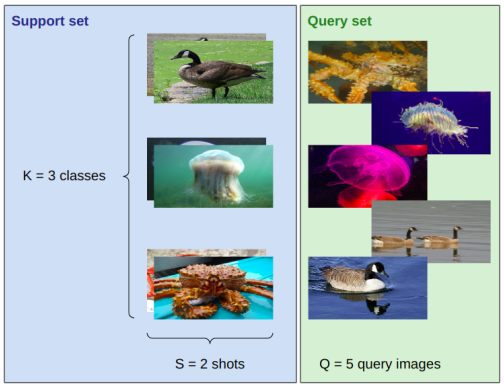
(a) *mini*-ImageNet **(b)** CUB

Figure 3: Exemples d'images de benchmarks few-shot (*mini*-ImageNet et CUB).

Les bancs d'essais standardisés dans le domaine comprennent *mini*-ImageNet [Rus+15], *tiered*-ImageNet [Ren+18], CUB [Wah+11], FC100 [ORLL18] et CIFAR-FS [Ber+19]. Des exemples illustratifs sont présentés dans la figure 3, et une illustration plus particulière d'un problème d'apprentissage avec peu d'exemples est présenté dans la figure 4.

Si la littérature s'est d'abord concentrée sur des générations de problèmes d'apprentissage avec peu de données peu hétérogènes (toujours le même nombre de classes et d'exemples étiquetés pour chaque classe) [Vin+16 ; SSZ17 ; Sun+18 ; SE18 ;

(a) Example d'un ensemble de données d'entraînement



(b) Example d'un ensemble de données à tester

Figure 4: Exemple d'un scénario de la Classification Few-Shot.

Kim+19 ; GK19 ; Liu+19], plus récemment des contributions ont souligné le besoin de considérer des scénarios plus diversifiés [Vei+21 ; Tri+20]. Une partie importante de mes travaux de thèse ont été réalisés sur ces nouvelles façons de générer les problèmes.

Dans la suite du chapitre, nous proposons une vision abstraite générale permettant de décrire la plupart des méthodes proposées dans le domaine. Cette vision se découpe en trois étapes :

1. Le préentrainement, à l'aide des jeux de données génériques et de validation, d'un extracteur de vecteurs caractéristiques, le plus souvent un RC;
2. L'utilisation de routines de transformations des vecteurs caractéristiques, ayant pour objectif de les rendre plus faciles à manipuler;
3. Le déploiement d'algorithmes d'apprentissage automatique sur les vecteurs transformés. Ces algorithmes utilisent à la fois les données étiquetées et les données sur lesquelles la prédiction doit être faite, on dit qu'ils sont semi-supervisés.

Les contributions de ma thèse touchent à ces trois étapes, résumées dans la figure 5.

Pour la première étape, plusieurs solutions ont été proposées dans la littérature. Certaines s'inspirent du principe du méta-apprentissage [Vin+16 ; SSZ17 ; FAL17 ; RL17 ; Sun+18 ; Li+19], c'est-à-dire apprendre à apprendre avec peu de données. Si ces techniques bénéficient d'une grande popularité, elles obtiennent typiquement de moins bonnes performances sur les bancs d'essais standardisés. L'autre solution qui nous intéresse davantage dans le cadre de mes travaux de thèse consiste à utiliser différents types de régularisations pendant l'apprentissage du RC extracteur de caractéristiques ayant pour principal objectif d'améliorer la performance quand il est déployé sur de nouvelles classes. Ces techniques incluent de la régularisation par interpolations linéaires (*mixup* [Zha+17]), des tâches prétextes (par exemple des rotations artificielles des images d'entrées [GSK18a]), de la distillation [ZS20 ; Yua+20 ; Tia+20 ; Riz+21], de l'auto-supervision [Man+20 ; MSN21 ; Rod+20 ; Ma+21 ; Kho+20] et bien d'autres... Par abus de langage, on qualifie souvent cette

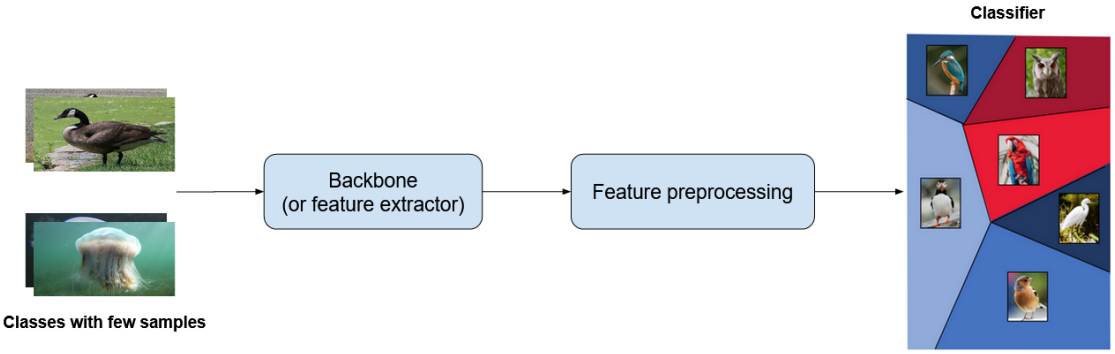


Figure 5: Illustration du pipeline général de la Classification Few-Shot. Source: `https://medium.com/sap-machine-learning-research/deep-few-shot-learning-a1caa289f18`.

seconde façon de faire d'apprentissage par transfert, bien que le terme pourrait s'appliquer aussi à la première.

Pour la seconde étape, les auteurs ont proposé un nombre important d'opérations de normalisation et de prétraitements des vecteurs caractéristiques [Wan+19b ; Lic+20 ; Wan+19b], le plus souvent sans justification théorique, pour améliorer les performances du système entier. On peut globalement décrire l'objectif de ces opérations comme cherchant à transformer les distributions des vecteurs caractéristiques, le plus souvent difficiles à modéliser, en des distributions quasi-Gaussiennes.

Enfin, pour la troisième étape, deux types de méthodes émergent principalement : 1) les méthodes de régression logistique [Che+19a ; Man+20 ; Bou+20a] ; et 2) les méthodes de partitionnement basée sur des métriques [Wan+19b ; Ren+18 ; Lic+20 ; Bat+22 ; HLLJ19], comme l'algorithme de classification par le plus proche centroide précédemment décrit, et illustré figure 6. Nous décrivons en détails plusieurs de ces méthodes qui auront joué le rôle de compétiteurs à nos propres propositions pendant le travail de ma thèse, notamment l'algorithme soft-KMEANS.

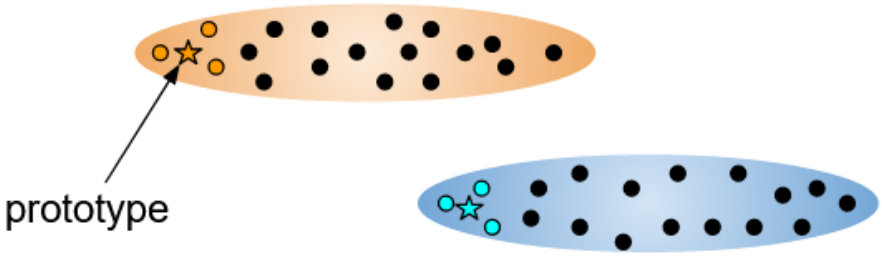


Figure 6: Illustration d'un classificateur CPPC.

Les trois chapitres suivants mettent en lumière plusieurs contributions importantes de ma thèse, principalement liées aux étapes 2 et 3 précédemment introduites.

En ce qui concerne l'étape de prétraitement des vecteurs caractéristiques, deux questions sont importantes. La première est de trouver des prétraitements améliorant le rapport signal à bruit des représentations obtenues, c'est-à-dire contribuant à rendre les distributions associées à différentes classes plus facilement séparables.

La seconde est de s'assurer que les distributions obtenues prennent des formes similaires à des distributions Gaussiennes, car l'hypothèse de distributions Gaussiennes est souvent le prérequis des algorithmes de classification en apprentissage avec peu de données étiquetées.

Pour répondre à la première question, nous proposons dans ce manuscrit une méthode basée sur le traitement de signaux sur graphes, présentée dans le chapitre 3. Pour répondre à la deuxième question, nous proposons d'appliquer une technique appelée transformation de puissance qui peut aider à augmenter de manière significative les performances en remodelant les distributions des vecteurs caractéristiques. Cette technique sera présentée en détail dans le chapitre 4.

En termes de contributions sur les algorithmes de classification semi-supervisée, nous présentons dans le chapitre 4 une méthode de partitionnement s'inspirant de la théorie sur le transport optimal [Vil09]. L'idée ici est d'utiliser des estimations temporaires de prédiction sur les données non-étiquetées pour mieux estimer les centres de nos parties. L'algorithme proposé a permis de tenir la première place sur les bancs d'essais standardisés pendant de nombreux mois et est la contribution de ma thèse qui a eu le plus fort impact en terme de citations [LSA21 ; CVK21 ; Ort+21 ; ZK22]. La stratégie développée a été critiquée à juste titres par certains auteurs [Vei+21] car elle exploite explicitement la notion d'équidistribution des données non-étiquetées entre les classes considérées, laquelle était présente dans la plupart des bancs d'essais de l'époque, bien qu'il s'agisse a priori d'une supposition peu réaliste.

Dans le chapitre 5, nous proposons un autre algorithme de partitionnement s'appuyant sur l'inférence bayésienne variationnelle [FR12 ; WBJ05 ; CB01 ; BN06] et la réduction adaptative des dimensions pour aligner et estimer au mieux les centres des parties. Contrairement à la contribution précédente, l'algorithme proposé ne nécessite aucune information préalable sur l'ensemble des données non-étiquetées, et atteint des performances de pointe dans le cas où les bancs d'essais génèrent des problèmes plus diversifiés.

La contribution du chapitre **chapitre 3** correspond à l'article:

Graph-based interpolation of feature vectors for accurate few-shot classification **Hu, Y.**, Gripon, V. and Pateux, S., 2021, January. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 8164-8171). IEEE.2020

Dans ce chapitre, nous en profitons pour ajouter des discussions supplémentaires sur la méthode proposée. L'idée ici est d'utiliser des outils tirés du traitement de signaux sur graphes, lesquels ont suscité un intérêt croissant en raison de leur capacité à capturer les relations entre les échantillons selon des métriques de similarité. Dans le cadre de la classification d'images avec quelques exemples, notamment dans des contextes transductifs, ces méthodes ont déjà été appliquées avec succès dans plusieurs travaux tels que [Che+21a ; Kim+19 ; SE18]. Cependant, ces travaux utilisent ces techniques directement dans l'étape d'extraction de vecteurs caractéristiques, réduisant de fait la performance atteignable avec de simples RC bien entraînés. Au contraire, dans notre approche, nous utilisons ces techniques uniquement dans l'étape de prétraitement des vecteurs caractéristiques, permettant

de mieux bénéficier des apports respectifs des RC et des outils du traitement de signaux sur graphes. La méthode que nous proposons est considérée comme l'une des premières à utiliser le graphe pour prétraiter les vecteurs caractéristiques dans un contexte d'apprentissage avec peu d'exemples annotés et a apporté une augmentation significative de la performance par rapport à l'état de l'art de l'époque.

Cependant, l'un des principaux inconvénients de la méthode proposée est qu'elle nécessite de concevoir soigneusement un graphe pour obtenir les meilleurs gains de précision. Dans l'article, la construction du graphe est accompagnée de trois hyperparamètres dédiés, et leur réglage dans la pratique peut être difficile en raison de l'absence d'un ensemble de validation sur la tâche considérée. Cet inconvénient est atténué par le fait que nos expériences montrent que le réglage des hyperparamètres n'est pas très sensible. Afin de contourner cette limitation, une solution possible serait de trouver les meilleurs hyperparamètres en utilisant des tâches similaires à celle considérée.

Dans le **chapitre 4**, nous présentons une combinaison de deux articles [HGP21b ; HSS18] contribuant aux étapes de prétraitement des vecteurs caractéristiques et de conception d'un algorithme de classification semi-supervisée. Nous présentons le contexte général et l'article, puis nous discutons les contributions de notre méthode proposée ainsi que les limites et les perspectives. Les articles discutés sont :

Leveraging the Feature Distribution in Transfer-Based Few-Shot Learning **Hu, Y.**, Gripon, V. and Pateux, S., 2021, September. In International Conference on Artificial Neural Networks (pp. 487-499). Springer, Cham.2021

Squeezing Backbone Feature Distributions to the Max for Efficient Few-Shot Learning **Hu, Y.**, Pateux, S. and Gripon, V., 2022. Algorithms, 15(5), p.147.2022

Outre le prétraitement des vecteurs caractéristiques, un bon modèle de prédiction exige également que l'algorithme de classification soit bien conçu. Les algorithmes populaires incluent les méthodes de régression logistique [Dhi+20 ; Man+20] qui apprennent les paramètres de frontière de décision avec une perte minimisée, et les méthodes de partitionnement telles que Kmeans [HW79] qui effectuent des prédictions via des centroides de classe typiquement estimés à partir d'hypothèses de Gaussianité. Étant donné le contexte transductif de mes travaux, nous proposons dans ces contributions deux choses : 1) utiliser la transformation de puissance dans le cadre du prétraitement des vecteurs caractéristiques afin de les aligner avec des hypothèses gaussiennes, et 2) la construction d'un algorithme s'appuyant sur le transport optimal (OT) [Vil08] qui fonctionne dans un cadre d'espérance-maximisation (EM). Avec l'aide de 1) et 2), notre méthode proposée "PT+MAP" a obtenu des performances de pointe sur différents bancs d'essais.

Bien que l'algorithme de classification soit capable d'apporter des gains importants avec l'aide du transport optimal, un inconvénient majeur de l'algorithme est son exigence pour la distribution équilibrée des échantillons non étiquetés, hypothèse peu réaliste en pratique. C'est pourquoi, afin de résoudre ce problème, dans notre travail "Squeezing Backbone Feature Distributions to the Max for Efficient Few-Shot Learning", qui peut également être considéré comme une version étendue du

travail précédent, nous proposons une version modifiée de l'algorithme qui tente de prendre en compte une incertitude sur le nombre d'échantillons non-étiquetés dans chaque classe considérée. Nous intégrons également une adaptation de la régression logistique dans le cadre de l'EM, ajustant davantage les centroïdes de classe sur la base des pseudo-étiquettes des échantillons non-étiquetés. À cette fin, notre algorithme nouvellement modifié est capable d'obtenir des résultats encore améliorés par rapport à PT+MAP.

Dans notre première tentative d'aborder le problème du déséquilibre, nous avons proposé dans ce travail une version modifiée d'OT pour essayer de réduire l'effet des a prioris en boostant uniquement les classes qui ont le moins de poids [Lic+20]. Cependant, sans connaître la proportion exacte d'échantillons non-étiquetés par rapport aux classes, l'algorithme a tendance à égaliser ces échantillons non étiquetés et entraîne donc toujours une baisse significative de la précision dans un cadre très déséquilibré.

Dans le **chapitre 5**, nous continuons l'amélioration de la conception de l'algorithme de classification semi-supervisée. L'article présenté est le suivant :

Adaptive Dimension Reduction and Variational Inference for Transductive Few-Shot Classification Hu, Y., Pateux, S. and Gripon, V., Arxiv preprint.2022

Bien que des travaux antérieurs tentent d'aborder le problème des contraintes préalables par rapport aux échantillons non étiquetés, la performance de ces derniers n'est toujours pas idéale face au cadre déséquilibré proposé dans [Vei+21], ce qui suggère la limitation de l'OT sans estimation préalable. C'est pourquoi, dans ce travail intitulé "Adaptive Dimension Reduction and Variational Inference for Transductive Few-Shot Classification", nous proposons une nouvelle méthode s'appuyant sur l'inférence bayésienne variationnelle, ainsi qu'une réduction dimensionnelle adaptative qui utilise l'analyse discriminante linéaire probabiliste pour projeter itérativement les données dans des dimensions inférieures afin de prédire les étiquettes. La méthode proposée, appelée "BAVARDAGE", est capable d'atteindre des performances de pointe dans le cadre non équilibré, et des résultats compétitifs dans le cadre équilibré sans aucune connaissance préalable.

Cependant, les solutions proposées s'accompagnent d'un certain nombre d'hyperparamètres, dont certains sont difficiles à régler sans avoir accès à un ensemble de tâches de validation pertinent. Ce problème récurrent de gagner quelques pourcents de précision au prix de l'ajout d'hyperparamètres pourrait être au cœur des discussions dans le domaine, car il est plus problématique qu'avec la classification standard où les ensembles de validation permettent le réglage de ces hyperparamètres. La tendance récente vers une évaluation plus diversifiée dans les bancs d'essais standardisés, notamment avec l'essor de Metadataset [Tri+20], est certainement un pas dans la bonne direction.

Enfin, dans le **chapitre 6**, nous exposons les conclusions de ce manuscrit, en réaffirmant les contributions sur les étapes de prétraitement des vecteurs caractéristiques et de conception de l'algorithme de classification. Nous discutons aussi de ce que nous pensons être des directions importantes pour le futur de la discipline.

Nous mentionnons ainsi l'évolution des pratiques sur les résolutions des images considérées : habituellement à 84x84 pixels, elles ont évolué dernièrement vers des formats plus grands, par exemple du 224x224, permettant de mieux capturer des motifs précis et très localisés. De fait, la performance des systèmes d'apprentissage avec peu d'images annotées s'améliore [Che+21b ; Luo+21], même si les techniques restent identiques. La littérature s'intéresse de plus en plus à la question du choix du jeu de données générique, mais aussi à la façon de le découper : il est assez clair que certains jeux de données génériques sont adaptés pour un transfert vers certaines tâches d'apprentissage avec peu de données annotées, mais d'autres non. Trouver automatiquement quel jeu de données générique utiliser est donc une question d'importance pour l'avenir de la discipline[SCA20 ; Laf+22 ; Ben+22c]. L'utilisation de données supplémentaires [Xin+19 ; Sch+19 ; Zha+21 ; Che+21b ; Bat+22], par exemple des données non-annotées traitées de façon auto-supervisée [PH21 ; Isl+21] est aussi une direction importante pour la recherche à venir, étant donné l'importance que prennent ces techniques dans des contextes non contraints par la quantité de données disponibles. D'autres pistes, incluant l'apprentissage actif [BI17 ; PZS20 ; Mül+22 ; Li+22], la désambiguisation lors de la présence de plusieurs objets dans la scène [Ben+22a], ou encore la prise en compte de la sémantique sont également mentionnées.

Contents

Acknowledgments

I would like to show my deepest appreciation to my supervisors Stéphane Pateux and Vincent Gripon for their invaluable effort, patience and feedback over the course of my thesis. Words can not express my gratitude to them for generously providing their knowledge and expertise. I am also deeply indebted to my defense committee: Stéphane Pateux, Vincent Gripon, Céline Hudelot, Yannis Avrithis, Julyan Arbel and Sébastien Lefèvre, who kindly accepted to be the jury members to examine my work.

In addition, this journey could not have been undertaken without my follow-up committee Teddy Furon and Matthijs Douze, who ensure the smooth running of the programme on the basis of the doctoral charter ("charte du doctorat") and the training agreement ("convention de formation"). Moreover, this endeavor would not have been possible without the generous support from Orange through the CIFRE thesis contract 2019/1863, who generously financed my research.

I am also thankful to my colleagues at Orange and IMT Atlantique for their constant assistance on the equipment and administrative procedures. Additionally, I would like to acknowledge their editing help, incisive feedback and moral support that have motivated and inspired me.

Lastly, I would be remiss in not mentioning my family, especially my parents, girlfriend, and friends. Their belief in me has kept my spirits and motivation high during this process.

Chapter 1

Introduction

In this chapter we firstly present the context of my thesis, starting by the development of Machine Learning and Deep Learning over the past years. And we introduce classification task along with some well-known methods ranging from basic logistic regression to convolutional neural networks. Then we present the problematic of this thesis that we seek to address, namely to perform image classification with few labeled data. Finally we present our contributions during the three years of my PhD.

1.1 Scientific context

1.1.1 Machine Learning and Deep Learning

Within the past decade, there have been a growing interest for machine learning and deep learning in particular. This success can be explained in part by the resolution of old open problems in many different domains [LBH15; BLH21], including vision [KSH17; Ian+16; Sze+15; He+16], natural language processing [Vas+17; Dev+18; Dev+18], games [Sil+16; Sil+17], audio [Gem+17] and even more recently biology [Jum+21].

Contrary to classical computer science, machine learning does not require an explicit solution to the considered problem, but can infer one instead, given enough data/observations. With the growing availability of large data sources, notably thanks to Internet, machine learning takes an increasingly important role within automation in society.

There are in particular two settings where machine learning is the gold standard: 1) when there is no explicit solution available to the considered problem, and 2) when it would be too costly to implement such a solution, or it would require too much computational complexity.

A very common example of a problem that falls into category 1) is that of recognizing complex objects (e.g. persons or animals) in images under diverse conditions (e.g. exposition, orientation, image quality, etc). Indeed, the only system that is able to solve this problem is the brain of animals, and its functioning is still not fully

understood, even when it comes to only the visual parts of it. As a consequence, machine learning is the only option available to reach human level performance on such tasks.

As far as category 2) is concerned, a classical example is that of playing the game of Go, which is known to be challenging even for modern hardware. In this case, there exists explicit solutions based on tree search algorithms such as Minimax [VNM07; Sto79] that would lead to an optimal way of playing the game, but those would require way too much computational complexity to be reasonably implemented. Here again, machine learning appears to be the only viable option.

However, a major drawback of machine learning approaches is that they typically consist of inferring a function, let us denote it f, based on a finite number of observations $(\mathbf{x}, f(\mathbf{x}))$. This problem is often referred to as "supervised learning", where supervised means that we have access to examples of expected outputs in our considered task. When described this way, machine learning consists of an extrapolation problem, and consequently is in general ill-posed. As a matter of fact, it is often that the input space of the searched function f is infinite (or at least unreasonably large). Without any other prior about f, there are thus infinitely many possibilities that agree with the given examples. Finding which one is a good extrapolation is then impossible, at least from a mathematical perspective.

To circumvent this problem, it is very common in machine learning to consider a restrained family of functions in which our solution is to be found. As a consequence, there can in some cases be only one extrapolation to the given examples or even none, in which case, we typically look for the function that agrees the most with the given examples. This situation can seem paradoxical, and is often referred to as the bias/variance trade-off in the machine learning literature [BN06].

There are many such families of functions, and presenting all of them is out of scope of the current document. Instead, we will focus on very specific such families that are commonly used in modern machine learning and that are at the root of the contributions in this manuscript.

In the scope of this document, we are interested in a specific sort of supervised learning where outputs are categorical. They can only take a finite (and typically small) number of different values, and each categorical value represents the class label of the associated observation. For instance, for a model that differentiates whether an image is of a cat or a dog, the output of f would be whether of the value 0 or 1 where 0 represents the category dog and 1 for cat.

We call such a problem a classification problem, which appears in many practical applications ranging from predicting the objects present in an input image, to diagnosis of certain diseases from medical data or even recognizing sounds. Therefore, supervised learning in classification consists in learning f that maps the inputs to the labels. In the next sections we present differents classes of supervised learning methods for classification.

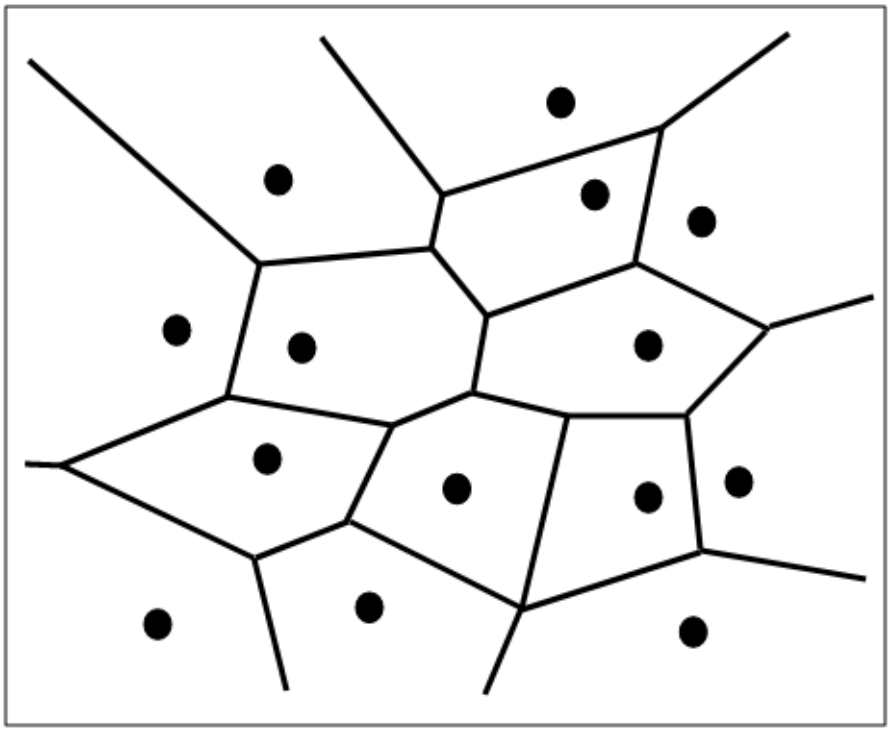


Figure 7: Voronoi diagram on a 2D plane.

1.1.2 Metric-based methods

In metric-based methods, the key idea is to use the given examples as anchors defining areas of influence to define the searched function f elsewhere in the input space. The most celebrated such method is called "nearest neighbor classification", and it consists in partitioning the input space depending on the closest given example, each part being mapped to the output of this specific example.

This has the effect of creating so-called Voronoi Cells [AK00], which have interesting mathematical properties. Among others, Voronoi Cells can approximate any function (under mild hypothesis) given enough examples. Typically, the Euclidean distance is used if working with a metric space, but other metrics can be found in the literature. Fig. 7 is an illustration of creating Voronoi Cells, in which each cell is created based on the labeled data point in black.

A simple extension to nearest neighbor classification is nearest class mean classification (NCM) in which the possibly multiple examples that belong to the same class are first averaged before creating the Voronoi Cells, resulting in a single connected part of space for each considered class. In problems where the inputs can be noisy, this solution can lead to more robust partitions than nearest neighbor classification.

1.1.3 Logistic regression

Contrary to nearest neighbor classification that only predicts class labels, logistic regression [Cox58; Cra02] can derive confidence level about its prediction. Namely, it is characterized by outputting the probability of a data point belonging to each class, and the probability is computed based on an active function apply on linear

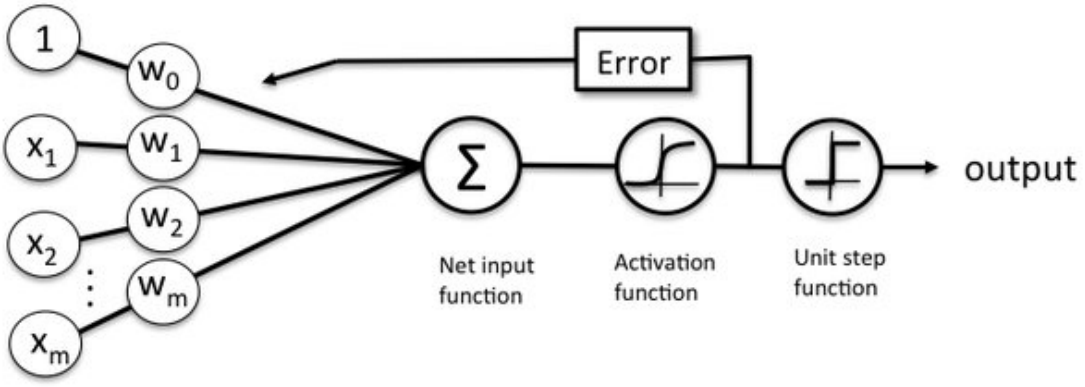


Figure 8: Illustration of logistic regression. Source: `http://rasbt.github.io/mlxtend/user_guide/classifier/LogisticRegression/`.

regression [Wei05; SL12; MPV21]. The method requires a training process with labeled examples to learn its parameters for class prediction.

Fig. 8 illustrates the logistic regression algorithm for binary classification. In detail, we consider a vector $\mathbf{w} = [w_0, w_1, ..., w_m]$ containing parameters to be trained, along with input features $\mathbf{x} = [x_0, x_1, ..., x_m]$ we compute the logit (or log odds ratios) to be the dot product between these two vectors:

$$z = w_0 x_0 + w_1 x_1 + ... + w_m x_m = \mathbf{w}^T \mathbf{x}. \tag{1}$$

Note that here w_0 refers to the bias (also denoted as b in many works) and x_0, which always equals to 1, is an additional variable that introduces the bias.

We notice that Eq. 1 corresponds to a linear regression model where z is a continuous value ranging from $-\infty$ to $+\infty$. However, when it comes to binomial classification (with class 0 and 1), we need a boundary between the values that are classified as 0 or 1. Therefore, linear regression may not be feasible as there is no boundary to its value. To address that challenge, logistic regression adds an activation function on top of the logit to map it to 0 or 1, making the model a preferable choice compared to linear regression. And the activation function (or logistic regression function) is defined as follows:

$$\phi(z) = \frac{1}{1 + e^{-z}}. \tag{2}$$

It is called a Sigmoid function [HM95; Nar97; Mar+08] and has been used in binary classification to convert logits to probabilities. Fig. 9 illustrates the curve of the Sigmoid function, we can observe that the function outputs values between 0 and 1 that can be interpreted as probabilities of class belongings. Depending on the threshold (usually predefined to be 0.5), we can make label predictions for novel observations and evaluate the performance.

The training process in Logistic regression aims at estimating the parameters $\mathbf{w}$ that allow a good classification. To perform that, the algorithm trains $\mathbf{w}$ so that

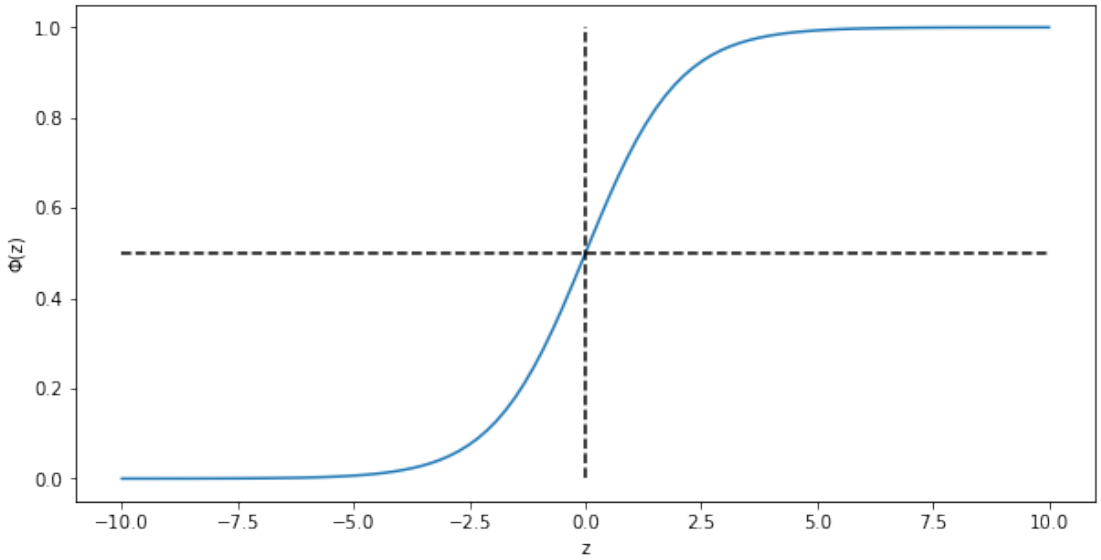


Figure 9: Illustration of the Sigmoid function.

the error would be minimized. The training process in logistic regression requires 1) observations $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ where y_i indicates the class label of $\mathbf{x}_i$; 2) an objective function that computes the loss between labels and predictions and 3) An optimizer that minimizes the cost function. For 1), the training process involves all the labeled data for loss calculation. As for the objective function (or loss function), oftentimes we define it according to the task, in the case of binary classification we use the binary Cross-Entropy loss function that can be defined as follows:

$$l = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\phi(z_i)) - (1 - y_i)\log(1 - \phi(z_i)), \tag{3}$$

where $z_i = \mathbf{w}^T\mathbf{x}_i$ is the logit value for observation $\mathbf{x}_i$, along with $\phi(z_i)$ the probability for the corresponding class. We can see that a wrongly predicted sample (high $\phi(z_i)$ under $y_i = 0$ or low $\phi(z_i)$ under $y_i = 1$) would result in a high value of loss. Therefore, the objective of training is to minimize the loss function so that samples are correctly labeled.

From Eq. 3 we notice that logistic regression fits into the Maximum Likelihood Estimation (MLE) framework, in which the goal is to maximize the conditional probability of observing the data given a specific probability distribution and its parameters $\sum_{i=1}^{N} \log P(\mathbf{x}_i; \mathbf{w})$, assuming the independence of observations. Therefore, supervised learning can be framed as a conditional probability problem of predicting the probability of the output given the input $\sum_{i=1}^{N} \log P(y_i|\mathbf{x}_i; \mathbf{w})$. In the case of logistic regression for binary classification, we assume a Binomial probability distribution for the observations, and the likelihood function corresponds to Eq. 3 in the negative form (maximizing the likelihood is equivalent to minimizing the loss function).

The way we train a logistic regression model to its optimum is through an optimizer,

the goal of which is to reduce loss using gradient descent that requires the gradient of the cost function. Therefore in logistic regression, we minimize the loss by searching in the direction that corresponds to the negative partial derivative of the cost function with respect to the parameter $\mathbf{w}$ (Eq. 4):

$$\begin{aligned} \text{Objective :} \quad & \min_{\mathbf{w}} l, \\ \text{Partial derivative :} \quad & \frac{\partial l}{\partial \mathbf{w}} = \frac{1}{N} \sum_{i=1}^{N} (\phi(z_i) - y_i)\mathbf{x}_i. \end{aligned} \tag{4}$$

During training, gradient descent will iterate along the negative gradient direction of $\mathbf{w}$ until reaching convergence where the model parameters become stable. The basic form of the training process in logistic regression is described in Eq. 5:

$$\begin{aligned} &\text{Iterate} \quad e \quad \text{epochs :} \\ &\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{1}{N} \sum_{i=1}^{N} (\phi(z_i) - y_i)\mathbf{x}_i, \end{aligned} \tag{5}$$

where an epoch denotes a training iteration in which the model is learned through a complete pass of the training data, and η is called the learning rate or the search step that has to be carefully chosen in order for the model to converge.

There exists several optimizers to train a model such as SGD [Bot10], Adam [KB15] and so on. Apart from their commonality on the use of gradient descent, they differ mainly in terms of their strategies to find the minimal loss, for example the use of weight decay and the choice of learning rate.

Weight decay is a well-known strategy to prevent the model from becoming overly complex (also called overfitting), the idea is to penalize model complexity by e.g. adding the square of all training parameters to the cost function so that some parameters that may contribute to overfitting would be dialed down to much smaller values or 0. There also exist other forms for the added term depending on the metric, for instance in [MVDGB08] the authors propose to add the absolute value of all training parameters. The type of methods is called "Lasso" and is also well-known for its effectiveness in reducing overfitting.

As for the learning rate, there exists several techniques operated under a learning rate schedule [Ben12]. For example in a basic scheduler the learning rate is constant regardless of the training epochs, and a multi-step scheduler [Sen+13] decays the learning rate with a certain multiplicative factor when a certain number of epochs is reached. More recently, a cosine annealing scheduler [LH16] is applied where the training starts off with a very large learning rate and then aggressively decreases it to a value near 0, before again increasing the learning rate. This variation of the learning rate happens according to the cosine annealing schedule.

In summary, logistic regression is a training-based method that attempts to find parameters that minimize the prediction loss. The algorithm requires parameters to train and predict probabilities instead of hard labels. However, due to the fact that the output in Eq. 1 is the sum of input features and parameters, logistic regression

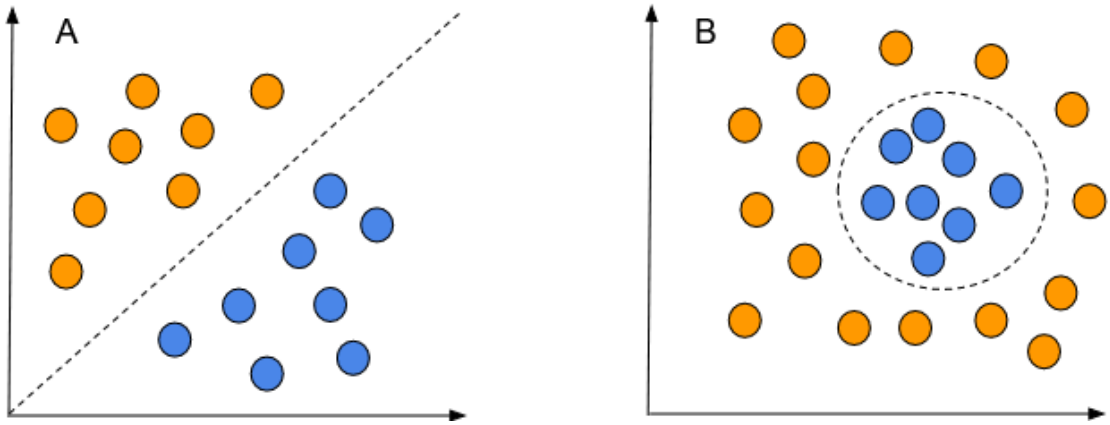


Figure 10: Linearly separable samples (A) vs. non-linearly separable samples (B).

is thus considered a linear model that works only for data points that are linearly separable for classification (illustrated in Fig. 10.A), whereas in metric-based methods such as nearest neighbors classification there is no such requirement.

With all of that said, the training process in logistic regression and its corresponding strategies are similar of those used in Deep Neural Networks. In fact, logistic regression can be considered as a one-layer neural network that only learns the parameters for the classifier. As for DNNs, we will introduce in detail in the next sections.

1.1.4 From Multi-Layer Perceptron to Convolutional NNs

Rather than using a logistic regression directly on raw inputs, it might be beneficial to apply it on a transformed version of the input, that typically aims at making the problem linearly separable. Conventional methods such as Support-Vector Machines (SVMs) [CV95] along with kernel-based techniques [HSS08] are applied to tackle the non-linearity of the input. However, as the number of data grows larger and more complex, SVMs become impractical due to the surge of computational effort.

With the advent of Deep Neural Networks (DNNs), such networks have brought significant increase of performance in a variety of domains thanks to their abilities to interpret large-scale data. From Fig. 11 we observe that DNNs contain multiple hidden layers of so-called neurons compared with logistic regression, which can be seen as an one-layer model. With an activation function applied on each layer after the linear outputs, this grants the opportunity to explore data through multiple transformations.

When it comes to manipulating continuous signal such as an image or an audio clip, a very popular method is to rely on Convolutional Neural Networks (CNNs [LB+95]). The main interest of using CNNs is that they are equivariant to translations, meaning that if we denote c their mathematical function, and T a translation, then $c \circ T = T \circ c$. This is especially interesting when dealing with vision problems where intuitively a translation of the image is similar to a camera shot that would be obtained by slightly moving the objective of the camera, in which case the objects

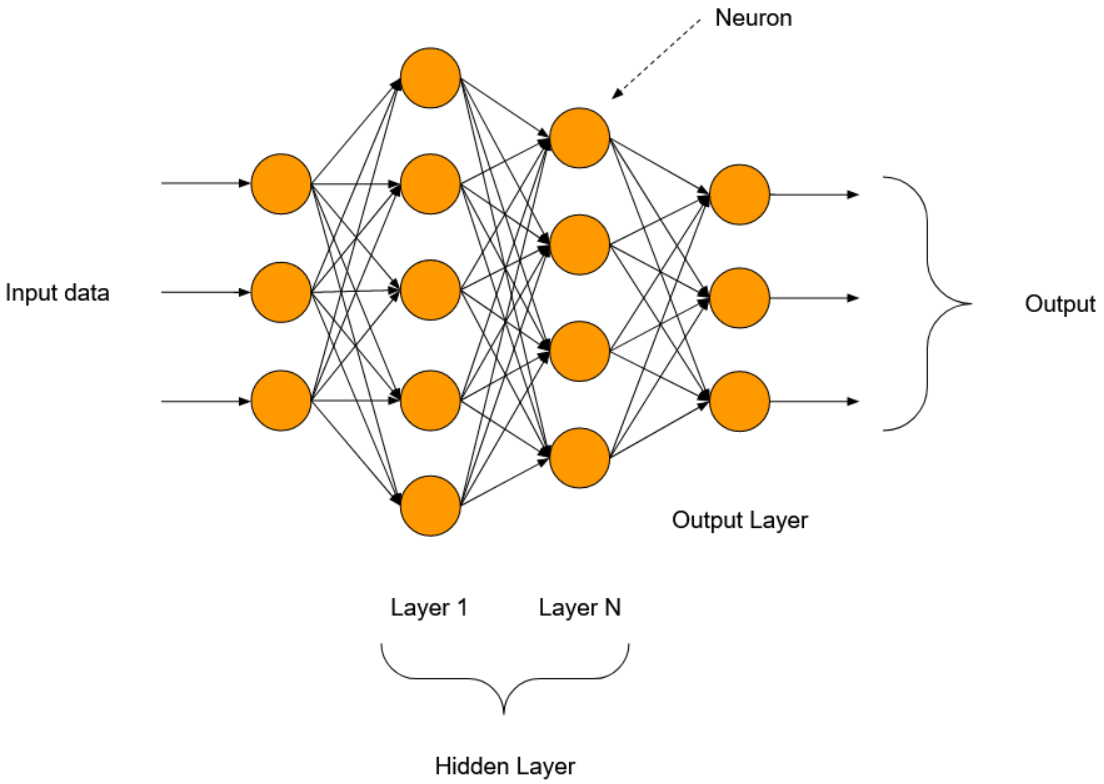


Figure 11: A general model of a deep neural network.

in the scene should not be different, but just translated in the opposite direction.

For image inputs, such a network is made by assembling layers that are defined using a 4d tensor $\mathbf{K}$ (here we ignore the bias tensor for simplicity). The tensor $\mathbf{K}$ is applied to a 3d input, where the first dimension corresponds to "channels". Typically for raw images there is one channel for each primary color (red, green and blue), and the two other dimensions correspond to the width and the height of the considered image. The role of a CNN is to reduce the targeted images into simpler forms that are easy to process while capturing critical information for predictions, and it is generally composed of 1) Convolutional layers, 2) Pooling layers and 3) Fully connected layers.

1.1.4.1 Convolutional layer

Usually a convolutional layer contains convolutional operations on the input using a so-called kernel or filter. Namely, for an input $\mathbf{x}_{in}$ of size $c_{in} \times w_{in} \times h_{in}$ representing the channels (or depth), width and height, the tensor $\mathbf{K}$ of size $c_{out} \times c_{in} \times w_k \times h_k$ is defined to contain c_{out} filters/kernels of size $c_{in} \times w_k \times h_k$ that are used to perform element-wise matrix multiplication on the patch $\mathbf{P}$ of the input. Fig. 12 illustrates the movement of a 3x3x3 kernel applied on a 3-dimensional input. We can see that the kernel has the same depth as the input, and it moves from left to right, up and down to perform convolutional operation on different parts of the input until all parts are traversed. All the results are summed to have a squashed one-layer feature output of size $w_{out} \times h_{out}$, the values of which are detailed in the next

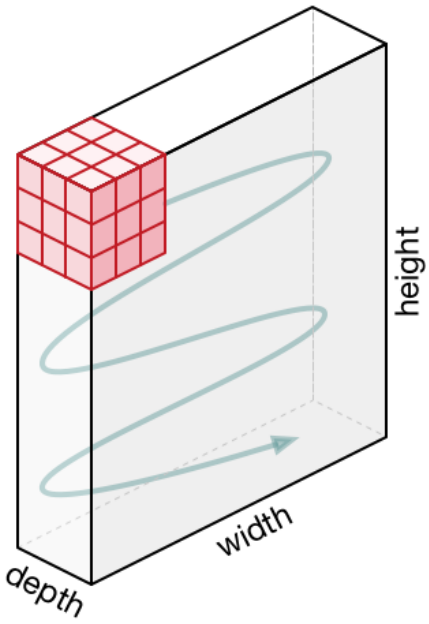


Figure 12: Illustration of a kernel (in pink) and its movement throughout the input. Source: `https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53`.

paragraphs, and the overall output tensor $\mathbf{x}_{out}$ will thus be of size $c_{out} \times w_{out} \times h_{out}$.

As for the value of w_{out} and h_{out}, they depend on a few predefined parameters during matrix multiplication: 1) a stride value s that determines the moving stride of the kernel hovering the input; 2) a padding value p indicating the extended dimensions added to the input in order to maintain/decrease the dimension of the feature output. Note that usually we apply the same s and p on both width and height, and we can thus compute the size of the feature output to be as follows:

$$\begin{aligned} w_{out} &= \left\lfloor \frac{w_{in} - w_k + 2p}{s} \right\rfloor + 1, \\ h_{out} &= \left\lfloor \frac{h_{in} - h_k + 2p}{s} \right\rfloor + 1. \end{aligned} \tag{6}$$

For CNNs applied on images, the convolutional operation has the advantage of capturing the spatial dependencies in an image. And different kernels allow the network to explore multiple aspects of the input that are critical for predictions.

On top of the affine transformation described above, a nonlinear function σ is applied to the result. Most of the time, this function consists of a Rectified Linear Unit (ReLU [Aga18]), that suppresses the negative values in its input, but in some cases we use other nonlinear functions. An example that comes into play in our contributions is a leaky-ReLU [MHN+13] that keeps a small fraction in the negative part which allows the gradients to flow on during training. In addition, it is also very common to add a batch-normalization layer between the affine transform and the nonlinear transform. A batch-normalization [IS15] layer normalizes the output

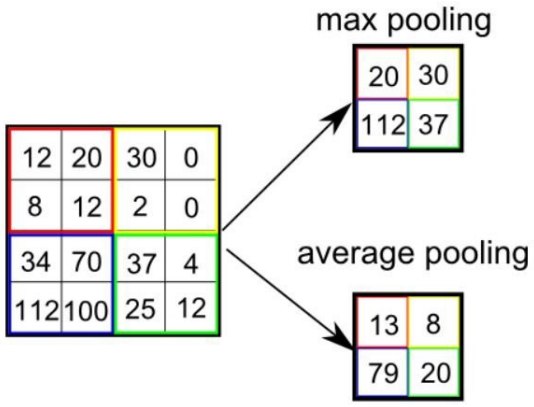


Figure 13: Illustration of pooling process. Source: `https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53`.

of the previous layers for each mini-batch sent into the model, which has the benefit of reducing the effect of data scales, making the training faster and bringing better stability for optimizers.

1.1.4.2 Pooling layer

In addition to convolution layers, pooling layer is also an important element in CNNs. A pooling layer mainly aims at reducing the spatial size of the feature output so that the model requires less computational power during training, reduces noise while maintaining its effectiveness. There are two common types of pooling: 1) max pooling that outputs the maximum value from the image patch covered by the kernel, and 2) average pooling that returns the averaged value of the patch. In Fig. 13 we illustrate these 2 pooling techniques with a 2x2 kernel and a stride of 2.

1.1.4.3 Classifier

At the end of a CNN we usually add a classifier that consists of a simple logistic regression. Namely, it is composed of 1) a fully connected layer that aims at learning the non-linearity of high-level features obtained by the outputs of convolutional layers; and 2) A Softmax layer that computes the soft class assignment for each input (i.e. probability of an input belonging to each class). This layer operates with a Softmax function [Bri89] that is mainly applied in multi-class classification, it is often added in the final layer in order to learn the probability of class belongings.

Having presented the main components of a CNN, its layers can be can be assembled in many different ways such as composition, concatenation, and addition. A very popular method to assemble layers is to use ResNet blocks. Fig. 14 [Ye+20] shows an example of a ResNet block, we observe that it contains 3 convolutional layers, each one consists of $c_{out} = C$ kernels of size 3x3 (stride value $s = 1$) and is followed by a batch normalization and a leaky ReLU operation. The outcome of those layers is then added with the input (processed with a 1x1 convolutional layer and a batch normalization) to prevent vanishing gradients during training. Finally a

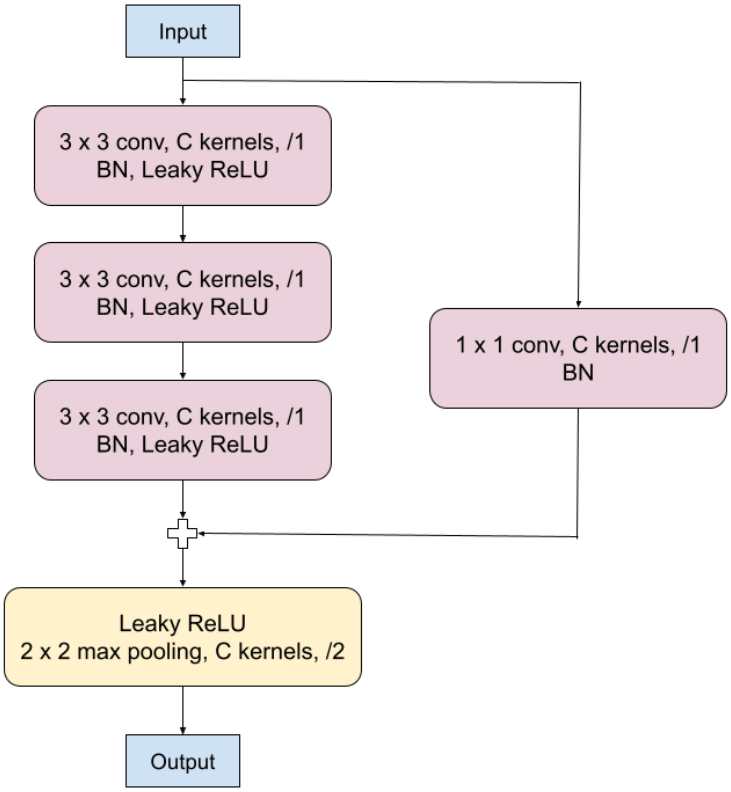


Figure 14: Illustration of a ResNet block.

2x2 max pooling with $s = 2$ can be applied to reduce feature dimensions for the block output.

Note that there are numerous versions of ResNet [He+16; Ye+20] that differ in terms of depth (i.e. the number of ResNet blocks), block structure and positioning of operations. In our work we use two popular ResNet architectures: ResNet12 and WideResNet28-10. Here we use a ResNet12 that consists of 4 blocks described in Fig. 14 followed by a 5x5 average pooling in the end to obtain transformed features of the input in a lower dimensional space (Fig. 15). As for WideResNet28-10 [ZK16] we use the same network structure as [Man+20] throughout the course of this thesis.

In the considered ResNet architecture, the global pooling operation at the end of the architecture to get a shape that works with dense layers so that no flattening is required. ResNet architectures are mainly used in the context of classification, and are able to obtain competitive performance.

1.1.5 Size of architectures

In the past decade, there have been numerous breakthrough in the field of machine learning. In vision, in 2015 for the first time a neural network [Sze+15] is able to outperform humans in the task of predicting the nature of an object in an input image. In 2016, for the first time a totally automated software is able to beat the best Go players [Sil+16]. In 2020, DeepMind proposes alphafold to predict

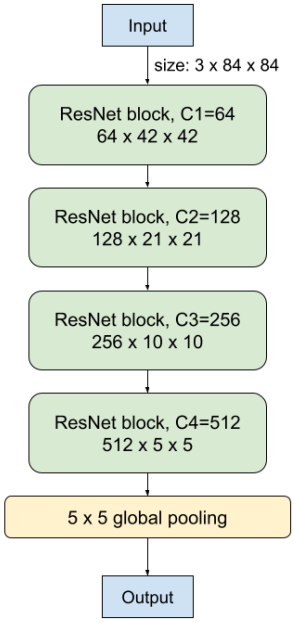


Figure 15: Example of a ResNet12, where C1-C4 indicate the value of c_{out} after each block, and the feature dimension of the output after average pooling is 512.

the 3D structure of proteins based on their DNA sequence [Jum+21]. All these breakthroughs were obtained thanks to the use of deep neural networks (DNNs), of which CNNs are a specific subcase.

It is a question of prime importance to understand why DNNs have the ability of achieving such generalization abilities in so many different domains. Some authors have argued about potential reasons for this [Mal16], but it is fair to say that there is no consensus as of today about the fundamental reasons for this success.

In 2020, an interesting figure was released that (Fig. 16) depicts the evolution of the required computational complexity to achieve state-of-the-art performance in various domains of AI. This figure is reproduced thereafter. It is interesting to note that for the major part of the second half of the 20th century, as well as the first decade of the 21st century, the trend followed Moore's law that described the evolution of processors. It is not a surprise that achieving the state-of-the-art performance requires utilizing most of the available computational resources at one time.

Starting in the 2010s, the trend suddenly changed because of the use of GPUs and TPUs to replace classical processors in the computations. It is clear that these devices are allowing for several orders of magnitude of more computational complexities compared to simple processors. DNNs are one of the few solutions that are able to fully leverage the power of these new devices (e.g. parallel computing [BNH19]), and as such it is expected that DNNs perform better than other solutions that would not be compatible with such hardware.

DNNs tend to have a complexity that is growing with their number of parameters and computing power, even though it is not a systematic truth. And as such,

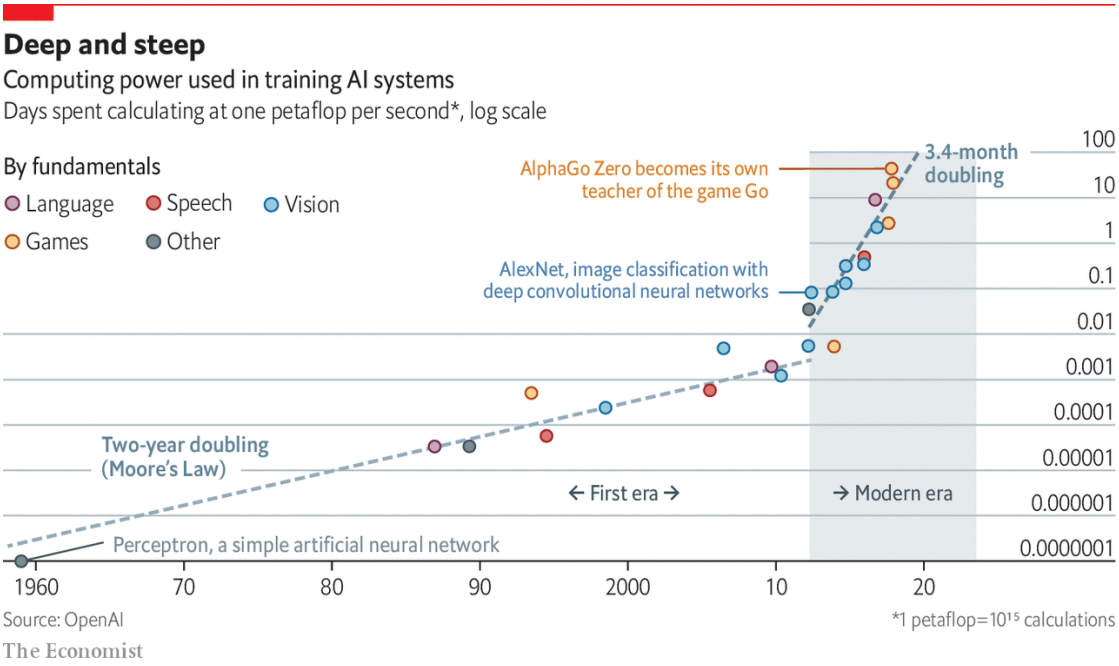


Figure 16: Evolution of the required computational complexity for SOTA performance in AI. Source: `https://www.economist.com/technology-quarterly/2020/06/11/the-cost-of-training-machines-is-becoming-a-problem`.

the most recent models that achieve state-of-the-art performance typically require a huge number of parameters. This trend has been accelerated with the recent introduction of transformers, which are very demanding architectures of DNNs. In vision, models can be found that require hundreds of millions of parameters [He+16; KSH17; BMRG17; HSS18], and in natural language processing, it is not rare to see models with billions of parameters [Rad+19; Bro+20; Ros20]. For instance, Fig. 17 shows the exponential growth growth of model parameters in the domain of Natural Language Processing over the course of 3 years.

1.1.6 Need of data

When learning from scratch, an architecture that relies on millions of free parameters cannot be efficiently trained when given a few training samples. As a matter of fact, this would likely cause an underdetermination problem that would lead to dramatic overfitting.

So together with the growing number of parameters in considered architectures, the datasets used to train those are also growing in size [Rus+15; Rad+18; Rad+19; Bro+20]. Yet, there are numerous applications for which there is no availability of such massive datasets. In such cases, it is needed to find an alternative so that the best performing, huge architectures, can be deployed efficiently.

Among the solutions, transfer learning [PY09; Dai+09; TS10; WKW16] is the most commonly used methodology. The idea is to train architectures with a large available dataset in a supervised manner, even if the dataset differs from the actual task of interest, and then to use domain adaptation techniques to solve the

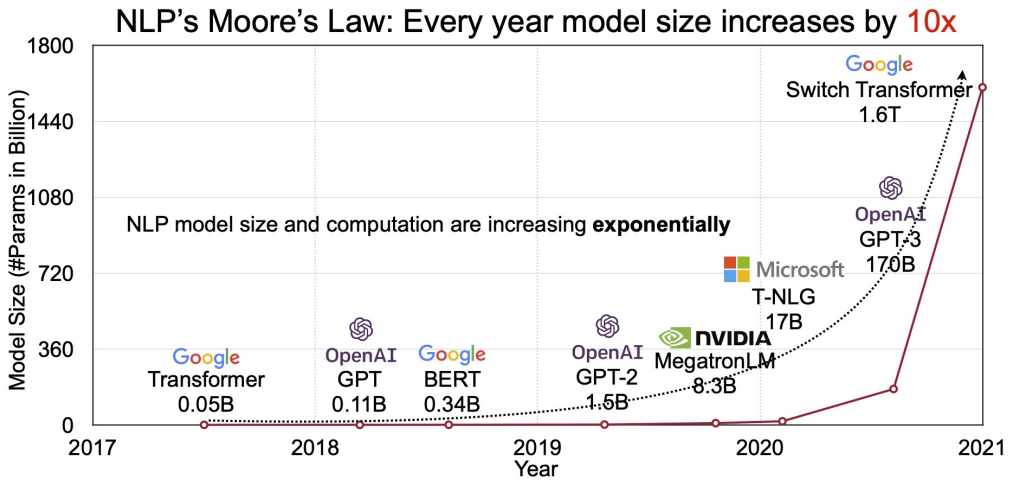


Figure 17: Evolution of model size in NLP from 2017 to 2021. Source: `https://hanlab.mit.edu/projects/efficientnlp_old/`.

considered task. Another training technique of transfer learning consists in using self-supervised learning [Ale+15; GSK18b; DGE15; ZIE16], where the idea is to exploit large collections of data that are unlabelled, to train the architectures and then to adapt these architectures to the considered task.

In our work, we mainly focus on transfer learning, and we make use of some techniques introduced in the field of self-supervised learning (SSL).

1.2 Problematic addressed in this thesis

In this thesis, we deal with situations where there are no massive data available to learn models. For models in the domain of Natural Language Processing (NLP) or image recognition, they often require a large amount of data to learn. Therefore, the thriftiness of data could present a big challenge. Here we present the corresponding research area called Few-Shot Learning and discuss in more detail about its settings.

1.2.1 Few-Shot Learning

The few-shot learning literature is not novel, and the idea falls back to the 90s [Bro+93].

A very interesting seminal work about few-shot learning in the deep learning era is the one presented in [STT12]. In this paper, the authors introduce the toy problem of classification of Tufas, illustrated in Figure 18.

In this problem, we are given images of alien objects that were artificially generated by designers. These objects do not exist. Three of them are highlighted and we are said that they belong to the same category and the aim is to find the other instances of that category.

Interestingly, it is a problem that is very easily solved by humans in that humans learn through concepts such as the form of the Tufas. The fact that they all have

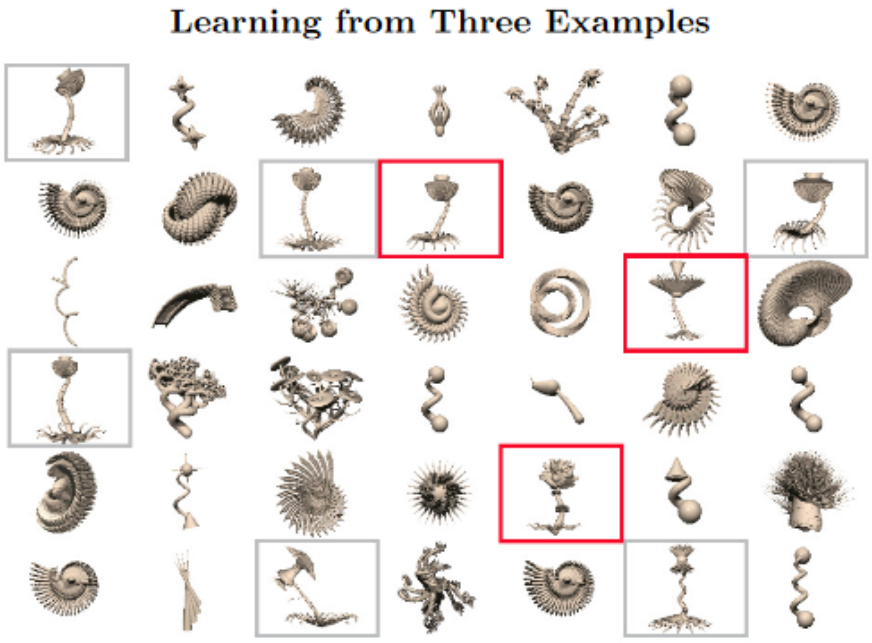


Figure 18: Classification of Tufas: Given only 3 Tufas examples that are boxed in red, the goal is to find out the other Tufas. Source: Ruslan Salakhutdinov, Joshua Tenenbaum, and Antonio Torralba. "One-shot learning with a hierarchical non-parametric bayesian model". In: Proceedings of ICML Workshop on Unsupervised and Transfer Learning. JMLR Workshop and Conference Proceedings. 2012, pp. 195–206. URL: `http://proceedings.mlr.press/v27/salakhutdinov12a/salakhutdinov12a.pdf`. It was published (and can be reproduced) under the terms of Creative Commons Attribution 2.0 licence.

similar roots and spiral-shaped stems separates them from the others. However, this task would be much more difficult to solve when using machine learning solutions, in particular if learning from scratch for the reasons mentioned before.

More generally, the problem of few-shot learning, or more precisely few-shot classification, consists of inferring the class of unlabeled samples based on the observations of only a few labeled samples and possibly a few unlabeled ones.

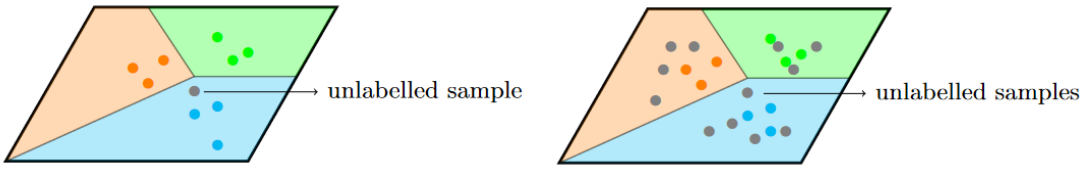
For all the reasons presented before, it is a very challenging problem, and it can be applicable to many different practical use cases. We detail some of them in the next section.

1.2.2 Inductive and transductive settings

It is natural to distinguish two types of few-shot tasks: 1) inductive settings and 2) transductive settings.

In inductive settings, we are only given a few labeled samples for each considered category to train our classifier. Then the purpose is to independently classify previously unseen new samples.

This setting is likely to occur in situations where the acquisition of samples is costly,



(a) Inductive settings **(b)** Transductive settings

Figure 19: Illustration of inductive and transductive few-shot settings.

for example with satellite imaging or some types of medical imaging.

Another possibility is that we are facing very rare events, for which only a few samples are available.

In transductive settings, the problem becomes a semi-supervised learning problem, where we have access not only to a few labeled samples for each category but also to a batch of unlabeled samples on which the predictions are to be made.

Because of the presence of unlabeled samples on top of the labeled ones, there are more possible solutions that are able to use the structure of the unlabeled data.

This setting is likely to occur in situations where the acquisition of data is not the main issue, but the labeling of data can be costly. It is for example the case when it is required to hire experts to label the data, or for exploratory projects.

Another possibility is when the samples of interest are rare and undetectable in our dataset, and as such even labeling a large portion of it would likely only lead to a small number of samples in some categories.

An illustration of the difference between inductive and transductive setting is shown in Figure 19 where colors represent classes and colored points denote the labeled samples in each class. We can see that compared with inductive settings where only one unlabeled sample is allowed at a time for inference, transductive settings allow the access to many unlabeled data, making the problem a semi-supervised one.

To solve a few-shot problem, related literature has been focusing mainly on the following three parts: 1) backbone training that learns a feature extractor using a large available dataset [SSZ17; Che+19a; Man+20; Rod+20; Liu+21a]; 2) feature preprocessing that aligns the extracted features for further modeling [Lic+20; Wan+19b] and 3) classifier design that builds a classifier in order for label predictions [Ren+18; Che+19a; Lee+19; Lic+20]. They will be thoroughly discussed in the next chapter.

1.2.3 Problematic

The few-shot literature has become a very trendy subject within the past few years, with dozens of publications [SSZ17; Vin+16; FAL17; Wan+19b; Man+20; Rus+19; Zha+20; Ye+20; Zik+20; Lic+20; Bou+20b; Kim+19; SE18].

In our work, we have been primarily interested in focusing on transductive few-shot problems. Because vision is often the field where datasets are the most easily accessible, and because it is the one that is by far the most considered in the literature, we also only focused on vision applications within this thesis, even though we consider many of our contributions to be easy to adapt to other domains.

Our main purpose was to investigate the various steps in the pipeline to solve transductive few-shot vision classification problems, and to try to come up with the best performing solution overall.

As such, we focused a lot on obtaining the best performance on standardized benchmarks in the field. Yet, we were also very concerned about justifying the various steps used in our methods and making sure that the proposed solutions would be usable in different contexts/settings. A notable achievement of this thesis was that I was able to reach the first rank on many challenging few-shot benchmarks, competing with dozens of other works.

1.2.4 Contributions

Thus, during the three years of my PhD, I contributed to the three steps mentioned before: 1- learning of the feature extractor, 2- preprocessing of the features and 3- design of the semi-supervised classifier.

Below, I summarize my main contributions and how they relate to each of these steps:

Graph-based interpolation of feature vectors for accurate few-shot classification **Hu, Y.**, Gripon, V. and Pateux, S., 2021, January. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 8164-8171). IEEE.2020

There has been growing interest in Graph Neural Networks (GNN) since their ability to capture the relationships among features according to similarity metrics. In few-shot image classification, especially in transductive settings, GNNs have already been applied in several works such as [Che+21a; Kim+19; SE18]. However the GNNs in these works are mainly used as feature extractors (also called backbones) for a generic dataset to train, which could result in inferior performance compared with a CNN feature extractor like ResNet12. Therefore, in our work "Graph-based interpolation of feature vectors for accurate few-shot classification" we propose to use graph as a preprocessing technique for features extracted from a pretrained backbone. Our proposed method is considered among the first to use graph to preprocess features and has brought significant increase in accuracy compared with inductive baseline.

Improving Classification Accuracy with Graph Filtering Hamidouche, M., Lassance, C., **Hu, Y.**, Drumetz, L., Pasdeloup, B. and Gripon, V., 2021, September. In 2021 IEEE International Conference on Image Processing (ICIP) (pp. 334-338). IEEE.2021

In previous work we show the effectiveness of graph applied on extracted features as a preprocessing technique. However, the proposed graph covers the entire labeled samples containing different classes, which might lead to sub-optimal results due to

the confusion it brings when diffusing features that do not belong to the same class. To address that and explore more functionality of a graph, in this work "Improving Classification Accuracy With Graph Filtering" the graph is applied on each class with the corresponding labeled samples. In the paper we prove the low-pass effect of a graph on class centroids, reducing intra-class noise while keeping the centroid expectations unchanged. The proposed graph used as a filter further improves the performance especially when there are more than 1 labeled samples per class.

Leveraging the Feature Distribution in Transfer-Based Few-Shot Learning **Hu, Y.**, Gripon, V. and Pateux, S., 2021, September. In International Conference on Artificial Neural Networks (pp. 487-499). Springer, Cham.2021

The above two works mainly focus on feature preprocessing and how the use of graph can be beneficial for class predictions. Besides that, a good prediction model also requires the classifier to be well designed. Popular classifiers include logistic regression methods [Dhi+20; Man+20] that learns the boundary parameters to a minimum loss, and clustering methods such as Kmeans [HW79] that performs predictions via estimated class centroids based on Gaussian assumptions. Given the transductive settings, in this work "Leveraging the Feature Distribution in Transfer-Based Few-Shot Learning" we invest in 1) proposing to use Power Transform as part of the feature preprocessing in order to align features with Gaussian assumptions, and 2) building a classifier based on Optimal Transport (OT) [Vil08] that operates under an Expectation-Maximization (EM) framework. With the help of 1) and 2), our proposed method "PT+MAP" obtained state-of-the-art performance on various few-shot settings. The method has been reused and applied by other works in numerous occasions.

Squeezing Backbone Feature Distributions to the Max for Efficient Few-Shot Learning **Hu, Y.**, Pateux, S. and Gripon, V., 2022. Algorithms, 15(5), p.147.2022

Although the classifier in previous work is able to bring large gains with the help of Optimal Transport, one major drawback of the algorithm is its requirement for unlabeled samples' distribution, which is not desirable considering that the proportions of unlabeled samples to be predicted should be unknown in real world scenarios. Therefore in order to address that, in our work "Squeezing Backbone Feature Distributions to the Max for Efficient Few-Shot Learning", which can also been seen as an extended version of the previous work, we suggest a modified version of algorithm that attempts to take into account the variation of class proportions from one class to another. In addition we also integrate a logistic regression based algorithm into the EM framework that further adjusts class centroids based on the pseudo labels on unlabeled samples. To that end our newly modified algorithm is able to obtain decent results compared with PT+MAP, moreover the added logistic regression is able to boost further the prediction accuracy.

Adaptive Dimension Reduction and Variational Inference for Transductive Few-Shot Classification **Hu, Y.**, Pateux, S. and Gripon, V., Arxiv preprint.2022

Although previous work attempts to address the problem of prior constraints with

respect to unlabeled samples, the performance of which is still not ideal in face of the unbalanced setting proposed in [Vei+21], suggesting the limitation of OT without prior estimation. Therefore, in this work 'Adaptive Dimension Reduction and Variational Inference for Transductive Few-Shot Classification' we propose a novel method based on Variational Bayesian inference, along with Adaptive Dimension Reduction that uses Probabilistic Linear Discriminant Analysis to iteratively project data into lower dimensions for label predictions. The proposed method (called 'BAVARDAGE') is able to reach state-of-the-art performance on the unbalanced setting, and competitive results on the balanced setting without any prior knowledge as well.

EASY: Ensemble Augmented-Shot Y-shaped Learning: State-Of-The-Art Few-Shot Classification with Simple Ingredients Bendou, Y., **Hu, Y.**, Lafargue, R., Lioi, G., Pasdeloup, B., Pateux, S. and Gripon, V., 2022. J. Imaging, 8(7), p.179. 2022

Besides preprocessing and classifier design on the extracted features, another important aspect of few-shot classification is the training of a deep neural network, the goal of which is to learn a feature extractor (backbone), i.e. parameters of all layers of the network except the last one, on a generic dataset so that it is able to generalize well on the novel limited data. Therefore, in this work 'EASY: Ensemble Augmented-Shot Y-shaped Learning: State-Of-The-Art Few-Shot Classification with Simple Ingredients' we apply a self-supervised learning technique that learns a backbone by co-training an auxiliary rotation classifier on labeled data. Furthermore, we also propose 1) a multi-crop technique that can be viewed as a pseudo attention model in order to find the zones of interest in an image; 2) an ensemble method in which the extracted features are the concatenation of several backbones pre-trained in the same manner. Our pre-trained backbones are proven to be effective on a variety of benchmarks, the concatenated features are able to reach state-of-the-art performance with a simple soft-kmeans classifier.

1.3 Outline of the manuscript

In chapter 2 we introduce the standard few-shot transductive setting for image classification, as well as the general pipeline of tackling the problem. We also present related works in the field and how they are positioned in the pipeline. In chapter 3, 4 and 5 we present our main contributions that are related to the following papers: [HGP21a; HGP21b; HPG22a] and [HPG22b], along with reflections and discussions of the proposed methods. And finally chapter 6 draws conclusions and discussions about our work.

Chapter 2

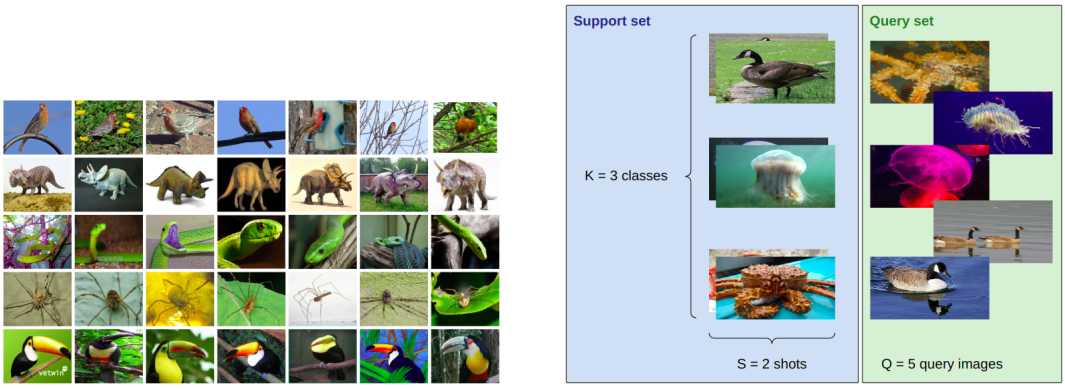
Standard transductive few-shot pipeline

In previous chapter we briefly discuss the topic of my thesis: few-shot classification, as well as its inductive and transductive settings. In this chapter we detail the standard transductive few-shot image classification settings, including the notations, problem statement and benchmarks. Next, we present the pipeline to tackle the problematic: 1) backbone training, 2) feature preprocessing and 3) classifier design. For each step in the element we also present the corresponding related works along with some well-known methods.

2.1 Notations and problem statement

In a typical scenario of Few-Shot image Classification (FSC), we are given 1) a generic training set that contains a large number of labeled samples, enough to learn a deep learning model, and 2) a test set (also called a few-shot task) that is composed of only few labeled samples belonging to classes distinct from those of the generic training set along with some unlabeled samples from the same classes to perform classification.

Benchmarking in the few-shot domain [Vin+16; Ren+18; ORLL18] usually relies on three class-distinct datasets: 1) a base class set that constitutes our generic training set, 2) a novel class set containing many labeled samples from which few-shot tasks are drawn randomly for evaluation purposes, and finally 3) a validation class set that is usually used to tune hyper-parameters before evaluation. A few-shot task is itself composed of 1) a support set in which we have labeled data belonging to K novel classes (S samples per class), and 2) a query set that contains a total number of Q unlabeled data (belonging to the same K novel classes) on which to measure performance. Therefore a few-shot task contains $N = KS + Q$ data samples in total and the goal is to predict class labels of the query set given the support set that has few labeled samples per class. Classically the experiment is conducted on 1-shot 5-way ($K = 5$, $S = 1$) or 5-shot 5-way ($K = 5, S = 5$) scenarios, given $Q = 75$. Considering that few-shot tasks are randomly drawn from the novel class set, they can vary from one another, therefore here we evaluate the performance



(a) Example of a training set **(b)** Example of a test set

Figure 20: Example of a scenario for Few-Shot Classification.

by computing the averaged accuracy over a large number of randomly generated few-shot tasks (usually $10,000$), with a confidence interval of 95% also reported as a criteria. Note that there are some recent works [Vei+21; ORLL18] that propose a 10-shot and a 20-shot setting, which could bring relative large increase in accuracy. However, for settings that grant relatively large numbers of labeled samples, it is not clear whether they still belong to the domain of few shot.

In Fig. 20 we illustrate the scenario of a FSC problem. We can notice that in the test set of this example, there are 3 classes ($K = 3$) with 2 labeled samples ($S = 2$) for each of them, constituting the support set. And there are 5 ($Q = 5$) unlabeled samples in the query set on which we aim at performing classification.

Depending on how the query set is distributed over classes, the transductive Few-Shot Classification can be further divided into two settings: balanced and unbalanced settings.

2.1.1 Balanced setting

A balanced setting implies the exact same number of unlabeled samples per test class, namely we select $q = \frac{Q}{K}$ samples in each of the K classes for label predictions. This may seem obvious and is applied in most works that focus on few-shot classification. However, in a real world scenario the distribution of unlabeled samples among classes is ought to be random and unknown.

2.1.2 Unbalanced setting

Proposed by [Vei+21], an unbalanced setting provides a more realistic scenario where we are not aware of how unlabeled samples are distributed among the K test classes. In mathematical terms, this setting selects unlabeled data using a K-dimensional probability simplex which contains K non-negative numbers that add up to 1, and each dimension represents the proportion of the element with respect to the rest. In [Vei+21], the authors propose to obtain the probability simplex by using a symmetric Dirichlet distribution parameterized by $\boldsymbol{\alpha} = \alpha\mathbf{1}$, where $\mathbf{1}$ is the all-one vector. Namely, denote $\boldsymbol{\pi}$ as a K-dimensional vector representing mixing

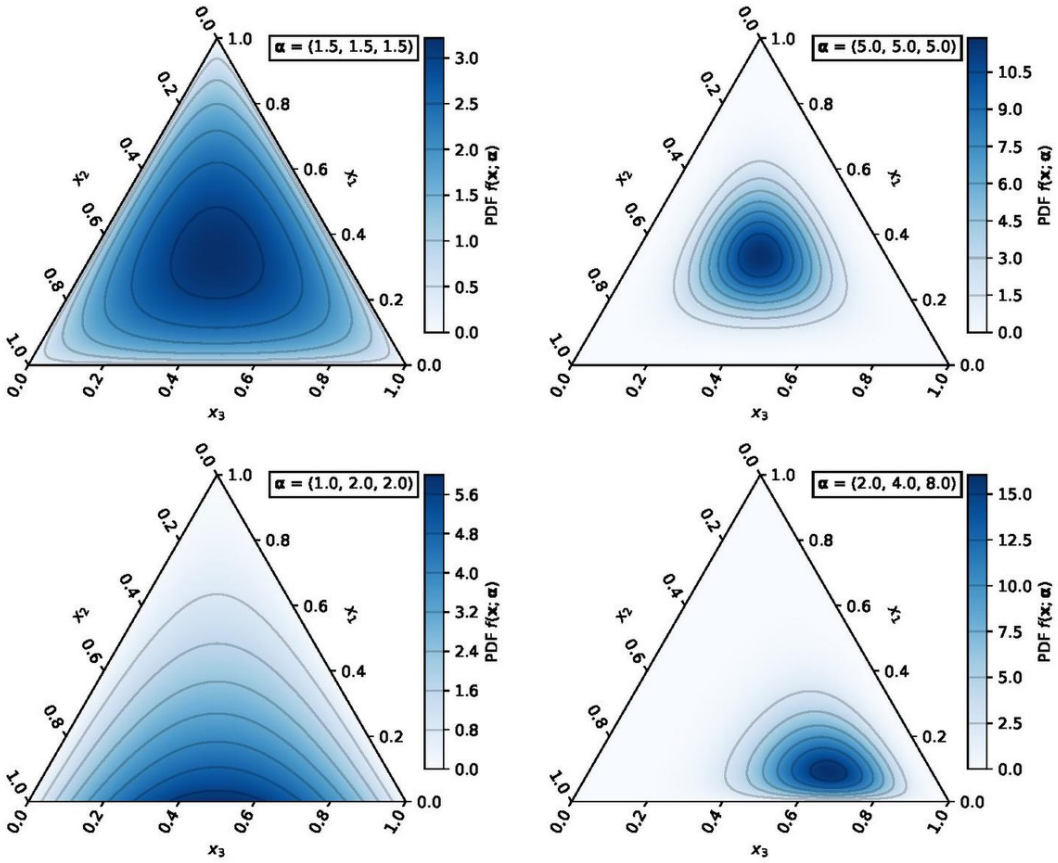


Figure 21: Density probability function of Dirichlet distribution with different $\boldsymbol{\alpha}$ in 3-dimensional scenario. Source:`https://en.wikipedia.org/wiki/Dirichlet_distribution`.

ratios between the classes, it is thus obtained as follows:

$$\boldsymbol{\pi} = [\pi_1, ..., \pi_k, ..., \pi_K] \sim Dir(\boldsymbol{\alpha}) = C(\boldsymbol{\alpha}) \prod_{k=1}^{K} \pi_k^{\alpha-1} = \frac{\Gamma(\sum_{k=1}^{K} K\alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)} \prod_{k=1}^{K} \pi_k^{\alpha-1}. \quad (7)$$

According to the density probability function of Dirichlet distribution presented in Eq. 7 and Fig. 21, we observe that a larger α_o indicates a more uniform distribution for unlabeled samples, i.e. a more balanced scenario. On the contrary, smaller α_o would suggest a more unbalanced situation. In this manuscript we follow the same setting as [Vei+21] and let $\alpha_o = 2$, given the fact that the total number of unlabeled samples Q is fixed.

In this thesis we mainly focus on the standard transductive few-shot setting where we conduct experiments using one benchmark at a time. We compute the prediction accuracy on 1-shot 5-way and 5-shot 5-way scenarios, with $Q = 75$ in both balanced and unbalanced settings.

2.2 Standard benchmarks

Standardized benchmarks for Few-Shot image Classification include *mini*-ImageNet [Rus+15], *tiered*-ImageNet [Ren+18], caltech-ucsd birds-200-2011 (CUB) [Wah+11], FC100 [ORLL18] and CIFAR-FS [Ber+19].

2.2.1 *mini*-ImageNet

mini-ImageNet[1] is a subset of ILSVRC-12 [Rus+15]. It contains a total of 60,000 color images of size 84×84 belonging to 100 classes (600 images per class), these 100 classes are divided into 64, 16, and 20 classes respectively for the constitution of base, validation and novel class sets.

2.2.2 *tiered*-ImageNet

tiered-ImageNet[2] is a larger subset of ILSVRC-12 with 608 classes (779,165 color images of size 84×84 in total) grouped into 34 higher-level categories in the ImageNet human-curated hierarchy. These categories are split into 20, 6, and 8 disjoint sets of base, validation, and novel categories, corresponding to a base set of 391 classes, 97-class validation set and 160-class novel set. As argued in [Ren+18], this split nears the root of the ImageNet hierarchy resulting in a more challenging, yet realistic regime where novel classes are less similar to base classes than with *mini*-ImageNet.

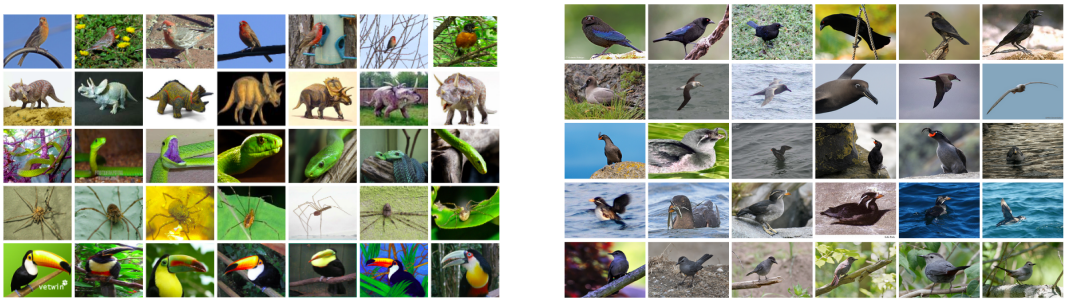
2.2.3 CUB

CUB[3] is a challenging dataset annotated with a total of 11,788 images belonging to 200 bird species. Compared with *tiered*-ImageNet that is more of a coarse-grained dataset with super-categories, CUB is a fine-grained dataset with minor differences between classes. The dataset follows a 100-50-50 base-validation-novel split (Image size: 84×84) for Few-Shot Classification.

2.2.4 FC100

FC100[4] is a recent split dataset based on CIFAR-100 [KH+09] for the problem of few shot. It contains 20 high-level categories split into 12, 4, 4 disjoint categories for training, validation and test. And there are 60, 20, 20 low-level base, validation and novel classes in the corresponding split containing 600 images of size 32×32 per class. Compared with the above 3 datasets, we notice that FC100 has smaller image size, making it quite challenging.

[1] `https://github.com/yaoyao-liu/mini-imagenet-tools`
[2] `https://github.com/yaoyao-liu/tiered-imagenet-tools`
[3] `http://www.vision.caltech.edu/datasets/cub_200_2011`
[4] `https://github.com/ElementAI/TADAM`



(a) *mini*-ImageNet **(b)** CUB

Figure 22: Image examples from few-shot benchmarks (*mini*-ImageNet and CUB).

2.2.5 CIFAR-FS

CIFAR-FS[5] is another split from CIFAR-100 that shares the same number of class splits in the base-validation-novel structure as *mini*-Imagenet. Each class contains 600 images of size 32x32. The split is arbitrary, resulting in better average accuracy than FC100.

Although there are other datasets such as Meta-dataset, DCASE [MHV16], IBC [Bon+21] and Danish Fungi 2020 [Pic+22; Ben+22c] that are recently proposed for few-shot classification, in our work we stick to the standard benchmarks that are presented above.

2.3 Overview of the standard pipeline

Solving a FSC problem usually implies to follow a standard pipeline that we describe thereafter:

A first step is to learn a backbone (also called a feature extractor) using the base classes. We denote it f_θ, where θ represents the parameters of a deep neural network. This step contains a variety of training strategies ranging from episodic training methods in the early works [FAL17; Lee+19; LSQ20] to traditional batch training approaches [Che+19a; Man+20] used in Transfer Learning.

Using the pretrained backbone, a few-shot task takes the form of feature vectors extracted from f_θ. These feature vectors are usually preprocessed before being fed to a classifier. Especially when a classifier requires that features belong to a certain distribution to work well, a well-designed preprocessing method can significantly help boost the performance. Works such as [Che+19a; Lic+20] propose techniques including mean subtraction and graph filtering that boost accuracy.

The last step aims at designing a classifier to be applied onto the preprocessed feature vectors. The literature on this step primarily involves 1): representative models, e.g. clustering methods [Che+19a; Zik+20; YLX21; Lic+20; HGP21b] that attempt to find good parameters for cluster estimations; and 2) discriminative models, e.g. logistic regression methods [Che+19a; Bou+20a; Vei+21] that try

[5] `https://github.com/bertinetto/r2d2`

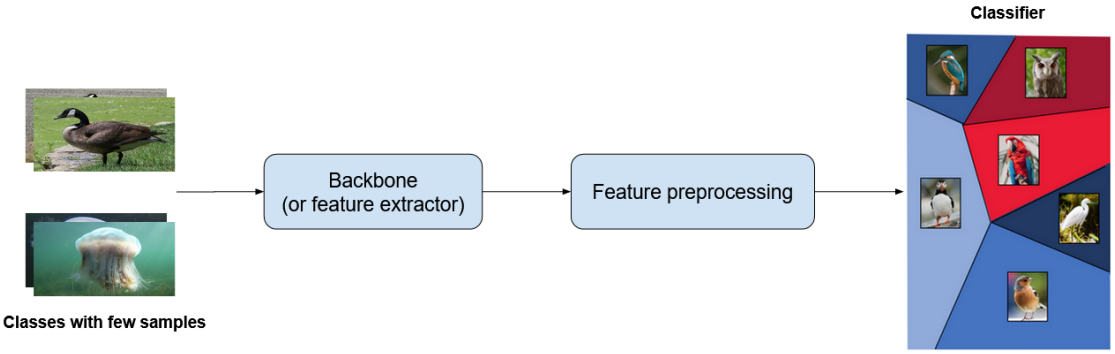


Figure 23: Illustration of the general pipeline for FSC. Images from: `https://medium.com/sap-machine-learning-research/deep-few-shot-learning-a1caa289f18`.

to find good decision boundaries that minimize prediction error. In Fig. 23 we summarize this general pipeline to tackle FSC.

In Table 1 we summarize the steps in the general pipeline along with their functionalities and some corresponding works that we consider to be representative. Note that there may be other works [Wan+20] that categorize differently the proposed methods tackling FSC. Given the growing types of methods that are applied, our proposed categorization attempts to include as many approaches as possible from the earlier works to the most recent ones. In the next sections we will present in more details some selected related works.

Table 1: General pipeline for Few-Shot Classification.

Pipeline	Functionality	Representative works
1. Backbone training	Find good embeddings	MAML [FAL17] ProtoNet [LSQ20] Baseline++ [Che+19a]
2. Feature preprocessing	Adjust distributions	SimpleShot [Wan+19b] TAFSSL [Lic+20]
3. Classifier design	Classify features	Baseline++ [Che+19a] Soft-kmeans [Ren+18] TAFSSL [Lic+20]

2.4 Backbone training

In this section we present the early works on backbone training as well as its evolution over the past years, including the current state-of-the-art methods.

2.4.1 Meta Learning paradigm

Early works on FSC follow the Meta Learning paradigm, the principle of which is often referred as "learning to learn". This idea is inspired from the fact that

humans learn new concepts and skills much efficiently. Small children who have seen dogs and flowers only a few times can quickly tell them apart. People who know how to ride a bike are likely to discover the way to ride a motorcycle fast with little or even no demonstration. Therefore, Meta Learning involves algorithms that learn how to learn through a suite of relevant tasks and is often applied in data-scarce scenarios. In implementation, Meta Learning paradigm is often applied and manifested as episodic training, where the base class set is regrouped into an ensemble of few-shot tasks (also called episodes) before fed into a model, episodic training seeks for a model to learn how to perform on new prediction tasks with novel classes through a series of prediction tasks with base classes.

Generally speaking, there are 2 common Meta Learning approaches that are widely used in the domain of few shot: Optimization based methods and Metric based methods.

2.4.1.1 Optimization base methods

The goal of this type of methods is to learn an optimizer that initializes the model parameters using the training data, so that the model is in a well-established position where only a few more steps are required for the model to perform well for the unseen few shot tasks. Therefore, optimization based methods usually define a learner to learn from a series of prediction tasks, and a meta-learner whose role is to update the learner's parameters using the support set of an unseen task so that the learner can quickly adapt to this new task. There exists several well-known methods in this area, for instance the authors in MAML [FAL17] train on base class tasks with a stochastic gradient decent optimizer, and in Meta-LSTM [RL17] the authors propose to use a LSTM-based meta learner that is thus memory-augmented.

2.4.1.2 Metric based methods

This includes a set of popular methods that aims at finding good embedding for the input data by learning a metric or distance function that measures distance in a low-dimension space. For example the well-known Matching Network [Vin+16] learns an attention kernel in which the attention weight between two data samples is obtained as the cosine similarity; Relation Network [Sun+18] uses Mean Square Error (MSE) as the loss function instead of cross-entropy due to the fact that the proposed metric focuses more on relation scores between two samples, which are computed by regression; and Prototype Network [SSZ17] proposes to define a prototype feature vector, which can be interpreted as the class center representing the cluster, for each class as the mean feature vector of labeled samples belonging to that class, and the distribution over classes for a given test input is a softmax over the inverse of distances between the test data embedding and prototype feature vectors.

In order to develop metrics that are more robust and contain task-specific information, more sophisticated methods are proposed such as [Li+19] where the authors add a plug network to select task-relevant features inside embeddings so that the model can tell the inter-class uniqueness and intra-class commonality for a specific task. In [Lee+19] and [Ber+19], the authors create a class-weight generator by

training the model with a linear classifier (e.g. SVM) in order for the model to minimize generalization error across a distribution of tasks. In the same vein, methods using Graph Neural Networks (GNN) [GMS05] [KZS15] are also proposed in the backbone training process. For example, in [SE18; Kim+19; GK19; Liu+19], the authors incorporate the idea of semi-supervised learning [CSZ09] as a mean to benefit from the unlabeled query data samples when learning from a task, therefore graph methods used in backbone training are more suitable for the transductive setting.

2.4.2 Transfer Learning paradigm

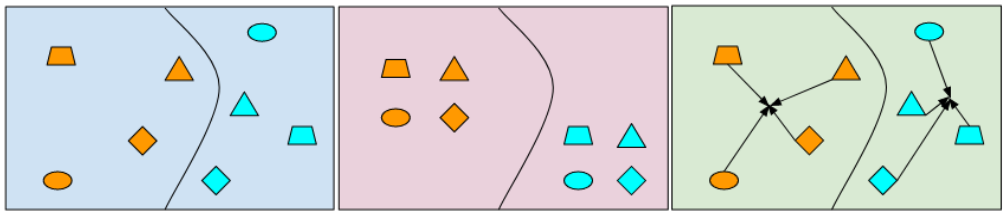
Another possible solution to tackle Few-Shot Classification is Transfer Learning. The idea here being to learn from historical data and make predictions given new unseen data samples. Models that are trained based on transfer are more likely to learn the patterns of historical data and map them onto the new input data. In FSC, Contrary to the Meta Learning paradigm, Transfer Learning based methods train backbones with base dataset in the form of mini-batch (i.e. batch of a fixed number of data points), a regular form used in the training process for deep learning models, then we apply few-shot tasks on a pretrained model to obtain predictions. To distinguish the different data forms in FSC during backbone training, here we denote batch training as the learning process with input data in mini-batches, as opposed to episodic training in the Meta Learning paradigm. In this vein, there exist a variety of techniques that are used to train a model in order for better feature generalizations.

In early work [Che+19a], the authors propose a cosine classifier during training that views column-wise weight parameters as class prototypes so that the model learn better cluster representations for each class. In the meantime, other training techniques such as Self-Supervised Learning (SSL), Distillation, Data augmentation and a recently proposed Two-stage training, etc. start to gain popularity in FSC and works often combine their proposed methods with one or several of these techniques to reach upmost accuracy.

2.4.2.1 Self-Supervised Learning

The main objective of SSL is to leverage the underlying structure of the training data and predict the hidden part of the input from the observable part. Oftentimes SSL can help a model acquire more skills by learning multiple tasks without additional input data, and it is applied regularly to learn large models in NLP [Dev+18; Su+19; Wan+19a]. In FSC, works have been discussing about the equivariance and invariance of the feature representations with the help of SSL methods.

Equivariant presentations indicate similar feature vectors for an image and its transformations, since they are all derived from the same image content. For instance the authors in [Man+20; MSN21; Rod+20] propose to co-train a rotation classifier [GSK18a] by predicting the rotation degrees of artificially rotated images, along with another technique called Exemplar [Dos+14] that aims at making the embedding robust to more image transformations.



(a) Baseline manifold **(b)** Equivariance **(c)** Invariance

Figure 24: Main goal of SSL methods for backbone training. Colors represent different classes and shapes indicate different transformations.

While equivariance can improve the robustness of embedding with respect to its transformations, it does not address class discrimination. For an image belonging to a certain class, different transformations of the image do not change the fact they all belong to the same class and should thus be distanced from images that belong to other classes. Therefore, invariant representations address the issue of class discrimination and attempt to render the classifier invariant to transformations. The proposed methods applied in FSC are often associated with Contrastive Learning, for example in [OHT21] the authors propose to add a contrastive loss function as an auxiliary objective for training. In [Ma+21] the authors learn a model by adopting Contrastive Learning in a supervised manner [Kho+20], meaning that the positive and negative samples for an image are selected with their class labels known. Other methods such as [Liu+21a; Luo+21; Riz+21] all incorporate the similar idea based on sampling positive and negative samples to train backbones with contrastive loss. In Fig. 24 we illustrate the baseline manifold as well as equivariant and invariant manifolds that we attempt to achieve via SSL methods.

2.4.2.2 Distillation

Distillation is another popular technique used in the training process. Considering that a one-hot class representation (observed target labels) for images from the same class may not be the best way due to the nuances such as background or other non-targeted objects among these images [ZS20; Yua+20], we need training methods so that the real target labels (soft labels) of the input images can be learned, therefore the use of distillation can give a better level of generalization and robustness. The first work that adapts distillation into FSC is [Tia+20] in which the authors firstly train a teacher model, followed by a student model with the same network structure to learn the probability simplex (soft target labels) from the teacher. In [Riz+21] the authors apply distillation to images along with the corresponding transformed ones. Thanks to the simplicity of this method, a fair amount of works [MSN21; Riz+21; LW21] apply distillation mixed with other techniques to further boost the performance.

2.4.3 Data augmentation

This technique has been widely used and has become a standard procedure for backbone training, including in the domain of few shot. Earlier works [ZZK19;

Che+19b; Wan+18] seek to cut images into patches or deform them in order to provide more samples. In [Man+20] the authors apply Manifold Mixup to artificially create images with new labels. Some other works [Xin+19; Sch+19] incorporate the semantic information of class labels for a better prototype alignment.

Recent works [Zha+20; Luo+21] propose a new setting in which the test data are processed with images of original resolution instead of images tailored for the benchmark. Namely, this tends to give better performance since image with larger resolution contains more information for the process. And usually in FSC, data augmentation methods are applied in combination of other techniques such as self-supervised learning, distillation and so forth, and they can be effective in both Meta Learning and Transfer Learning paradigms.

2.4.4 Others

There also exists many works that propose other methods for the backbone training in the domain of FSC, for example in [Xu+21; Zha+19] the authors learn models based on variational bayes [Sun+19; Vla+19; Wu+19a]. Authors in [DSM19] apply and analyse ensemble methods in the context of few shot. In [Bat+20; Bat+22] the authors propose a new "CNAPS" architecture that attemps to realise domain adaptation between base data and few shot tasks. [Yue+20] proposes a Structural CausalModel (SCM) for the causalities among the pretrained knowledge. Some works [Dhi+20; SCA20; Ben+22c] also discuss the design of base dataset as well as the backbone structures ranging from the original ResNet12 [He+16] to WideResNet [ZK16], and finally to a modified ResNet12 [Ye+20; Tia+20] that reaches competitive performance.

More recently, two-stage training becomes a newly proposed training strategy that combines both batch training and episodic training. Namely a model is firstly learned and initialized with a batch training, then an episodic training is performed on top of the model, with the same training data reshaped into few-show prediction tasks. This technique adds up the advantages of both training methods, i.e. methods based on batch training tend to learn better patterns from historical data while lacking experience for solving a specific task, Meta Learning based methods on the contrary focus more on learning the task-solving experience as well as the task-relevant parameters, less on the patterns of a seen dataset.

Recent literature proposes various methods based on this two-stage training process, for instance in [Ye+20] the authors propose to episodically train an attention model on top of a pretrained backbone. And authors in [Zha+20] propose to train episodes using Optimal Transport on image patches in the second training stage. Along with other works such as [Rod+20; WTH21; Rod+20], this training procedure can also be integrated into other techniques presented above.

2.5 Feature preprocessing

Next comes the intermediate step of preprocessing features extracted from a pretrained backbone (also called a feature extractor), aiming at better aligning

features to fit into distribution assumptions (oftentimes Gaussian) for the classifier design in the next section.

Since preprocessing methods have been studied in the early literature [Tuk77; CVS15; SS97] that can be widely applied in different vision tasks and settings, the techniques proposed in the field of FSC are similar. For instance in [Wan+19b] the authors propose a mean-subtraction method that subtracts all feature vectors of a few-shot task by the mean vector of the training set before applying L2 normalization, aiming at better aligning features of this specific task. And in [Lic+20] the authors apply the same procedure as [Wan+19b] except that they use the mean vector of the entire few-shot task or two mean vectors (support set mean and query set mean) for mean-subtraction.

Depending on how a backbone is pretrained, one can decide whether to use feature preprocessing. Usually for methods that follow the Meta Learning paradigm, it is not necessary to preprocess novel features thanks to the fact that feature alignment has already been dealt with during episodic training. Therefore for these methods we use raw feature vectors of a test set to evaluate the performance directly from a pretrained backbone [SSZ17; Ye+20] or finetune these features with logistic regression methods [Che+19a; Man+20]. However, for Transfer Learning based methods, preprocessing methods are often beneficial to adjust raw features for further usage. For a given backbone initialized with batch training, observations tend to show that a mean-subtraction followed by a L2 normalization on raw features give a very similar accuracy to if we apply an episodic training on these features.

2.6 Classifier design

The last and yet crucial step for an FSC algorithm to work well is the choice of its classifier. The goal is to design a classifier that predicts labels of the query data in a way that minimizes errors. According to the literature on FSC, methods related to this step can be mainly categorized into logistic regression methods and clustering methods.

2.6.1 Logistic regression

Works in FSC or other research areas mainly apply logistic regression during what is called finetuning (with unseen data after backbone training), the goal is to find decision boundaries that minimize prediction errors with the help of available data. And the errors are minimized by finding the global minimum of a convex cost function via Gradient descent.

Earlier works such as [Che+19a; Man+20] both use a standard logistic regression model on raw features, namely they add a sigmoid function on the weighted sum of weight coefficients and features, followed by a cross-entropy loss function to quantify errors. Also in [Che+19a] the authors apply a cosine classifier in which the weight vectors are normalized and thus seen as class prototypes when multiplied with features, the proposed classifier computes the loss based on the cosine similarity

of samples and class centers. Given that logistic regression is itself supervised which requires the labels of the input, a model can be easily overfitted in a few-shot scenario with few labeled data. Therefore, transductive methods have been proposed to make use of the query set. Generally the idea is to obtain pseudo labels for the unlabeled data before proceeding them with logistic regression. For instance in [LSA21], besides the cross-entropy loss on the support set, the authors propose to add another loss based on the pseudo labels of the query set and select query samples that have the least loss. In [Bou+20a] the authors propose a loss based on the mutual information between query samples and their latent labels, the loss consists of a term for conditional entropy that encourages the model to predict confident scores, and another regularization term that encourages uniform label distributions while preventing degenerate solutions. Following the same method, [Vei+21] proposes to down-weight the marginal entropy term in order to minimize the Kullback-Leibler divergence between predicted marginal distribution and uniform distribution.

2.6.2 Clustering methods

Clustering is an unsupervised problem of categorizing each data point into a specific group according to the similarity among samples in the group. Namely, clustering methods assume samples that belong to the same class should have similar features while samples belonging to different classes should not. Given the nature of this type of methods, it comes in handy in transductive Few-Shot Classification in which the query instances are available.

In practice, given a test set that contains both labeled and unlabeled samples, we are in a semi-supervised scenario in making use of the support set. And there exist several basic yet important methods that tackle the problem of few shot.

2.6.2.1 Nearest Class Mean (NCM)

One of the most basic methods is called NCM. For each class, a NCM classifier firstly computes its prototype as the mean vector of samples belonging to that class, then we assign an unlabeled sample to the nearest class according to the L2 distance. Fig. 25 illustrates the NCM classifier. In a 2-way (orange and blue) scenario, each class has 3 labeled samples (colored data points) along with unlabeled ones (data points in black). The prototype is computed as the mean vector of labeled samples.

In FSC, [Wan+19b] is among the first works to propose the use of NCM. However, there are limits related to this method. Firstly, due to few labeled samples in a test set, the estimated prototypes might be heavily skewed and do not well represent the class, as shown in Fig. 25 for instance. Moreover, this method is primarily applied in disregard of the unlabeled samples (inductive setting) in few-shot literature.

2.6.2.2 Kmeans

Compared with NCM that is mainly a supervised method, Kmeans is an unsupervised algorithm that makes use of the query set to level the prototype estimations in an iterative manner. In FSC, Kmeans is usually based on class prototypes

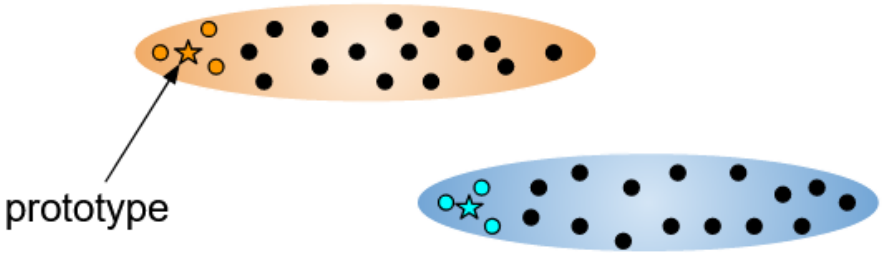
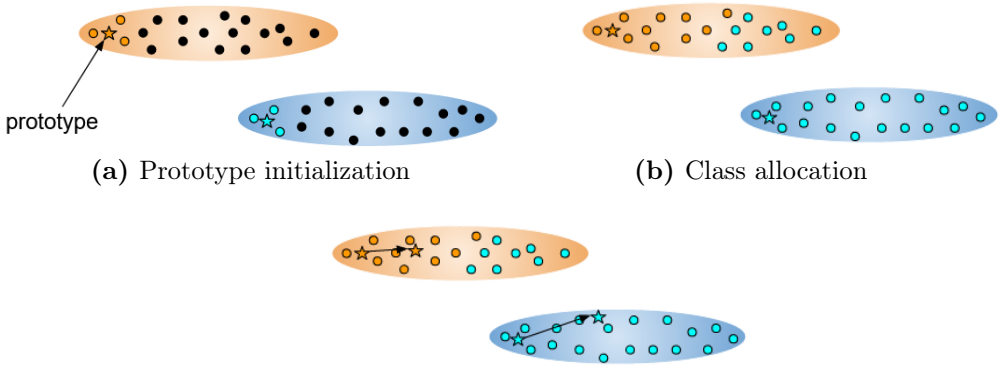


Figure 25: Illustration of a Nearest Class Mean classifier.



(a) Prototype initialization

(b) Class allocation

(c) Prototype update

Figure 26: Illustration of Kmeans algorithm.

initialized with support set samples. Namely we compute the class presentations to be the mean of labeled data, same as in NCM, then we allocate unlabeled examples to their nearest prototypes, finally we update the prototypes according to the class allocations on both support and query set of a few-shot task. The process is repeated until the prototypes turn stable. Fig. 26 illustrates the algorithm in a nutshell, we can see that the updated prototypes are better positioned to represent the classes.

2.6.2.3 Mean shift

Different from Kmeans, Mean shift [Che95; Der05] algorithm does not require the number of clusters as an input parameter, instead this method defines a range parameter that represents the radius of a circled area of a prototype. For data points within this area, we assume these samples belong to the prototype class and thus compute and shift the prototype according to the density of data samples. Note that this is also an iterative process until the prototypes are stabilized.

There are some commonalities between Kmeans and Mean shift algorithms, for instance they both use hard class assignment, i.e. each unlabeled sample is assigned to exactly one cluster. And they both make use of the metric based on L2 distance for optimizations. However, these two methods only consider the prototypes of the targeted clusters without taking their variances or covariances into account, which may lead to sub-optimal solutions.

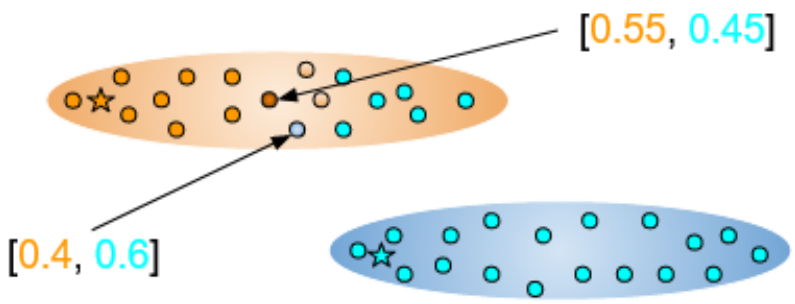


Figure 27: Illustration of soft class assignment.

2.6.2.4 Soft-kmeans

Different from the above two methods, a Soft-kmeans algorithm uses probabilistic modeling while estimating the variance of a cluster. Soft-kmeans operates under an Expectation-Maximization (EM) framework which aims at maximizing the likelihood to find the statistical parameters of a cluster. Therefore, the algorithm alternates between two steps: 1) E-step that tries to find soft class allocations using the current cluster parameters, and 2) M-step that aims at updating these cluster parameters according to the latest class allocations.

Fig. 27 illustrates the concept of soft class assignment, we can see that instead of a hard assignment that is either 1 or 0, the soft assignment gives the probability of a data point belonging to any class. Therefore the updated prototype is the weighted average of samples whose weights are the probabilities of them belonging to the corresponding cluster. In a Gaussian Mixture Model (GMM), Soft-kmeans also takes into account the cluster variances, and the distances between a sample and the prototypes are presented as probabilities using a softmax function.

Note that Kmeans also operates under the EM framework, it is often called hard EM due to its hard assignment nature. And given that all samples are given the same weight for prototype update, kmeans favors spherical clusters while assuming all clusters to have the shared variance of 1 in each feature dimension. Soft-kmeans on the other hand frees up more restrictions for cluster estimations and allows more parameters to be involved. In FSC, we usually design classifiers based on Gaussian assumptions on clusters, and there exist works [Ren+18; Lic+20; Bat+22; HLLJ19] that propose methods that are presented above along with some variants.

2.6.2.5 Optimal Transport

Recent state-of-the-art methods propose clustering methods based on Optimal Transport [Vil09] (OT), the goal of OT applied in FSC is to allocate samples to classes with a minimum cost. On this front we believe to be among the first to incorporate OT under the EM framework for class assignment and parameter update in transductive FSC and achieve significant gain in terms of accuracy, it is the main contribution of my thesis and will be explained in detail. Later on more and more approaches such as [LSA21; CVK21; Ort+21; ZK22] follow suit and develop methods based on OT. However, one of the major concerns of OT based methods is that they require prior knowledge about the distribution of the

query set to work well, while in most real world scenarios we are not aware of how the test data are selected. For instance in an unbalanced setting where unlabeled samples are selected according to $\boldsymbol{\alpha}$ of a Dirichlet distribution, these methods tend to have a catastrophic drop in accuracy [Vei+21].

2.6.2.6 Dimension reduction

In order for a better cluster estimation on recognizing images with few examples in the feature space, dimension reduction can be helpful to reduce noise that exist in features, given the fact that in FSC the feature dimension is several times larger than the total number of samples in a few-shot task. Dimension-reduced feature vectors also make a model more stable with respect to its parameters. In [Lic+20] the authors propose to use PCA under Gaussian assumptions and ICA under non-Gaussian assumptions to reduce dimensions before applying clustering methods. And there is still research that needs to be conducted at this front, given that both PCA and ICA are unsupervised methods that do not consider the labels of samples. In the next chapters we present our contributions as well as discussions about our proposed methods.

2.7 Problems tackled in this manuscript

In the next chapters, we will discuss our main contributions in tackling transductive few-shot classification.

Our contributions have been mainly focused on the feature preprocessing and classifier design steps in the pipeline. Concerning the feature preprocessing step, feature alignment is an important question. In this vein, there are two aspects that can be studied: 1) the aim to group similar features while distancing dissimilar ones; and 2) the aim to adjust features so that their distributions become more desirable for the classifiers that follow.

To address 1), in this manuscript we propose a method based on graph neural networks, which is presented in Chapter 3. And for 2) we propose to apply a technique called Power Transform that can help significantly increase the performance by reshaping the feature distributions to be close to Gaussian. This will be presented in Chapter 4 in detail.

In terms of classifier design, the essence is to find parameters that can best estimate a cluster. Under the Gaussian assumption on the feature distributions, in Chapter 4 we present our other contributions that put forward a clustering method established on Optimal Transport, further improved by applying a logistic regression algorithm that makes use of the unlabeled data. Our proposed methods reach state-of-the-art performance under class-balanced prior.

Moreover, in order to tackle the unbalanced few-shot setting that is more applicable for real world scenarios. In Chapter 5 we propose another clustering algorithm that is based on Variational Bayesian inference and adaptive dimension reduction to best align and estimate clusters. The proposed algorithm requires no prior about

the query set, it reaches state-of-the-art performance in the unbalanced setting and competitive results in the balanced setting.

Therefore, in the next chapters we present in detail our contributions that focus on feature preprocessing and classifier design for transductive few-shot classification.

Chapter 3

Graph-based Interpolation of Feature Vectors for Accurate Few-Shot Classification

In the previous chapter we presented the standard few-shot settings in detail and the pipeline used to tackle the problematic. In this chapter we present our work contributing to the feature preprocessing step in the pipeline, the corresponding paper [HGP21a] and additional discussions about the proposed method.

3.1 Context

Since the concept of meta learning was applied in deep learning, few-shot classification has become a popular research area with well-known methods such as [FAL17; SSZ17; Vin+16], all tackling the task in the inductive setting.

With more studies on the subject over the past years, the concept of transfer learning has also proven to be effective for the task [Che+19a; Man+20], the idea is to train a feature extractor/backbone using a generic dataset so that it generalizes well for the limited unseen data. Compared with meta learning paradigm that learns to learn a task, transfer learning focuses more on learning the knowledge from an existing data and transferring them onto novel tasks.

In addition, a novel semi-supervised setting in few-shot classification (called transductive setting) has been introduced [Ren+18] as well, early works use graph neural networks [SE18; Kim+19; Liu+18] during backbone training (especially episode training) to assimilate samples belonging to the same class based on the adjacent matrix on a few-shot task. Although models trained using graph neural networks help increase the prediction accuracy, they lack task-specific information for the test data, resulting in less superior performance compared with transfer learning based methods.

While graph neural networks used as backbones do not show superior performance compared with CNNs, they do have the advantage of grouping similar features based on their similarities, since the construction of graph consists of building a

similarity matrix for all feature vectors. Therefore, we suggest that the use of graph could be beneficial on the feature level alone as well, namely in feature preprocessing, without the involvement of training.

In terms of feature preprocessing, early work [Wan+19b] proposes mean-subtraction with the mean vector of base dataset followed by a L2 normalization, work in [Lic+20] subtracts the feature vectors of a few-shot task by the mean vector of all samples in order to reduce episode bias.

Inspired by graph neural networks, in this work we use a graph to perform feature preprocessing. Namely, our proposed method is operated on the extracted features from a pre-trained backbone, as is usually the case for other preprocessing methods. The rationale is to filter out large frequencies over the graph that would likely correspond to outliers. To this end, we adapt the methodology introduced in Simplified Graph Convolutional [Wu+19b] (SGC) where features are smoothed using a graph connecting samples. Contrary to this work, we have no access to any explicit graph in our framework, and thus we build such a graph exploiting the similarity between samples of the considered few-shot task, both support and queries, in the feature space.

In this chapter, we address the problem of adaptation of extracted features in improving the few-shot classification. As a result, we present our work [HGP21a]:

Graph-based interpolation of feature vectors for accurate few-shot classification **Hu, Y.**, Gripon, V. and Pateux, S., 2021, January. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 8164-8171). IEEE.2020,

in which we introduce a graph-based methodology to preprocess feature vectors in the context of transductive few-shot classification. The code can be found at `https://github.com/yhu01/transfer-sgc`.

3.2 Paper on graph-based interpolation of features

Graph-based Interpolation of Feature Vectors for Accurate Few-Shot Classification

Yuqing Hu
Electronics Dept., IMT Atlantique
Orange Labs
France
Email: yuqing.hu@imt-atlantique.fr

Vincent Gripon
Electronics Dept., IMT Atlantique
Brest, France
Email: vincent.gripon@imt-atlantique.fr

Stéphane Pateux
Orange Labs
Cesson-Sévigné, France
Email: stephane.pateux@orange.com

***Abstract*—In few-shot classification, the aim is to learn models able to discriminate classes using only a small number of labeled examples. In this context, works have proposed to introduce Graph Neural Networks (GNNs) aiming at exploiting the information contained in other samples treated concurrently, what is commonly referred to as the transductive setting in the literature. These GNNs are trained all together with a backbone feature extractor. In this paper, we propose a new method that relies on graphs only to interpolate feature vectors instead, resulting in a transductive learning setting with no additional parameters to train. Our proposed method thus exploits two levels of information: a) transfer features obtained on generic datasets, b) transductive information obtained from other samples to be classified. Using standard few-shot vision classification datasets, we demonstrate its ability to bring significant gains compared to other works.**

I. Introduction

Deep learning is the state-of-the-art solution for many problems in machine learning, specifically in the domain of computer vision. Relying on a huge number of tunable parameters, these systems are able to absorb subtle dependencies in the distribution of data in such a way that it can later generalize to unseen inputs. Numerous experiments in the field of vision suggest that there is a trade-off between the size of the model (for example expressed as the number of parameters [1]) and its performance on the considered task. As such, reaching state-of-the-art performance often requires to deploy complex architectures. On the other hand, using large models in the case of data-thrifty settings would lead to a case of an underdetermined system. This is why few-shot learning is particularly challenging in the field.

In order to overcome this limitation of deep learning models, several works propose to use Graph Neural Networks (GNNs) [2], [3], [4], [5]. GNNs are a natural way to exploit information available in other samples to classify, a setting often referred to as transductive in the literature. However, most often introduced GNNs come with their own set of parameters to be added to the already numerous parameters to tune to solve the considered task. As a consequence, many of these methods do not achieve top-tier results when compared to state-of-the-art solutions.

In this work, we propose to incorporate a graph-based method with no additional parameters, as a way to naturally bring transductive information in solving the considered task. The first step of the method consists in training a feature extractor with abundant data, followed by an interpolation strategy using well designed graphs. The graphs considered in this paper use vertices to represent each sample of the batch, and their edges are weighted depending on the similarity of corresponding feature vectors. The graph is thus used to interpolate features and thus share information between inputs. Once the features have been interpolated, we simply use a classical Logistic Regression (LR) to classify them. This work comes with the following claims:

- We introduce a three-stage method for few-shot classification of input images that combines state-of-the-art transfer learning [6], a graph-based interpolation technique and logistic regression.
- We empirically demonstrate that the proposed method reaches competitive accuracy on standardized benchmarks in the field of few-shot learning and largely surpasses the current works using GNNs.
- We analyze the importance of each step of the method and discuss hyperparameters influence.

The paper is organized as follows. In Section II, we present related works. In Section III we introduce our proposed methodology. In Section IV, we show experimental results on standard vision datasets and discuss hyperparameters influence. Finally, Section V is a conclusion. The source code can be found at https://github.com/yhu01/transfer-sgc.

II. Related Work

Optimization based methods: Recent work on few-shot classification contains a variety of approaches, some of which can be categorized as meta-learning [7] where the goal is to train an optimizer that initializes the network parameters using a first generic dataset, so that the model is able to reach good performance with only a few more steps on actual considered data. The well-known MAML method [8] trains on different tasks with a basic stochastic gradient decent optimizer [9] and Meta-LSTM [10] utilizes a LSTM-based meta-learner that is thus memory-augmented. Meta-learning can be thought of as a refined transfer method, where the few-shot setting is taken into consideration directly when training on the generic dataset. Although both MAML and Meta-LSTM

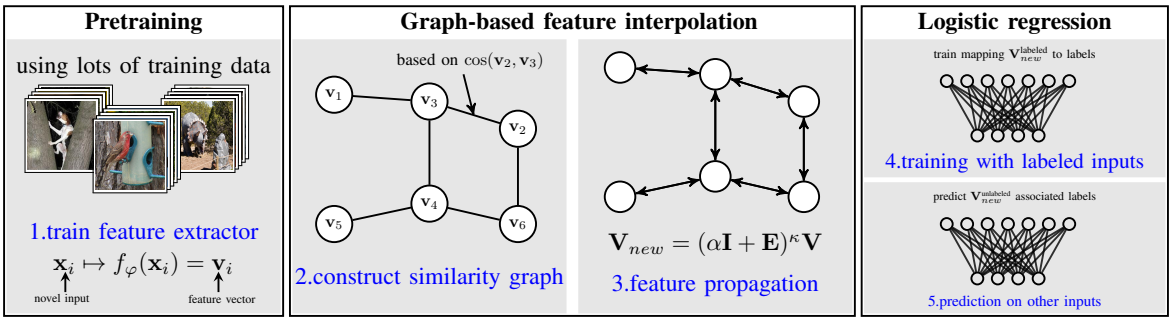


Fig. 1. Illustration of the proposed method. The proposed method is composed of three stages. During the pretraining stage, a classical backbone is trained using large datasets (step 1.). This trained backbone is then used to extract features of a novel dataset, comprising few supervised inputs. During feature interpolation, first is built a similarity graph depending on the cosine similarity between extracted features of both labeled and unlabeled available data (step 2.). Then this graph is used to diffuse (i.e. interpolate) features of similar (neighbor) samples (step 3.). The obtained representations are used to train a simple logistic classifier (step 4.) using the supervised data. Finally, in step 5., the trained classifier is used to perform predictions on unlabeled data.

achieve good performance with quick adaptation, this type of solution suffers from the domain shift problem [9] as well as the sensitivity of hyperparameters.

Embedding based methods: Another popular approach aims at finding compact embedding for the input data by learning a metric that measures the distance in a low-dimensional way. Matching Nets [11] and Proto Nets [12] learn a nearest-neighbor classifier by comparing the distance between the query inputs and labeled inputs with a certain metric, while Relation Nets [13] construct a new neural network that learns the metric itself. If some of these methods are able to outperform MAML, they mainly suffer from over-fitting and a lack of task specific information.

Therefore, ideas have been proposed to address these issues. For example in [14], a plug network is added to find task-relevant features inside embeddings so that the model can tell the inter-class uniqueness and intra-class commonality for a specific task. In [15] and [16], the authors create a class-weight generator by training the model with a linear classifier (e.g. SVM) in order for the model to minimize generalization error across a distribution of tasks. More recently, the use of graph methods [17] [18] starts to gain momentum in the few-shot learning problems. For example, in [2], [3], [4], [5], the authors incorporate the idea of semi-supervised learning [19] as a mean to benefit from the unlabeled query input data when solving a task, what is referred to as the transductive setting. Many recent works propose neural networks able to handle inputs supported on graphs [20]. For example, in GCN [21], the authors introduce a graph convolution operator, that can be used in cascade to generate deep learning architectures. In GAT [22], the authors enrich GCN with additional learnable attention kernels. In SGC [23], the authors propose to simplify GCN by using only one-layer systems on powers of the adjacency matrix of considered graphs. Interestingly, they reach state-of-the-art accuracy with fewer parameters.

Hallucination based methods: Other methods propose to augment the training sets by learning a generator that can hallucinate novel class data using data-augmentation techniques [9]. In [24], the authors extract labeled data into different components and then combine them using learned transformations, while in [25], the authors aim at constructively deforming original samples with new samples drawn from another dataset. However, these methods lack precision as in the way the data is generated, which results in coarse and low-quality synthesized data that can sometimes lead to unsignificant gains in performance [26].

Transfer based methods: As in our work, transfer learning is another possible solution to solve few-shot classification problems. The main idea is to first train a feature extractor using a generic dataset [27], [28], then process these features directly when solving the new task. In [9] a distance-based classifier is applied to train the backbone (i.e. the feature extractor), and in [6], the authors aims at improving the feature quality by adding self-supervised learning and data-augmentation techniques during training. These methods have been proven to perform generally well, yet the challenge remains to fine-tune using the limited amount of labeled data.

In our work, we propose to align multiple ingredients that have been introduced in this section. Namely, we use transfer with graph-based interpolation. We mainly use transfer to exploit information contained in massive generic datasets, and we use a graph method to leverage the additional information available in both labeled and unlabeled inputs. Following the transductive setting, our proposed method can be considered as similar to [5], [2], [3], [4], but contrary to their works, we adopt a strategy in which the considered graph-based method contains no additional parameters to be trained. Our method can also be seen as a modification of Simplified Graph Convolutions [23], where contrary to their work we infer a graph structure from the latent representations of data.

III. Methodology

A. Problem statement

Consider the following problem. We are given two datasets, termed $\mathbf{D}_{base}$ and $\mathbf{D}_{novel}$ with disjoint classes. The first one (called "base") contains a large number of labeled examples

from K_b different classes. The second one (called "novel") contains a small number of labeled examples, along with some unlabeled ones, all from K_n new classes. Our aim is to accurately predict the class of the unlabeled inputs of the novel dataset. There are a few important parameters to this problem: the number of classes in the novel dataset K_n, the number of training samples s for each corresponding class, and the total number of unlabeled inputs Q.

Note that in previous works [5], authors consider that there are exactly $q = Q/K_n$ unlabeled inputs for each class. We consider that this is non-practical, since in most applications there is no reason to think that this holds. We shall see in Section IV that this has strong implications in terms of performance, especially when q is small. Indeed, in practice the Q unlabeled examples are drawn uniformly at random in a pool containing the same amount of unlabeled inputs for each class. So, when Q is large, the central limit theorem tells us that the number of drawn inputs from each class should be similar, whereas it can be highly contrasted when Q is small, leading to an imbalanced case.

B. Proposed solution

Our method is illustrated in Figure 1. We first train a backbone deep neural network able to discriminate inputs from the base dataset $\mathbf{D}_{base} = \{(\mathbf{x'}_1, \ell_1), ..., (\mathbf{x'}_m, \ell_m)\}$, where $\mathbf{x'}_i \in \mathbb{R}^d$ and $1 \leq \ell_i \leq K_b$. The proposed methodology builds upon using this pretrained architecture as a generic feature extractor, what is referred to as *transfer* in the literature [27]. Usually, a common way to extract features is to process data belonging to the novel dataset using the penultimate activation layer. Here, we obtain the extractor $f_\varphi : \mathbb{R}^d \to \mathbb{R}^h$, where φ are the learnable parameters trained using only the base dataset.

We then directly make use of the transferred representations $f_\varphi(\mathbf{D}_{novel}) = \{f_\varphi(\mathbf{x}), \mathbf{x} \in \mathbf{D}_{novel}\}$. Based on these, we build a k nearest neighbor graph using cosine similarity:

$$\cos(f_\varphi(\mathbf{x}), f_\varphi(\mathbf{y})) = \frac{f_\varphi(\mathbf{x})^\top f_\varphi(\mathbf{y})}{\|f_\varphi(\mathbf{x})\|_2 \|f_\varphi(\mathbf{y})\|_2}.$$

This graph contains as many vertices as the total number of inputs in the novel dataset (both labeled and unlabeled ones). Then, we train a model of simplified graph convolution model, that is supervised only for labeled inputs.

The rationale behind this method is twofold: 1) the pre-trained backbone should be able to find good discriminative features since it is trained on a sufficiently large labeled dataset 2) the graph-based interpolation technique should be able to benefit from both the supervised inputs and the unlabeled ones, resulting in significant gains in accuracy when compared to methods that would ignore the unlabeled data.

We show in the experiments that this method is also able to outperform other methods that use the unlabeled data especially when the number of labeled inputs is very limited.

The details of the proposed method are provided in the following paragraphs, first the pre-training stage (i.e. training the generic backbone), followed by the feature interpolation and logistic regression stages.

Pre-training: We follow the methodology introduced in [6]. In more details the feature extractor f_φ and a distance-based classifier $D_{\mathbf{W}_b}$(parametrized by $\mathbf{W}_b$) [29] are trained on $\mathbf{D}_{base}$, where we compute the cosine distance between an input feature $f_\varphi(\mathbf{x'}_i)$ and each weight vector in $\mathbf{W}_b$ in order to reduce the intra-class variations [9]. The training process consists of two sub-stages: the first sub-stage utilizes rotation-based self-supervised learning technique [30] where each input image is randomly rotated by a multiple of 90 degrees. We then co-train a linear classifier to tell which rotation was applied. Therefore, the total loss function of this sub-stage is given by:

$$L_A = L_{\text{class}} + L_{\text{rotation}}. \tag{1}$$

The second sub-stage fine-tunes the model with Manifold Mixup [31] technique for a few more epochs, where the outputs of hidden layers in the neural network are linearly combined to help the trained model generalize better. The total loss in this sub-stage is given by:

$$L_B = L_{\text{ManifoldMixup}} + 0.5(L_{\text{class}} + L_{\text{rotation}}). \tag{2}$$

With this training process, we are able to obtain robust input representations that generalize well to novel classes.

Feature interpolation: We consider fixed the pretrained parameters φ of f_φ. Before training a new classifier $C_{\mathbf{W}_n}$ on the transferred representations of the novel dataset, we propose to interpolate features using a graph.

In details, we define a graph $G_T(\mathbf{V}, \mathbf{E})$ [21] where vertices matrix $\mathbf{V} \in \mathbb{R}^{(sK_n+Q)\times h}$ contains the stacked features of labeled and unlabeled inputs [2]. To build the adjacency matrix $\mathbf{E} \in \mathbb{R}^{(sK_n+Q)\times(sK_n+Q)}$, we first compute:

$$\mathbf{S}[i,j] = \begin{cases} \cos(\mathbf{V}[i,:], \mathbf{V}[j,:]) & if i \neq j \\ 0 & otherwise \end{cases}, \tag{3}$$

where $\mathbf{V}[i,:]$ denotes the i-th row of $\mathbf{V}$. Note that in all backbone architectures we use in the experiments, the penultimate layers are obtained by applying a ReLU function, so that all coefficients in $\mathbf{V}$ are nonnegative. As a result, coefficients in $\mathbf{S}$ are nonnegative as well. Also, note that $\mathbf{S}$ is symmetric.

Then, we only keep the value $\mathbf{S}[i,j]$ if it is one of the k largest values on the corresponding row or on the corresponding column in $\mathbf{S}$. So, as soon as $k \geq (sK_n + Q - 1)$, all values are kept. Otherwise, $\mathbf{S}$ contains many 0s.

Finally, we apply normalization on the resulting matrix:

$$\mathbf{E} = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}, \tag{4}$$

where $\mathbf{D}$ is the degree diagonal matrix defined as:

$$\mathbf{D}[i,i] = \sum_j \mathbf{S}[i,j].$$

Therefore, the graph vertices represent all inputs (both labeled and unlabeled) of the novel dataset. Its nonzero weights are based on the cosine similarity between corresponding transferred representations.

We then apply feature propagation [23] to obtain new features for each vertex. The formula is:

$$\mathbf{V}_{new} = \underbrace{(\alpha\mathbf{I} + \mathbf{E})^{\kappa}}_{\text{“diffusion matrix”}} \mathbf{V}, \quad (5)$$

in which κ and α are both hyperparameters, and $\mathbf{I}$ is the identity matrix. The role of κ is important: providing κ is too small, the new feature of a vertex will only depend on its direct neighbors in the graph. Using larger values of κ allows to encompass for more indirect relationships. Using a too large value of κ might drown out the information by averaging over all inputs. Similarly, α allows to balance between the neighbors representations and self-ones.

Logistic regression: Finally, a softmax classifier is trained using only the labeled vertices. We denote by $\mathbf{V}_{new}^{\text{labeled}}$ the subset of $\mathbf{V}_{new}$ corresponding to labeled vertices, then the predicted results $\hat{\mathbf{Y}}$ can be written following this formula:

$$\hat{\mathbf{Y}}^{\text{labeled}} = softmax(\mathbf{V}_{new}^{\text{labeled}}\mathbf{W}_n), \quad (6)$$

where $\mathbf{V}_{new}^{\text{labeled}} \in \mathbb{R}^{(sK_n)\times h}$, $\hat{\mathbf{Y}} \in \mathbb{R}^{(sK_n)\times K_n}$ and $\hat{\mathbf{Y}}[i,j]$ denotes the probability of vertex i being categorized as being in the j-th class.

Prediction is performed using the same principle, but using unlabeled inputs instead: denote by $\mathbf{V}_{new}^{\text{unlabeled}}$ the subset of $\mathbf{V}_{new}$ corresponding to unlabeled inputs, then we have the decision:

$$\hat{\mathbf{Y}}^{\text{unlabeled}}[i] = \arg\max_j((\mathbf{V}_{new}^{\text{unlabeled}}\mathbf{W}_n)[i,j]). \quad (7)$$

In Table I we summarize the main parameters and hyperparameters of the considered problem and proposed solution. Let us point out that the proposed graph-based method does not contain any parameter to train.

TABLE I
PARAMETERS AND HYPERPARAMETERS OF THE CONSIDERED PROBLEM AND PROPOSED SOLUTION (# STANDS FOR “NUMBER”).

Novel dataset parameters	
K_n	# classes
s	# supervised inputs per class
Q	total # of unsupervised inputs
Proposed method hyperparameters	
$1 \leq k < sK_n + Q$	# nearest neighbors to keep
$\kappa \in \mathbb{N}^*$	power of the diffusion matrix
$0 \leq \alpha \leq 1$	strength of self-representations

IV. EXPERIMENTAL VALIDATION

A. Datasets

We perform our experiments on 3 standardized few-shot classification datasets: miniImageNet [11], CUB [32] and CIFAR-FS [16]. These datasets are split into two parts: a) K_b classes are chosen to train the backbone, called base classes, b) K_n classes are drawn uniformly in the remaining classes to form the novel dataset, called novel classes. Among the K_n drawn novel classes, s labeled inputs per class and a total of Q unlabeled inputs are drawn uniformly at random. As in most related works, unless mentioned otherwise all our experiments are performed using $K_n = 5$ and $Q/K_n = 15$. We perform a run of 10,000 random draws to obtain an accuracy score and indicate confidence scores (95%) when relevant.

miniImageNet: It consists of a subset of ImageNet [33] that contains 100 classes and 600 images of size 84×84 pixels per class. According to the standard [10], we use 64 base classes to train the backbone and 20 novel classes to draw the novel datasets from. So, for each run, 5 classes are drawn uniformly at random among these 20 classes.

CUB: The dataset contains 200 classes and has a total of 11,788 images of size 84×84 pixels. We split it into 100 base classes to train the backbone and 50 novel classes to draw the novel datasets from.

CIFAR-FS: This dataset has 100 classes, each class contains 600 images of size 32×32 pixels. We use the same numbers as for the miniImageNet dataset.

B. Backbone models and implementation details

We perform experiments using 2 different backbones as the structure of feature extractor $f_\varphi(\mathbf{x})$.

Wide residual networks (WRN) [34]**:** We follow the settings in [6] by choosing a WRN with 28 convolutional layers and a widening factor of 10. The output feature size h is 640.

Residual networks (ResNet18) [35]**:** Our ResNet18 contains a total of 18 convolutional layers grouped into 8 blocks. Following the settings in [36], we remove the first two down-sampling layers and change the kernel size of the first convolutional layer to 3×3 pixels instead of 7×7 pixels. Here, $h = 512$.

For the pre-training stage and miniImageNet, we train all backbones for a total of 470 epochs from scratch using Adam optimizer [37] and cross-entropy loss, including 400 epochs on the first sub-stage and 70 epochs on the second sub-stage. For the logistic regression, we train with the same optimizer and loss function for 1000 epochs with learning rate being $1e-3$ and weight decay being $5e-6$, which typically requires of the order of one second of computation on a modern GPU. Note that we observed that convergence usually occurs much quicker than 1000 epochs. In the In-Domain settings two stages are trained on the same dataset with base classes and novel classes respectively, while in the Cross-Domain settings we use these splits from two different datasets (e.g. base classes from miniImageNet and novel classes from CUB).

C. Comparison with state-of-the-art methods

As a first experiment, we compare the raw performance of the proposed method with state-of-the-art solutions with WRN and ResNet18 as backbones. The results are presented in Table II. We fixed α, k and κ respectfully with $s = 1$ and $s = 5$ for the proposed method, as it empirically gave the best results. Note that the sensitivity of these hyperparameters is discussed later in this section.

We point out that the proposed method reaches state-of-the-art performance in both case of 1-shot and 5-shot classification for most of the time, whatever the choice of all considered datasets. Note that the gain we observe is higher in the 1-shot case than in the 5-shot case, this is expected as in the case of 1-shot, the unlabeled samples bring proportionally more information compared to the case of 5-shot. In the extreme case of s-shot, with s large enough, we expect the unlabeled samples to be almost useless.

We also perform experiments where the backbone has been trained using the base classes of miniImageNet but the few-shot task is performed using the novel classes of the CUB dataset. According to the results, we can draw conclusions very similar to the previous study, where the proposed method performs well for this specific task.

D. Comparaison with other GNN methods

In this experiment we compare our performance on miniImageNet with others that use Graph Neural Network to address the few-shot classification. As we can see in Table III, with a three-stage training strategy, our proposed method has largely surpassed the current GNN based methods that train an entire model at once, given the transductive setting.

E. Importance of the parameter-free graph interpolation

In our work, we considered using a parameter-free graph interpolation technique to diffuse features between inputs. As mentioned in the related work section, there are many alternatives, but they come with additional parameters. In the next experiment, we compare the accuracy of the method when using GCN [21] and GAT [22], instead of a simple interpolation. Results are presented in Table IV. We note that the best results are obtained using our designed graph interpolation, which we believe to be due to the fact we use fewer parameters in total. Graph interpolation also has the interest of being many times faster to train. In our experiments, each run took about 0.65 seconds to train using graph interpolation versus 1.18 seconds for GCN and 22.42 seconds with GAT, which happens to lead to the worst performance of our considered methods.

It is worth pointing out that a drawback of the proposed method is that it requires to train a logistic regression model each time a batch prediction is required. In other words, it can be limiting in settings where predictions to make are streamed. However, the time required to train the logistic regression model remains very small in our experiments (less than one second).

F. Influence of Parameters

We then inquire the importance of various parameters of the task to the performance of the proposed method. We begin by varying the number of supervised inputs s, and consider two settings: one where we dispose of an average of $Q/K_n = 5$ unsupervised inputs for each class and one where we dispose of $Q/K_n = 100$ of them. Results are depicted in Figure 2. As we can see, the performance of the method is highly influenced by the number of supervised inputs, as expected. Interestingly, there is a significant gap in accuracy between $Q/K_n = 5$ and $Q/K_n = 100$ for 1-shot setting, even if this gap diminishes as the number of supervised inputs is increased.

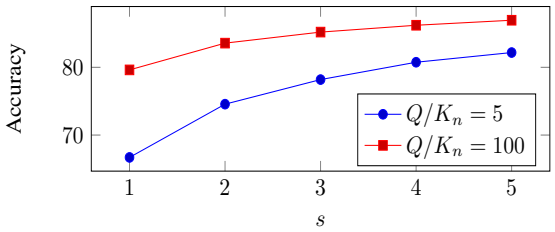


Fig. 2. Evolution of the accuracy of few-shot classification with miniImageNet (backbone: WRN) as a function of the number of supervised inputs s, and for various number of unsupervised queries q. We use $\alpha = 0.5$, $\kappa = 3$ and $k = 10$.

In the next experiment, we draw in Figure 3 the evolution of the performance of the method as a function of the number of unsupervised inputs Q, for 1-shot, 3-shot and 5-shot settings. This curve confirms two observations: a) in the case of 5-shot setting, the influence of the number of unsupervised inputs is little, and the accuracy of the method quickly reaches its pick and b) in the case of 1-shot setting, the number of unsupervised inputs significantly influences accuracy up to a few dozens. It is interesting to point out that about the same accuracy is achieved for 5-shot using $Q = 1$ and 1-shot using $Q = 100$, suggesting that 100 unsupervised inputs bring about the same usable information as 4 labeled inputs per class.

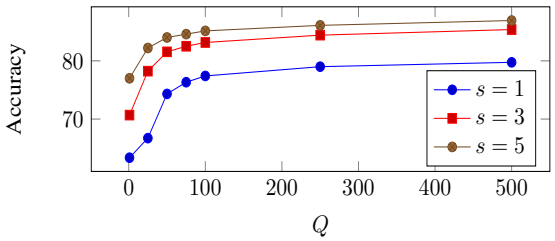


Fig. 3. Evolution of the accuracy of few-shot classification with miniImageNet (backbone: WRN) as a function of the number of query inputs Q, and for various number of unsupervised inputs s. We use $\alpha = 0.5$, $\kappa = 3$ and $k = \min(10, sK_n + Q - 1)$.

In the next experiment we look at the influence of the parameters κ and α which respectively control to which power the diffusion matrix is taken and the importance of self-representations. In Figure 4, we draw the obtained mean accuracy as a function of κ, α and k. We use $s = 1$ and $Q/K_n = 15$ in this experiment. There are multiple interesting conclusions to draw from this figure.

1) This curve justifies the previously mentioned choice of parameters, leading to the best performance.
2) We observe that when k is large and α is small, it is better not to use powers of the diffusion matrix. This is the only setting where this statement holds, emphasizing the fact that if the graph is not sparse and self-importance is low, powers of the diffusion

TABLE II
1-SHOT AND 5-SHOT ACCURACY OF STATE-OF-THE-ART METHODS IN THE LITERATURE, COMPARED WITH THE PROPOSED SOLUTION. WE PRESENT RESULTS USING WRN AND RESNET18 AS BACKBONES. FOR THE PROPOSED SOLUTION, WE USE THE HYPERPARAMETERS $\alpha = 0.5$, $k = 10$ AND $\kappa = 3$ FOR $s = 1$; $\alpha = 0.75$, $k = 15$ AND $\kappa = 1$ FOR $s = 5$.

		miniImageNet	
Method	Backbone	1-shot	5-shot
MAML [8]	ResNet18	$49.61 \pm 0.92\%$	$65.72 \pm 0.77\%$
Baseline++ [9]	ResNet18	$51.87 \pm 0.77\%$	$75.68 \pm 0.63\%$
Matching Networks [11]	ResNet18	$52.91 \pm 0.88\%$	$68.88 \pm 0.69\%$
ProtoNet [12]	ResNet18	$54.16 \pm 0.82\%$	$73.68 \pm 0.65\%$
SimpleShot [36]	ResNet18	$63.10 \pm 0.20\%$	$79.92 \pm 0.14\%$
S2M2_R [6]	ResNet18	$64.06 \pm 0.18\%$	$80.58 \pm 0.12\%$
LaplacianShot [38]	ResNet18	$72.11 \pm 0.19\%$	$82.31 \pm 0.14\%$
Transfer+Graph Interpolation (ours)	ResNet18	$\mathbf{72.40 \pm 0.24\%}$	$\mathbf{82.89 \pm 0.14\%}$
ProtoNet [12]	WRN	$62.60 \pm 0.20\%$	$79.97 \pm 0.14\%$
Matching Networks [11]	WRN	$64.03 \pm 0.20\%$	$76.32 \pm 0.16\%$
S2M2_R [6]	WRN	$64.93 \pm 0.18\%$	$83.18 \pm 0.11\%$
SimpleShot [36]	WRN	$65.87 \pm 0.20\%$	$82.09 \pm 0.14\%$
SIB [39]	WRN	$70.00 \pm 0.60\%$	$79.20 \pm 0.40\%$
BD-CSPN [40]	WRN	$70.31 \pm 0.93\%$	$81.89 \pm 0.60\%$
LaplacianShot [38]	WRN	$74.86 \pm 0.19\%$	$84.13 \pm 0.14\%$
Transfer+Graph Interpolation (ours)	WRN	$\mathbf{76.50 \pm 0.23\%}$	$\mathbf{85.23 \pm 0.13\%}$
		CUB	
Method	Backbone	1-shot	5-shot
S2M2_R [6]	ResNet18	$71.43 \pm 0.28\%$	$85.55 \pm 0.52\%$
ProtoNet [12]	ResNet18	$72.99 \pm 0.88\%$	$86.64 \pm 0.51\%$
Matching Networks [11]	ResNet18	$73.49 \pm 0.89\%$	$84.45 \pm 0.58\%$
LaplacianShot [38]	ResNet18	80.96%	88.68%
Transfer+Graph Interpolation (ours)	ResNet18	$\mathbf{86.05 \pm 0.20\%}$	$\mathbf{90.87 \pm 0.10\%}$
S2M2_R [6]	WRN	$80.68 \pm 0.81\%$	$90.85 \pm 0.44\%$
Transfer+Graph Interpolation (ours)	WRN	$\mathbf{88.35 \pm 0.19\%}$	$\mathbf{92.14 \pm 0.10\%}$
		miniImageNet⟶CUB	
Method	Backbone	1-shot	5-shot
Baseline++ [9]	ResNet18	$40.44 \pm 0.75\%$	$56.64 \pm 0.72\%$
SimpleShot [36]	ResNet18	48.56%	65.63%
LaplacianShot [38]	ResNet18	$\mathbf{55.46\%}$	66.33%
Transfer+Graph Interpolation (ours)	ResNet18	$51.67 \pm 0.24\%$	$\mathbf{69.83 \pm 0.18\%}$
Manifold Mixup [31]	WRN	$46.21 \pm 0.77\%$	$66.03 \pm 0.71\%$
S2M2_R [6]	WRN	$48.24 \pm 0.84\%$	$70.44 \pm 0.75\%$
Transfer+Graph Interpolation (ours)	WRN	$\mathbf{58.63 \pm 0.25\%}$	$\mathbf{73.46 \pm 0.17\%}$
		CIFAR-FS	
Method	Backbone	1-shot	5-shot
BD-CSPN [40]	WRN	$72.13 \pm 1.01\%$	$82.28 \pm 0.69\%$
S2M2_R [6]	WRN	$74.81 \pm 0.19\%$	$87.47 \pm 0.13\%$
SIB [39]	WRN	$80.00 \pm 0.60\%$	$85.30 \pm 0.40\%$
Transfer+Graph Interpolation (ours)	WRN	$\mathbf{83.90 \pm 0.22\%}$	$\mathbf{88.76 \pm 0.15\%}$

TABLE III
1-SHOT AND 5-SHOT PERFORMANCE (ON MINIIMAGENET) COMPARISON WITH OTHER GNN BASED METHODS. IN OUR EXPERIMENT WE USE THE SAME HYPERPARAMETERS AS TABLE II.

Method	1-shot	5-shot
GNN [2]	$50.33 \pm 0.36\%$	$66.41 \pm 0.63\%$
TPN [5]	$55.51 \pm 0.86\%$	$69.86 \pm 0.65\%$
wDAE-GNN [4]	$61.07 \pm 0.15\%$	$76.75 \pm 0.11\%$
Transfer+Graph Interpolation (ours)	$\mathbf{76.50 \pm 0.23\%}$	$\mathbf{85.23 \pm 0.13\%}$

TABLE IV
1-SHOT AND 5-SHOT ACCURACY ON MINIIMAGENET, WHEN USING THE WRN BACKBONE AND VARIOUS GRAPH NEURAL NETWORKS. WE USE THE SAME HYPERPARAMETERS AS TABLE II AND APPLY THEM TO ALL METHODS (WITH THE EXCEPTION OF κ FOR GCN AND GAT).

Method	1-shot	5-shot
Transfer+GAT	$65.38 \pm 0.89\%$	$76.00 \pm 0.67\%$
Transfer+GCN	$75.88 \pm 0.23\%$	$84.51 \pm 0.13\%$
Transfer+Graph Interpolation	$\mathbf{76.47 \pm 0.23\%}$	$\mathbf{85.23 \pm 0.13\%}$

*GAT is evaluated with 600 test runs.

matrix are likely to over-smooth the representations of neighbors.

3) When k is small (here: $k = 5$ or $k = 10$), there is little sensitivity to both α and κ (for $\kappa \leq 3$). This is an asset as it makes it simpler to find good hyperparameters.

4) The best results are achieved for smaller values of k, suggesting that cosine similarity between distant representations can be noisy and damaging to the performance of the method.
5) Note that in this experiment $s + Q/K_n = 16$. So using $k = 15$ would ideally select exactly 15 neighbors of the same class for each input. Interestingly, this choice of k does not lead to the best performance, showing the graph structure is not perfectly aligned with classes.

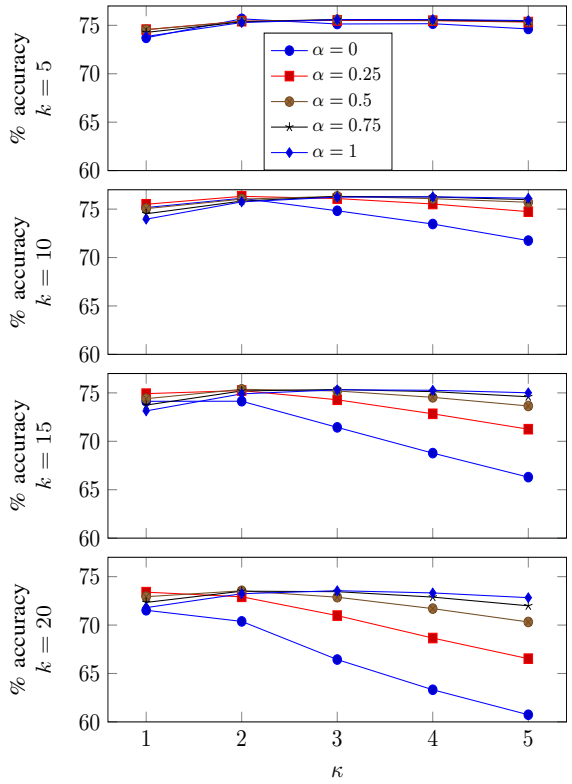


Fig. 4. Evolution of the accuracy of few-shot classification with miniImageNet (backbone: WRN) as a function of κ, α and k.

It is often disregarded the impact of class imbalance in the context of few-shot learning. As a matter of fact, since we only consider very few labeled examples, it does not make much sense to consider such a scenario. But in the context of transductive setting, it is highly probable that unlabeled inputs are imbalanced between classes. So we perform the next experiment by varying the number of examples chosen in two random classes from miniImageNet. We always make sure that the total number of queries to classify remains the same, that is 100. But we select q_1 of them in class 1 and $100 - q_1$ of them in class 2.

In Figure 5, we depict the evolution of the accuracy of the proposed method, as a function of q_1. As one can clearly see from this figure, there is an important influence of class imbalance towards the performance of the proposed method. This is expected as the generated graphs will have imbalanced communities as a consequence. This could be problematic to some application domains where such imbalance is expected to happen in considered datasets, as there is no direct way of correcting it. Obviously, if one has insights about the relative distribution between classes, simple data augmentation or sampling could be used for mitigation.

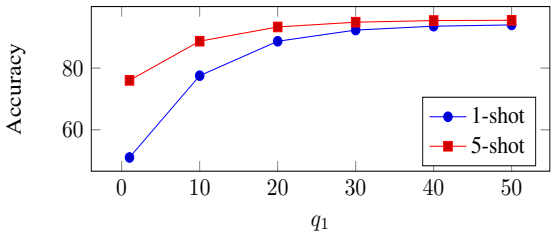


Fig. 5. Accuracy of 2-ways classification with unevenly distributed query data for each class, where the total number of query inputs remains constant. When $q_1 = 1$, we obtain the most imbalanced case, whereas $q_1 = 50$ corresponds to a balanced case. We use $\alpha = 0.5$, $\kappa = 3$ and $k = 10$.

However, this could be problematic to some application domains where such imbalance is expected to happen in considered datasets, as there is no direct way of correcting it. Obviously, if one has insights about the relative distribution between classes, simple data augmentation or sampling could be used for balancing this negative effect.

Finally, in Figure 6, we draw a representation of a typical graph obtained with the miniImageNet dataset, using Laplacian embedding [41], [42]. On this figure, we colored vertices depending on which class they belong to. Interestingly, this figure shows that some classes are easily separated in the graph, whereas others are much harder to discriminate. We believe that the main reason why these graphs are not perfectly segregating classes is because some dimensions obtained using the backbone are specialized on features completely irrelevant for the novel task.

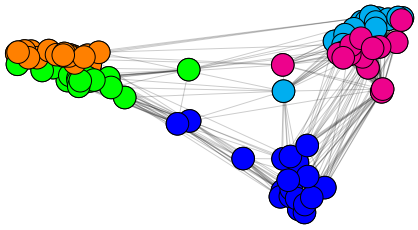


Fig. 6. Visualisation of a graph obtained using miniImageNet. Colors represent various classes. Vertices are placed close if they share many connections.

V. Conclusion

In this paper we introduced a novel method to solve the few-shot classification problem. It consists in combining three steps: a pretrained transfer, a graph-based interpolation technique and a logistic regression.

By performing experiments on standardized vision datasets, we obtained state-of-the-art results, with the most important gains in the case of 1-shot classification.

Interestingly, the proposed method requires to tune few hyperparameters, and these have a little impact on accuracy. We thus believe that it is an applicable solution to many practical problems.

There are still open questions to be addressed, such as the case of imbalanced classes, or settings where prediction must be performed on streaming data, one input at a time.

REFERENCES

[1] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[2] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," *arXiv preprint arXiv:1711.04043*, 2017.

[3] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11–20.

[4] S. Gidaris and N. Komodakis, "Generating classification weights with gnn denoising autoencoders for few-shot learning," *arXiv preprint arXiv:1905.01102*, 2019.

[5] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," *arXiv preprint arXiv:1805.10002*, 2018.

[6] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, and V. N. Balasubramanian, "Charting the right manifold: Manifold mixup for few-shot learning," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2218–2227.

[7] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 2012.

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

[9] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," 2019.

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

[11] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, 2016, pp. 3630–3638.

[12] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[13] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[14] H. Li, D. Eigen, S. Dodge, M. Zeiler, and X. Wang, "Finding task-relevant features for few-shot learning by category traversal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1–10.

[15] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 657–10 665.

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

[17] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. IEEE, 2005, pp. 729–734.

[18] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, 2015.

[19] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[20] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[23] F. Wu, T. Zhang, A. H. d. Souza Jr, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," *arXiv preprint arXiv:1902.07153*, 2019.

[24] H. Zhang, J. Zhang, and P. Koniusz, "Few-shot learning via saliency-guided hallucination of samples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2770–2779.

[25] Z. Chen, Y. Fu, Y.-X. Wang, L. Ma, W. Liu, and M. Hebert, "Image deformation meta-networks for one-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8680–8689.

[26] Y. Wang and Q. Yao, "Few-shot learning: A survey," *arXiv preprint arXiv:1904.05046*, 2019.

[27] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI Global, 2010, pp. 242–264.

[28] D. Das and C. G. Lee, "A two-stage approach to few-shot learning for image recognition," *IEEE Transactions on Image Processing*, 2019.

[29] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost," in *European Conference on Computer Vision*. Springer, 2012, pp. 488–501.

[30] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[31] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, A. Courville, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," *arXiv preprint arXiv:1806.05236*, 2018.

[32] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[34] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten, "Simpleshot: Revisiting nearest-neighbor classification for few-shot learning," *arXiv preprint arXiv:1911.04623*, 2019.

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

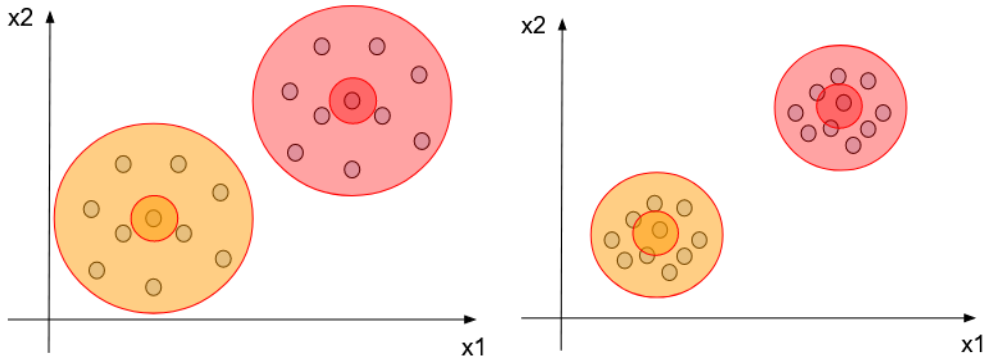
[38] I. M. Ziko, J. Dolz, E. Granger, and I. B. Ayed, "Laplacian regularized few-shot learning," *arXiv preprint arXiv:2006.15486*, 2020.

[39] S. X. Hu, P. G. Moreno, Y. Xiao, X. Shen, G. Obozinski, N. D. Lawrence, and A. Damianou, "Empirical bayes transductive meta-learning with synthetic gradients," *arXiv preprint arXiv:2004.12696*, 2020.

[40] J. Liu, L. Song, and Y. Qin, "Prototype rectification for few-shot learning," *arXiv preprint arXiv:1911.10713*, 2019.

[41] R. Horaud, "A short tutorial on graph laplacians, laplacian embedding, and spectral clustering," 2009.

[42] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.



(a) Before graph preprocessing **(b)** After graph preprocessing

Figure 28: Illustration of effects on preprocessing using graph on two clusters.

3.3 Discussions

Our proposed method is named "Transfer+Graph Interpolation". It is able to brought large increase in accuracy compared with selected baselines [Che+19a; Wan+19b] and reached competitive performance among methods in the transductive setting. This work has been cited and further extended by other works such as [LSA21; LAP21] that study the use of graph in a similar way. In this section we will address more details of our proposed method.

3.3.1 Rationale behind the use of graph

The rationale behind the use of graph neural networks can also be interpreted as a way of filtering high frequencies in signals. In signal processing, filtering is an operation that can be used to reduce the noise, with the reason being: if the signal is believed to yield a low-spectrum Fourier transform, then removing the large frequencies from it should remove mostly noise. In our paper, the graph we build connects samples that are similar in latent space. By diffusing the corresponding signals on this graph, we reduce the variation between samples that are connected, and thus we remove high (graph-)frequencies. So the main idea of using graph neural networks is to reduce the noise of each sample before classifying, which has the effect of densifying the local distributions of data.

3.3.2 Improved filtering by graph

Following the same kind of idea, in [Ham+21] we proposed a method to reduce intra-class variances by creating a graph per class. The proposed graph are used to design filters that remove high frequencies of samples belonging to the same class while keeping the expectation of the class prototype unchanged. Fig. 28 illustrates the effect of graph preprocessing on the extracted features. We can observe that the graph is able to reduce the cluster variance while keeping its prototype unchanged.

Compared with previously proposed Transfer+Graph Interpolation that applies graph on all samples of different clusters, this method filter samples of the same cluster, which reduces the confusion brought in by samples that do not belong to

the expected class. As experiments in the paper show, the newly designed graph gives slight improvement in the 5-shot setting.

3.3.3 Limitations

Using graphs to filter out noise is definitely not a novel idea, yet it demonstrated consistent results in our experiments. A main drawback of the method is that it requires to carefully design a graph to achieve the best accuracy gains. In the paper, the construction of the graph comes with 3 dedicated hyperparameters, and their tuning in practice may be difficult due to the absence of a validation set on the considered few-shot task. This drawback is mitigated by the fact our experiments show the tuning of hyperparameters is not very sensitive. In order to circumvent this limitation, a possible solution would be to find the best hyperparameters using proxy few-shot tasks, i.e. simulated few-shot tasks sampled from validation dataset.

Although the design of graph for processing features requires a lot of further research, another place where we can seek improvement is the classifier design. Given that in this paper we apply a logistic regression which is a discriminant method that 1) only uses support set samples for parameter training, which could potentially be improved by making use of samples in the query set as well; and 2) works better in the case of abundant labeled samples. In our subsequent work we look forward to propose an alternative approach based on clustering that attempts to model classes in the form of cluster and then perform classification based on the estimation of these clusters.

Chapter 4

Squeezing Backbone Feature Distributions to the Max for Efficient Few-Shot Learning

In this chapter we introduce a combination of two papers [HGP21b; HSS18] contributing to the feature preprocessing and classifier design steps in the pipeline. We present the general context and paper, then we discuss the contributions of our proposed method as well as limitations and perspectives.

4.1 Context

In our previous work we mainly focus on a preprocessing method based on graphs, followed by a logistic regression for classification. The preprocessing was mainly thought of as a mean to remove noise in an attempt to benefit the final accuracy.

However, another aspect of feature preprocessing that is worth to address is how the preprocessed features fit into the assumptions of a designed classifier. Given that we often follow Gaussian assumptions in few-shot classification, a more adapted preprocessing method could add extra benefit for the classifier. Unfortunately we find little work that discusses this point throughout the course of our research, this would be an interesting aspect for more further studies.

In addition, the design of the classifier is also a crucial part of the pipeline for good performance. Especially for methods based on clustering, the goal is to perform classification based on cluster estimations such as prototypes and covariance matrix. On this aspect, previous works have proposed several classifiers. For instance, in [Wan+19b] the authors use a Nearest Class Mean classifier that computes prototypes by averaging the labeled samples belonging to the same cluster. In [Ren+18; Bat+22] the authors apply a Soft-kmeans classifer that estimates prototypes by computing the weighted average of cluster assignment for all samples, where the class assignment for each sample is represented by probabilities. Considering a shared isotropic covariance matrix for all clusters, Soft-kmeans is able to reach competitive performance on transductive few-shot

classification. Another approach proposed in [Lic+20] is based on Mean Shift algorithm that estimates prototypes within a range of samples that are closed to the current ones.

Given promising results of previous works based on clustering methods. We seek further amelioration on the following two aspects: 1) feature distribution alignment with the classifier, and 2) the design of the classifier. Namely, in this work we introduce a two-step approach combining a preprocessing method aiming at making the feature distribution more gaussian-like, and an optimized clustering method using Optimal Transport (OT) meant to benefit from both support and query samples, achieving the state-of-the-art accuracy on many transductive vision few-shot classification benchmarks at the time it was published.

This work has been released in two successive papers: 1) [HGP21b]:

Leveraging the Feature Distribution in Transfer-Based Few-Shot Learning **Hu, Y.**, Gripon, V. and Pateux, S., 2021, September. In International Conference on Artificial Neural Networks (pp. 487-499). Springer, Cham.2021,

in which the code can be found at `https://github.com/yhu01/PT-MAP`; and 2) [HPG22a]:

Squeezing Backbone Feature Distributions to the Max for Efficient Few-Shot Learning **Hu, Y.**, Pateux, S. and Gripon, V., 2022. Algorithms, 15(5), p.147.2022,

in which the code can be found at `https://github.com/yhu01/BMS`.

In the first paper, we focused on the case where the number of queries in each class is known, and we propose an algorithm based on Optimal Transport in order for an optimal cluster assignment on the targeted samples. Note that we integrate OT into the Expectation Maximization framework so that the algorithm converges to a stable estimation. In the second paper, we adapted the methodology so that it is able to cope with settings where the distribution of query samples among classes can be arbitrary. In both cases, the preprocessing method uses a combination of Power Transform (PT) and sphering in order to reshape the feature distributions to be close-to-gaussian. The reason why PT is useful is because features are typically obtained after applying a rectified linear unit, which tends to produce truncated distributions that are dense around 0+. Then sphering allows us to focus on the angle rather than on the norm of obtained vectors. The proposed clustering methods are based on Optimal Transport (OT) aiming at allocating samples to classes with minimum cost while maximizing the posteriors. The first paper introduces a method called "PT+MAP" that largely boosted the transductive FSC performance in both 1 and 5 shot balanced settings. In [HPG22a] we proposed a modified version of the OT algorithm (called "PEME+BMS") that tries to limit the effect of priors about the query set.

4.2 Paper on leveraging feature distributions for maximum usage

algorithms

MDPI

Article

Squeezing Backbone Feature Distributions to the Max for Efficient Few-Shot Learning

Yuqing Hu [1,2,*], Stéphane Pateux [1] and Vincent Gripon [2]

[1] Orange S.A., F-84100 Paris, France; stephane.pateux@orange.com
[2] IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France; vincent.gripon@imt-atlantique.fr
* Correspondence: yuqing.hu@imt-atlantique.fr

Abstract: In many real-life problems, it is difficult to acquire or label large amounts of data, resulting in so-called few-shot learning problems. However, few-shot classification is a challenging problem due to the uncertainty caused by using few labeled samples. In the past few years, many methods have been proposed with the common aim of transferring knowledge acquired on a previously solved task, which is often achieved by using a pretrained feature extractor. As such, if the initial task contains many labeled samples, it is possible to circumvent the limitations of few-shot learning. A shortcoming of existing methods is that they often require priors about the data distribution, such as the balance between considered classes. In this paper, we propose a novel transfer-based method with a double aim: providing state-of-the-art performance, as reported on standardized datasets in the field of few-shot learning, while not requiring such restrictive priors. Our methodology is able to cope with both inductive cases, where prediction is performed on test samples independently from each other, and transductive cases, where a joint (batch) prediction is performed.

Keywords: few-shot learning; inductive and transductive learning; transfer learning; optimal transport

Citation: Hu, Y.; Pateux, S.; Gripon, V. Squeezing Backbone Feature Distributions to the Max for Efficient Few-Shot Learning. *Algorithms* **2022**, *15*, 147. https://doi.org/10.3390/a15050147

Academic Editors: Mounim A. El Yacoubi, Mehdi Ammi and Hui Yu

Received: 8 April 2022
Accepted: 21 April 2022
Published: 26 April 2022

1. Introduction

Thanks to their outstanding performance, deep learning methods have been widely considered for vision tasks such as image classification and object detection. In order to reach top performance, these systems are typically trained using very large labeled datasets that are representative enough of the inputs to be processed afterward.

However, in many applications, it is costly to acquire or annotate data, resulting in the impossibility of creating such large labeled datasets. Under this condition, it is challenging to optimize deep learning architectures considering the fact they typically are made of way more parameters than the dataset can efficiently tune. This is the reason why in the past few years, few-shot learning (i.e., the problem of learning with few labeled examples) has become a trending research subject in the field. In more detail, there are two settings that authors often consider: (a) "inductive few-shot", where only a few labeled samples are available during training, and prediction is performed on each test input independently, and (b) "transductive few-shot", where prediction is performed on a batch of (non-labeled) test inputs, allowing to take into account their joint distribution.

Few-shot learning is critical to many applications. To name a few, it has been considered for vision [1–3], audio [4–6], language [7–9], and medical imaging [10–12]. More generally, few-shot learning can be used to provide proofs-of-concept while limiting the costs of data labeling or to help in pseudo-annotation of datasets. This importance of the problem of few-shot learning explains the abundant literature across the recent years.

Many works in the domain are built based on a "learning to learn" guidance, where the pipeline is to train an optimizer [13–15] with different tasks of limited data so that the model is able to learn generic experience for novel tasks. Namely, the model learns a set of initialization parameters that are in an advantageous position for the model to adapt to a

new (small) dataset. Recently, the trend evolved towards using well-thought-out feature extractors, called backbones [1,2,16–19], that are trained one time on a large generic dataset in order to produce easily classified feature vectors.

A main problem of the existing methods is that they typically require priors about the data balance between considered classes to perform at their best [1,20]. These methods could be patched to work efficiently under other regimes but would still require the knowledge of data distribution between classes. In this work, we introduce a new methodology with a double aim: 1—providing state-of-the-art performance, as reported using standardized benchmarks in the field of few-shot learning, and 2—not requiring any priors about data distribution among classes.

To achieve this goal, we introduce a novel methodology, summarized in Figure 1, that combines feature preprocessing, self-distillation and an optimal transport-based framework. The utility of these ingredients is demonstrated using ablation tests.

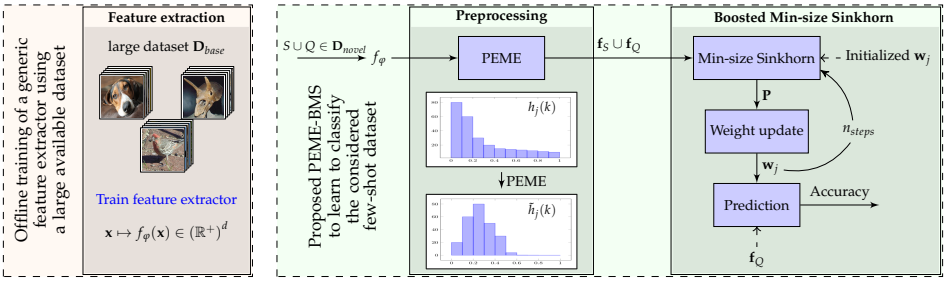


Figure 1. Illustration of the proposed method. A feature extractor is trained using a generic dataset. Obtained features on the few-shot dataset are then preprocessed using PEME (Power, Euclidian normalization, Mean subtraction, Euclidean normalization) to better align with a Gaussian distribution. They are then either directly fed to a classifier (inductive case), or processed through an optimal transport inspired algorithm using self-distillation and Boosted Min-Size Sinkhorn (transductive case).

The outline of the paper is as follows. In Section 2, we introduce related work and discuss the novelty of the proposed approach. In Section 3, we introduce the proposed methodology. Section 4 contains several experiments and benchmark results, along with corresponding discussions. Finally, Section 5 presents the conclusion.

2. Related Work

A large volume of works in few-shot classification is based on meta learning [15] methods, where the training data are transformed into few-shot learning episodes to better fit in the context of a few examples. In this branch, optimization-based methods [13–15,21–23] train a well-initialized optimizer so that it quickly adapts to unseen classes with a few epochs of training. Other works [24,25] apply data augmentation techniques to artificially increase the size of the training data in order for the model to generalize better to unseen data.

In the past few years, there has been a growing interest in transfer-based methods. The main idea consists of training feature extractors able to efficiently segregate novel classes it has never seen. For example, in [2,18], the authors train the backbone with a distance-based classifier [26] that takes into account the inter-class distance. In [2], the authors utilize self-supervised learning techniques [27] to co-train an extra rotation classifier for the output features, improving the accuracy in few-shot settings. More recent works adopt a two-stage training procedure [28–30] where the authors first batch-train a model, then use episodic training to better adjust class prototypes. There are also methods that train a model with a combination of different ingredients [31,32], e.g., distillation [33,34] under a teacher-student framework to better find the nuances between samples. Aside from approaches focused on training a more robust model, other approaches are built on top of a pre-trained feature extractor (backbone). For instance, in [35], the authors

implement a nearest class mean classifier to associate an input with a class whose centroid is the closest in terms of the ℓ_2 distance. In [20], an iterative approach is used to adjust the class prototypes. In [19], the authors build a graph neural network to gather the feature information from similar samples. Generally, transfer-based techniques often reach the best performance on standardized benchmarks.

Although many works involve feature extraction, few have explored the features in terms of their distribution [2,36,37]. Often, assumptions are made that the features in a class align to a certain distribution, even though these assumptions are seldom experimentally discussed. In our work, we analyze the impact of the feature distributions and how they can be transformed for better processing and accuracy. We also introduce a new algorithm to improve the quality of the association between input features and corresponding classes in typical few-shot settings.

Let us highlight the main contributions of this work. (1) We propose a novel preprocessing method to be applied to raw extracted features in order to make them more aligned with Gaussian assumptions. (2) We introduce a Wasserstein-based method to better align the distribution of features with that of the considered classes and combine it with self-distillation. (3) We show that the proposed method can bring a large increase in accuracy with a variety of feature extractors and datasets, leading to state-of-the-art results in the considered benchmarks. This work is an extended version of [1], with the main difference that here we consider the broader case where we do not know the proportion of samples belonging to each considered class in the case of a transductive few-shot, leading to a new algorithm called the Boosted Min-size Sinkhorn. We also propose more efficient preprocessing steps, leading to overall better performance in both inductive and transductive settings. Finally, we introduce the use of logistic regression with self-distillation in our methodology instead of a simple nearest class mean classifier.

3. Materials and Methods

In this section, we introduce the problem statement. We also discuss the various steps of the proposed method, including training the feature extractors, preprocessing the feature representations, and classifying them. Note that we made the code of our method available at https://github.com/yhu01/BMS (accessed on 1 February 2022).

3.1. Problem Statement

We consider a typical few-shot learning problem. Namely, we are given a *base* dataset $\mathbf{D}_{base}$ and a *novel* dataset $\mathbf{D}_{novel}$ such that $\mathbf{D}_{base} \cap \mathbf{D}_{novel} = \emptyset$. $\mathbf{D}_{base}$ contains a large number of labeled examples from K different classes and can be used to train a generic feature extractor. $\mathbf{D}_{novel}$, also referred to as a task or episode in other works, contains a small number of labeled examples (support set $\mathbf{S}$), along with some unlabeled ones (query set $\mathbf{Q}$), all from n *new* classes that are distinct from the K classes in $\mathbf{D}_{base}$. Our goal is to predict the classes of unlabeled examples in the query set. The following parameters are of particular importance to define such a few-shot problem: the number of classes in the novel dataset n (called n-way), the number of labeled samples per class s (called s-shot) and the number of unlabeled samples per class q. Therefore, the novel dataset contains a total of $l + u$ samples, where $l = ns$ are labeled, and $u = nq$ are unlabeled. In the case of an inductive few-shot, the prediction is performed independently on each one of the query samples. In the case of a transductive few-shot [20,38], the prediction is performed considering all unlabeled samples together. Contrary to our previous work [1], we do not consider knowing the proportion of samples in each class in the case of a transductive few-shot.

3.2. Feature Extraction

The first step is to train a neural network backbone model using only the base dataset. In this work, we consider multiple backbones with various training procedures. Once the considered backbone is trained, we obtain robust embeddings that should generalize well to novel classes. We denote by f_φ the backbone function, obtained by extracting the

output of the penultimate layer from the considered architecture, with φ being the trained architecture parameters. Thus, considering an input vector $\mathbf{x}$, $f_\varphi(\mathbf{x})$ is a feature vector with d dimensions that can be thought of as a simpler-to-manipulate representation of $\mathbf{x}$. Note that, importantly, in all backbone architectures used in the experiments of this work, the penultimate layers are obtained by applying a ReLU function so that all feature components coming out of f_φ are nonnegative.

3.3. Feature Preprocessing

As mentioned in Section 2, many works hypothesize, explicitly or not, that the features from the same class are aligned with a specific distribution (often Gaussian-like). However, this aspect is rarely experimentally verified. In fact, it is very likely that features obtained using the backbone architecture are not Gaussian. Indeed, usually, the features are obtained after applying a ReLU function [39] and exhibit a positive and yet skewed distribution mostly concentrated around 0 (more details can be found in the next section).

Multiple works in the domain [20,35] discuss the different statistical methods (e.g., batch normalization) to better fit the features into a model. Although these methods may have provable assets for some distributions, they could worsen the process if applied to an unexpected input distribution. This is why we propose to preprocess the obtained raw feature vectors so that they better align with typical distribution assumptions in the field. Denote $f_\varphi(\mathbf{x}) = [f_\varphi^1(\mathbf{x}), \ldots, f_\varphi^h(\mathbf{x}), \ldots, f_\varphi^d(\mathbf{x})] \in (\mathbb{R}^+)^d, \mathbf{x} \in \mathbf{D}_{novel}$ as the obtained features on $\mathbf{D}_{novel}$, and let $f_\varphi^h(\mathbf{x}), 1 \leq h \leq d$ denote its value in the hth position. The preprocessing methods applied in our proposed algorithms are as follows:

(E) Euclidean normalization. Also known as L2-normalization, which is widely used in many related works [19,35,37], this step scales the features to the same area so that large variance feature vectors do not predominate the others. Euclidean normalization can be given by:

$$f_\varphi(\mathbf{x}) \leftarrow \frac{f_\varphi(\mathbf{x})}{\|f_\varphi(\mathbf{x})\|_2} \tag{1}$$

(P) Power transform. The power transform method [1,40] simply consists of taking the power of each feature vector coordinate. The formula is given by:

$$f_\varphi^h(\mathbf{x}) \leftarrow (f_\varphi^h(\mathbf{x}) + \epsilon)^\beta, \quad \beta \neq 0 \tag{2}$$

where $\epsilon = 1 \times 10^{-6}$ is used to make sure that $f_\varphi(\mathbf{x}) + \epsilon$ is strictly positive in every position, and β is a hyper-parameter. The rationale of the preprocessing above is that power transform, often used in combination with euclidean normalization, has the functionality of reducing the skew of the distribution and mapping it to a close-to-Gaussian distribution, adjusted by β. After experiments, we found that $\beta = 0.5$ gives the most consistent results for our considered experiments, which corresponds to a square-root function that has a wide range of usage on features [41]. We will analyze this ability and the effect of power transform in more detail in Section 4. Note that power transform can only be applied if considered feature vectors contain nonnegative entries, which will always be the case in the remainder of this work.

(M) Mean subtraction. With mean subtraction, each sample is translated using $\mathbf{m} \in (\mathbb{R}^+)^d$, the projection center. This is often used in combination with euclidean normalization in order to reduce the task bias and better align the feature distributions [20]. The formula is given by:

$$f_\varphi(\mathbf{x}) \leftarrow f_\varphi(\mathbf{x}) - \mathbf{m} \tag{3}$$

The projection center is often computed as the mean values of feature vectors related to the problem [20,35]. In this paper, we compute it either as the mean feature vector of the base dataset (denoted as $\mathrm{M_b}$) or the mean vector of the novel dataset (denoted as $\mathrm{M_n}$), depending on the few-shot settings. Of course, in both of these cases, the rationale

is to consider a proxy to what would be the exact mean value of feature vectors on the considered task.

In our proposed method, we deploy these preprocessing steps in the following order: Power transform (P) on the raw features, followed by a Euclidean normalization (E). Then, we perform mean subtraction (M) followed by another Euclidean normalization at the end. The resulting abbreviation is PEME, in which M can be either M_b or M_n, as mentioned above. In our experiments, we found that using M_b in the case of inductive few-shot learning and M_n in the case of transductive few-shot learning consistently led to the most competitive results. More details on why we used this methodology are available in the experiment section.

When facing an inductive problem, a simple classifier such as a Nearest-Class-Mean classifier (NCM) can be used directly after this preprocessing step. The resulting methodology is denoted PEM_bE-NCM. However, in the case of transductive settings, we also introduce an iterative procedure, denoted BMS for Boosted Min-size Sinkhorn, meant to leverage the joint distribution of unlabeled samples. The resulting methodology is denoted PEM_nE-BMS. The details of the BMS procedure are presented thereafter.

3.4. Boosted Min-Size Sinkhorn

In the case of transductive few-shot, we introduce a method that consists of iteratively refining estimates for the probability each unlabeled sample belongs to any of the considered classes. This method is largely based on the one we introduced in [1], except it does not require priors about sample distributions in each of the considered classes. Denoting $i \in [1, \dots, l+u]$ as the sample index in $\mathbf{D}_{novel}$ and $j \in [1, \dots, n]$ as the class index, the goal is to maximize the following log post-posterior function:

$$\begin{aligned} L(\theta) &= \sum_i \log P(l(\mathbf{x}_i) = j | \mathbf{x}_i; \theta) \\ &= \sum_i \log \frac{P(\mathbf{x}_i, l(\mathbf{x}_i) = j; \theta)}{P(\mathbf{x}_i; \theta)} \\ &\propto \sum_i \log \frac{P(\mathbf{x}_i | l(\mathbf{x}_i) = j; \theta)}{P(\mathbf{x}_i; \theta)}, \end{aligned} \tag{4}$$

Here, $l(\mathbf{x}_i)$ denotes the class label for sample $\mathbf{x}_i \in \mathbf{Q} \cup \mathbf{S}$, $P(\mathbf{x}_i; \theta)$ denotes the marginal probability, and θ represents the model parameters to estimate. Assuming a Gaussian distribution on the input features for each class, here we define $\theta = \mathbf{w}_j, \forall j$ where $\mathbf{w}_j \in \mathbb{R}^d$ stand for the weight parameters for class j. We observe that Equation (4) can be related to the cost function utilized in optimal transport [42], which is often considered to solve classification problems, with constraints on the sample distribution over classes. To that end, a well-known Sinkhorn [43] mapping method is proposed. The algorithm aims at computing a class allocation matrix among novel class data for a minimum Wasserstein distance. Namely, an allocation matrix $\mathbf{P} \in \mathbb{R}_+^{(l+u)\times n}$ is defined where $\mathbf{P}[i, j]$ denotes the assigned portion for sample i to class j, and it is computed as follows:

$$\begin{aligned} \mathbf{P} &= \text{Sinkhorn}(\mathbf{C}, \mathbf{p}, \mathbf{q}, \lambda) \\ &= \underset{\tilde{\mathbf{P}} \in \mathbb{U}(\mathbf{p}, \mathbf{q})}{\text{argmin}} \sum_{ij} \tilde{\mathbf{P}}[i, j]\mathbf{C}[i, j] + \lambda H(\tilde{\mathbf{P}}), \end{aligned} \tag{5}$$

where $\mathbb{U}(\mathbf{p}, \mathbf{q}) \in \mathbb{R}_+^{(l+u)\times n}$ is a set of positive matrices for which the rows sum to $\mathbf{p}$ and the columns sum to $\mathbf{q}$, $\mathbf{p}$ denotes the distribution of the amount that each sample uses for class allocation, and $\mathbf{q}$ denotes the distribution of the amount of samples allocated to each class. Therefore, $\mathbb{U}(\mathbf{p}, \mathbf{q})$ contains all the possible ways of allocation. In the same equation, $\mathbf{C}$ can be viewed as a cost matrix that is of the same size as $\mathbf{P}$, each element in $\mathbf{C}$ indicates the cost of its corresponding position in $\mathbf{P}$. We will define the particular formula of the cost function for each position $\mathbf{C}[i, j], \forall i, j$ in details later on in the section. As for the second term on

the right of (5), it stands for the entropy of $\tilde{\mathbf{P}}$: $H(\tilde{\mathbf{P}}) = -\sum_{ij} \tilde{\mathbf{P}}[i,j] \log \tilde{\mathbf{P}}[i,j]$, regularized by a hyper-parameter λ. Increasing λ would force the entropy to become smaller, so that the mapping is less diluted. This term also makes the objective function strictly convex [43,44] and thus a practical and effective computation. From lemma 2 in [43], the result of the Sinkhorn allocation has the typical form $\mathbf{P} = \text{diag}(\mathbf{u}) \cdot \exp(-\mathbf{C}/\lambda) \cdot \text{diag}(\mathbf{v})$. It is worth noting that here we assume a soft class allocation, meaning that each sample can be "sliced" into different classes. We will present our proposed method in detail in the following paragraphs.

Given all that is presented above, in this paper, we propose an Expectation–Maximization (*EM*) [45] based method, which alternates between updating the allocation matrix $\mathbf{P}$ and estimating the parameter θ of the designed model, in order to minimize Equation (5) and maximize Equation (4). For a starter, we define a weight matrix $\mathbf{W}$ with n columns (i.e., one per class) and d rows (i.e., one per dimension of feature vectors), and for column j in $\mathbf{W}$, we denote it as the weight parameters $\mathbf{w}_j \in \mathbb{R}^d$ for class j in correspondence with Equation (4). It is initialized as follows:

$$\mathbf{w}_j = \mathbf{W}[:,j] = \mathbf{c}_j / \|\mathbf{c}_j\|_2, \tag{6}$$

where

$$\mathbf{c}_j = \frac{1}{s} \sum_{\mathbf{x} \in \mathbf{S}, \ell(\mathbf{x}) = j} f_\varphi(\mathbf{x}). \tag{7}$$

We can see that $\mathbf{W}$ contains the average of feature vectors in the support set for each class, followed by a L2-normalization on each column so that $\|\mathbf{w}_j\|_2 = 1, \forall j$.

Then, we iterate multiple steps that we describe thereafter.

a Computing costs

As previously stated, the proposed algorithm is an *EM*-like one that iterately updates model parameters for optimal estimates. Therefore, this step, along with Min-size Sinkhorn presented in the next step, is considered as the *E*-step of our proposed method. The goal is to find membership probabilities for the input samples; namely, we compute $\mathbf{P}$ that minimizes Equation (5).

Here, we assume Gaussian distributions, and features in each class have the same variance and are independent from one another (covariance matrix $\mathbf{\Sigma} = \mathbf{I}\sigma^2$). We observe that, ignoring the marginal probability, Equation (4) can be boiled down to negative L2 distances between extracted samples $f_\varphi(\mathbf{x}_i), \forall i$ and $\mathbf{w}_j, \forall j$, which is initialized in Equation (6) in our proposed method. Therefore, based on the fact that $\mathbf{w}_j$ and $f_\varphi(\mathbf{x}_i)$ are both normalized to be unit length vectors ($f_\varphi(\mathbf{x}_i)$ being preprocessed using PEME introduced in the previous section), here we define the cost between sample i and class j to be the following equation:

$$\begin{aligned} \mathbf{C}[i,j] &\propto (f_\varphi(\mathbf{x}_i) - \mathbf{w}_j)^2 \\ &= 1 - \mathbf{w}_j^T f_\varphi(\mathbf{x}_i), \end{aligned} \tag{8}$$

which corresponds to the cosine distance.

b Min-size Sinkhorn

In [1], we proposed a Wasserstein distance-based method in which the Sinkhorn algorithm is applied at each iteration so that the class prototypes are updated iteratively in order to find their best estimates. Although the method showed promising results, it is established on the condition that the distribution of the query set is known, e.g., a uniform distribution among classes on the query set. This is not ideal, given the fact that any priors about $\mathbf{Q}$ should be supposedly kept unknown when applying a method. The methodology introduced in this paper can be seen as a generalization of that introduced in [1] that does not require priors about $\mathbf{Q}$.

In the classical settings, the Sinkhorn algorithm aims at finding the optimal matrix $\mathbf{P}$, given the cost matrix $\mathbf{C}$ and regulation parameter λ presented in Equation (4)). Typically, it initiates $\mathbf{P}$ from a softmax operation over the rows in $\mathbf{C}$, then it iterates between normalizing

columns and rows of $\mathbf{P}$, until the resulting matrix becomes close to doubly stochastic according to $\mathbf{p}$ and $\mathbf{q}$. However, in our case, we do not know the distribution of samples over classes. To address this, we firstly introduce the parameter k, initialized so that $k \leftarrow s$, meant to track an estimate of the cardinal of the class containing the least number of samples in the considered task. Then, we propose the following modification to be applied to the matrix $\mathbf{P}$ once initialized: we normalize each row as in the classical case but only normalize the columns of $\mathbf{P}$ for which the sum is less than the previously computed min-size k [20]. This ensures at least k elements are allocated for each class, but not exactly k samples as in the balanced case.

The principle of this modified Sinkhorn solution is presented in Algorithm 1.

Algorithm 1 Min-size Sinkhorn

Inputs: $\mathbf{C}, \mathbf{p} = \mathbf{1}_{l+u}, \mathbf{q} = k\mathbf{1}_n, \lambda$
Initializations: $\mathbf{P} = Softmax(-\lambda \mathbf{C})$
for $iter = 1$ **to** 50 **do**
  $\mathbf{P}[i,:] \leftarrow \mathbf{p}[i] \cdot \frac{\mathbf{P}[i,:]}{\sum_j \mathbf{P}[i,j]}, \forall i$
  $\mathbf{P}[:,j] \leftarrow \mathbf{q}[j] \cdot \frac{\mathbf{P}[:,j]}{\sum_i \mathbf{P}[i,j]}$ if $\sum_i \mathbf{P}[i,j] < \mathbf{q}[j], \forall j$
end for
return P

c Updating weights

This step is considered as the M-step of the proposed algorithm, in which we use a variant of the logistic regression algorithm in order to find the model parameter θ in the form of weight parameters $\mathbf{w}_j$ for each class. Note that $\mathbf{w}_j$, if normalized, is equivalent to the prototype for class j in this case. Given the fact that in Equation (4), we also take into account the marginal probability, it can be further broken down as:

$$P(\mathbf{x}_i;\theta) = \sum_j P(\mathbf{x}_i | l(\mathbf{x}_i) = j;\theta) P(l(\mathbf{x}_i) = j), \tag{9}$$

We observe that Equation (4) corresponds to applying a softmax function on the negative logits computed through an L2-distance function between samples and class prototypes (normalized). This fits the formulation of a linear hypothesis between $f_\varphi(\mathbf{x}_i)$ and $\mathbf{w}_j$ for logit calculations, hence the rationale for utilizing logistic regression in our proposed method. Note that contrary to classical logistical regression, we implement here a form of self-distillation. Indeed, we use soft labels contained in $\mathbf{P}$ instead of one-hot class indicator targets, and these targets are refined iteratively.

The procedure of this step is as follows: now that we have a polished allocation matrix $\mathbf{P}$, we firstly initialize the weights $\mathbf{w}_j$ as follows:

$$\mathbf{w}_j \leftarrow \mathbf{u}_j / \|\mathbf{u}_j\|_2, \tag{10}$$

where

$$\mathbf{u}_j \leftarrow \sum_i \mathbf{P}[i,j] f_\varphi(\mathbf{x}_i) / \sum_i \mathbf{P}[i,j]. \tag{11}$$

We can see that elements in $\mathbf{P}$ are used as coefficients for feature vectors to linearly adjust the class prototypes [1]. Similar to Equation (6), here $\mathbf{w}_j$ is the normalized newly-computed class prototype that is a vector of length 1.

Next, we further adjust weights by applying a logistic regression, and the optimization is performed by minimizing the following loss:

$$\frac{1}{l+u} \cdot \sum_i \sum_j \; - \log\left(\frac{\exp(\mathbf{S}[i,j])}{\sum_{\gamma=1}^{n} \exp(\mathbf{S}[i,\gamma])}\right) \cdot \mathbf{P}[i,j], \tag{12}$$

where $\mathbf{S} \in \mathbb{R}^{(l+u)\times n}$ contains the logits, and each element is computed as:

$$\mathbf{S}[i,j] = \kappa \cdot \frac{\mathbf{w}_j^T f_\varphi(\mathbf{x}_i)}{\|\mathbf{w}_j\|_2}. \tag{13}$$

Note that κ is a scaling parameter, it can also be seen as a temperature parameter that adjusts the confidence metric to be associated with each sample. It is learnt jointly with $\mathbf{W}$.

The deployed logistic regression comes with hyperparameters on its own. In our experiments, we use an SGD optimizer with a gradient step of 0.1 and 0.8 as the momentum parameter, and we train over e epochs. Here, we point out that $e \geq 0$ is considered an influential hyperparameter in our proposed algorithm, $e = 0$ indicates a simple update of $\mathbf{W}$ as the normalized adjusted class prototypes (Equation (10)) computed from $\mathbf{P}$ in Equation (11), without further adjustment of logistic regression. In addition, note that when $e > 0$, we project columns of $\mathbf{W}$ to the unit hypersphere at the end of each epoch.

d Estimating the class minimum size

We can now refine our estimate for the min-size k for the next iteration. To this end, we firstly compute the predicted label of each sample as follows:

$$\hat{\ell}(\mathbf{x}_i) = \arg\max_j(\mathbf{P}[i,j]), \tag{14}$$

which can be seen as the current (temporary) class prediction.

Then, we compute:

$$k = \min_j \{k_j\}, \tag{15}$$

where $k_j = \#\{i, \hat{\ell}(\mathbf{x}_i) = j\}$, $\#\{\cdot\}$ representing the cardinal of a set.

Summary of the proposed method: all steps of the proposed method are summarized in Algorithm 2. In our experiments, we also report the results obtained when using a prior about $\mathbf{Q}$ as in [1]. In this case, k does not have to be estimated throughout the iterations and can be replaced with the actual exact targets for the Sinkhorn. We denote this prior-dependent version PEM_nE-BMS* (with an added $*$).

Algorithm 2 Boosted Min-size Sinkhorn (BMS)

Parameters: λ, e
Inputs: Preprocessed $f_\varphi(\mathbf{x}), \forall \mathbf{x} \in \mathbf{D}_{novel} = \mathbf{Q} \cup \mathbf{S}$
Initializations: $\mathbf{W}$ as normalized mean vectors over the support set for each class (Equation (6)); Min-size $k \leftarrow s$.
for $iter = 1$ **to** 20 **do**
  Compute cost matrix $\mathbf{C}$ using $\mathbf{W}$ (Equation (8)). # E-step
  Apply Min-size Sinkhorn to compute $\mathbf{P}$ (Algorithm 1). # E-step
  Update weights $\mathbf{W}$ using $\mathbf{P}$ with logistic regression (Equations (10)–(13)). # M-step
  Estimate class predictions $\hat{\ell}$ and min-size k using $\mathbf{P}$ (Equations (14) and (15)).
end for
return $\hat{\ell}$

3.5. Implementation Details

In order to stress the genericity of our proposed method with regards to the chosen backbone architecture and training strategy, we perform experiments using WRN [46], ResNet18 and ResNet12 [47], along with some other pretrained backbones (e.g., DenseNet [35,48]). For each dataset, we train the feature extractor with base classes and test the performance using novel classes. Therefore, for each test run, n classes are drawn uniformly at random among novel classes. Among these n classes, s labeled examples and q unlabeled examples per class are uniformly drawn at random to form $\mathbf{D}_{novel}$. The WRN and ResNet are trained following [2]. In the inductive setting, we use our proposed preprocessing steps PEM_bE

followed by a basic Nearest Class Mean (NCM) classifier. In the transductive setting, the preprocessing steps are denoted as PEM_nE in that we use the mean vector of a novel dataset for mean subtraction, followed by BMS or BMS* depending on whether we have prior knowledge onf the distribution of query set **Q** among classes. Note that we perform a QR decomposition on preprocessed features in order to speed up the computation for the classifier that follows. All our experiments are performed using $n = 5, q = 15, s = 1$ or 5. In our experiments, we perform 10,000 random runs to obtain the mean accuracy score and indicate confidence scores (95%) when relevant. For our proposed PEM_nE-BMS, we train $e = 0$ epoch in the case of 1-shot and $e = 40$ epochs in the case of 5-shot. As for PEM_nE-BMS*, we set $e = 20$ for 1-shot and $e = 40$ for 5-shot. As for the regularization parameter λ in Equation (5), it is fixed to 8.5 for all settings. The impact of these hyperparameters is detailed in the next sections.

4. Results and Discussions

4.1. Comparison with State-of-the-Art Methods

Performance on standardized benchmarks: in the first experiment, we conduct our proposed method on different benchmarks and compare the performance with other state-of-the-art solutions. The results are presented in Tables 1 and 2, and we observe that our method reaches the state-of-the-art performance in both inductive and transductive settings on all the few-shot classification benchmarks. Particularly, the proposed PEM_nE-BMS* brings important gains in both 1-shot and 5-shot settings, and the prior-independent PEM_nE-BMS also obtains competitive results on 5-shot. Note that for tieredImageNet we implement our method based on a pre-trained DenseNet121 backbone following the procedure described in [35]. From these experiments, we conclude that the proposed method can bring an increase in accuracy with a variety of backbones and datasets, leading to a state-of-the-art performance. In terms of execution time, we measured an average of 0.004 s per run. These results confirm the ability of the proposed methodology to reach state-of-the-art performance using the standardized benchmarks of the field of few-shot learning.

Table 1. The 1-shot and 5-shot accuracy of state-of-the-art methods in the literature on miniImageNet and tieredImageNet, compared with the proposed solution. Best results are in bold.

			miniImageNet	
Setting	**Method**	**Backbone**	**1-Shot**	**5-Shot**
Inductive	Matching Networks [49]	WRN	$64.03 \pm 0.20\%$	$76.32 \pm 0.16\%$
	SimpleShot [35]	DenseNet121	$64.29 \pm 0.20\%$	$81.50 \pm 0.14\%$
	S2M2_R [2]	WRN	$64.93 \pm 0.18\%$	$83.18 \pm 0.11\%$
	PT + NCM [1]	WRN	$65.35 \pm 0.20\%$	$83.87 \pm 0.13\%$
	DeepEMD[29]	ResNet12	$65.91 \pm 0.82\%$	$82.41 \pm 0.56\%$
	FEAT[28]	ResNet12	$66.78 \pm 0.20\%$	$82.05 \pm 0.14\%$
	PEM_bE-NCM (ours)	WRN	$\mathbf{68.43 \pm 0.20\%}$	$\mathbf{84.67 \pm 0.13\%}$
Transductive	BD-CSPN [50]	WRN	$70.31 \pm 0.93\%$	$81.89 \pm 0.60\%$
	LaplacianShot [51]	DenseNet121	$75.57 \pm 0.19\%$	$87.72 \pm 0.13\%$
	Transfer + SGC [19]	WRN	$76.47 \pm 0.23\%$	$85.23 \pm 0.13\%$
	TAFSSL [20]	DenseNet121	$77.06 \pm 0.26\%$	$84.99 \pm 0.14\%$
	TIM-GD [52]	WRN	77.80%	87.40%
	MCT [53]	ResNet12	$78.55 \pm 0.86\%$	$86.03 \pm 0.42\%$
	EPNet [54]	WRN	$79.22 \pm 0.92\%$	$88.05 \pm 0.51\%$
	PT + MAP [1]	WRN	$82.92 \pm 0.26\%$	$88.82 \pm 0.13\%$
	PEM_nE-BMS (ours)	WRN	$82.07 \pm 0.25\%$	$89.51 \pm 0.13\%$
	PEM_nE-BMS* (ours)	WRN	$\mathbf{83.35 \pm 0.25\%}$	$\mathbf{89.53 \pm 0.13\%}$

Table 1. *Cont.*

			tieredImageNet	
Setting	**Method**	**Backbone**	**1-Shot**	**5-Shot**
Inductive	ProtoNet [55]	ConvNet4	53.31 ± 0.89%	72.69 ± 0.74%
	LEO [56]	WRN	66.33 ± 0.05%	81.44 ± 0.09%
	SimpleShot [35]	DenseNet121	71.32 ± 0.22%	86.66 ± 0.15%
	PT + NCM [1]	DenseNet121	69.96 ± 0.22%	86.45 ± 0.15%
	FEAT[28]	ResNet12	70.80 ± 0.23%	84.79 ± 0.16%
	DeepEMD[29]	ResNet12	71.16 ± 0.87%	86.03 ± 0.58%
	RENet[30]	ResNet12	71.61 ± 0.51%	85.28 ± 0.35%
	PEM_bE-NCM (ours)	DenseNet121	**71.86** ± **0.21**%	**87.09** ± **0.15**%
Transductive	BD-CSPN [50]	WRN	78.74 ± 0.95%	86.92 ± 0.63%
	LaplacianShot [51]	DenseNet121	80.30 ± 0.22%	87.93 ± 0.15%
	MCT [53]	ResNet12	82.32 ± 0.81%	87.36 ± 0.50%
	TIM-GD [52]	WRN	82.10%	89.80%
	TAFSSL [20]	DenseNet121	84.29 ± 0.25%	89.31 ± 0.15%
	PT + MAP [1]	DenseNet121	85.75 ± 0.26%	90.43 ± 0.14%
	PEM_nE-BMS (ours)	DenseNet121	85.08 ± 0.25%	91.08 ± 0.14%
	PEM_nE-BMS* (ours)	DenseNet121	**86.07** ± **0.25**%	**91.09** ± **0.14**%

Table 2. The 1-shot and 5-shot accuracy of state-of-the-art methods on CUB and CIFAR-FS. Best results are in bold.

			CUB	
Setting	**Method**	**Backbone**	**1-Shot**	**5-Shot**
Inductive	Baseline++ [18]	ResNet10	69.55 ± 0.89%	85.17 ± 0.50%
	MAML [13]	ResNet10	70.32 ± 0.99%	80.93 ± 0.71%
	ProtoNet [55]	ResNet18	72.99 ± 0.88%	86.64 ± 0.51%
	Matching Networks [49]	ResNet18	73.49 ± 0.89%	84.45 ± 0.58%
	FEAT[28]	ResNet12	73.27 ± 0.22%	85.77 ± 0.14%
	DeepEMD[29]	ResNet12	75.65 ± 0.83%	88.69 ± 0.50%
	RENet[30]	ResNet12	79.49 ± 0.44%	91.11 ± 0.24%
	S2M2_R [2]	WRN	80.68 ± 0.81%	90.85 ± 0.44%
	PT + NCM [1]	WRN	80.57 ± 0.20%	91.15 ± 0.10%
	PEM_bE-NCM (ours)	WRN	**80.82** ± **0.19**%	**91.46** ± **0.10**%
Transductive	LaplacianShot [51]	ResNet18	80.96%	88.68%
	TIM-GD [52]	ResNet18	82.20%	90.80%
	BD-CSPN [50]	WRN	87.45%	91.74%
	Transfer + SGC [19]	WRN	88.35 ± 0.19%	92.14 ± 0.10%
	PT + MAP [1]	WRN	91.55 ± 0.19%	93.99 ± 0.10%
	LST + MAP [57]	WRN	91.68 ± 0.19%	94.09 ± 0.10%
	PEM_nE-BMS (ours)	WRN	91.01 ± 0.19%	94.60 ± 0.09%
	PEM_nE-BMS* (ours)	WRN	**91.91** ± **0.18**%	**94.62** ± **0.09**%
			CIFAR-FS	
Setting	**Method**	**Backbone**	**1-Shot**	**5-Shot**
Inductive	ProtoNet [55]	ConvNet64	55.50 ± 0.70%	72.00 ± 0.60%
	MAML [13]	ConvNet32	58.90 ± 1.90%	71.50 ± 1.00%
	RENet[30]	ResNet12	74.51 ± 0.46%	86.60 ± 0.32%
	BD-CSPN [50]	WRN	72.13 ± 1.01%	82.28 ± 0.69%
	S2M2_R [2]	WRN	74.81 ± 0.19%	87.47 ± 0.13%
	PT + NCM [1]	WRN	74.64 ± 0.21%	87.64 ± 0.15%
	PEM_bE-NCM (ours)	WRN	**74.84** ± **0.21**%	**87.73** ± **0.15**%

Table 2. *Cont.*

Setting	Method	Backbone	CIFAR-FS 1-Shot	CIFAR-FS 5-Shot
Transductive	DSN-MR [58]	ResNet12	78.00 ± 0.90%	87.30 ± 0.60%
	Transfer + SGC [19]	WRN	83.90 ± 0.22%	88.76 ± 0.15%
	MCT [53]	ResNet12	87.28 ± 0.70%	90.50 ± 0.43%
	PT + MAP [1]	WRN	87.69 ± 0.23%	90.68 ± 0.15%
	LST + MAP [57]	WRN	87.79 ± 0.23%	90.73 ± 0.15%
	PEM_nE-BMS (ours)	WRN	86.93 ± 0.23%	91.18 ± 0.15%
	PEM_nE-BMS* (ours)	WRN	**87.83 ± 0.22**%	**91.20 ± 0.15**%

Performance on cross-domain settings: in this experiment, we test our method in a cross-domain setting, where the backbone is trained with the base classes in miniImageNet but tested with the novel classes in the CUB dataset. As shown in Table 3, the proposed method gives the best accuracy both in the case of 1-shot and 5-shot, for both inductive and transductive settings. The ability of the proposed methodology to leverage feature vectors trained on a different dataset points out that its efficacy is not restricted to constrained settings where data distribution between the base and novel have to be identical.

Table 3. The 1-shot and 5-shot accuracy of state-of-the-art methods when performing cross-domain classification (backbone: WRN). Best results are in bold.

Setting	Method	1-Shot	5-Shot
Inductive	Baseline++ [18]	40.44 ± 0.75%	56.64 ± 0.72%
	Manifold Mixup [59]	46.21 ± 0.77%	66.03 ± 0.71%
	S2M2_R [2]	48.24 ± 0.84%	70.44 ± 0.75%
	PT + NCM [1]	48.37 ± 0.19%	70.22 ± 0.17%
	PEM_bE-NCM (ours)	**50.71 ± 0.19**%	**73.15 ± 0.16**%
Transductive	LaplacianShot [51]	55.46%	66.33%
	Transfer + SGC [19]	58.63 ± 0.25%	73.46 ± 0.17%
	PT + MAP [1]	63.17 ± 0.31%	76.43 ± 0.19%
	PEM_nE-BMS (ours)	62.93 ± 0.28%	79.10 ± 0.18%
	PEM_nE-BMS* (ours)	**63.90 ± 0.31**%	**79.15 ± 0.18**%

4.2. Ablation Studies

Ablation study on the proposed method: in this section, we have a closer look at the impact of our proposed methodology steps. The idea is to better understand the contribution of each step to the final performance. Namely, we conduct an ablation study on the prediction accuracy with or without (1) PEME, which is the proposed preprocessing steps on extracted raw features, and (2) proposed Boosted Min-sized Sinkhorn algorithm that integrates self-distillation for refined prototypes. Note that in the case of BMS*, the algorithm is equivalent to MAP presented in [1] without the newly proposed self-distillation method. In Table 4, we can see that both PEME and self-distillation play an important role in improving the prediction performance. As such, this experiment supports the interest of both steps to reach the best possible accuracy.

Table 4. Ablation study on our proposed PEME and BMS* with self-distillation on miniImageNet (backbone: WRN). Best results are in bold.

w/PEME	BMS* w/Self-Distillation	Accuracy 1-Shot	Accuracy 5-Shot
		$75.60 \pm 0.29\%$	$84.13 \pm 0.16\%$
✓		$82.92 \pm 0.26\%$	$88.82 \pm 0.13\%$
	✓	$80.19 \pm 0.27\%$	$87.40 \pm 0.13\%$
✓	✓	$\mathbf{83.35 \pm 0.25}\%$	$\mathbf{89.53 \pm 0.13}\%$

Generalization to backbone architectures. To further stress the interest of the ingredients inn the proposed method reaching top performance, in Table 5 we investigate the impact of our proposed method on different backbone architectures and benchmarks in the transductive setting. For comparison purposes, we also replace our proposed BMS algorithm with a standard K-Means algorithm where class prototypes are initialized with the available labeled samples for each class. We can observe that: (1) the proposed method consistently achieves the best results for any fixed backbone architecture, (2) the feature extractor trained on WRN outperforms the others with our proposed method on different benchmarks, (3) there are significant drops in accuracy with k-means, which stresses the interest of BMS, and (4) the prior on **Q** (BMS vs. BMS*) is of major interest for 1-shot, boosting the performance by an approximation of 1% on all tested feature extractors. Overall, these experiments demonstrate the interest of the proposed methodology with respect to existing alternatives.

Table 5. The 1-shot and 5-shot accuracy of the proposed method on different backbones and benchmarks. Comparison with the k-means algorithm. Best results are in bold.

Method	Backbone	miniImageNet 1-Shot	miniImageNet 5-Shot	CUB 1-Shot	CUB 5-Shot	CIFAR-FS 1-Shot	CIFAR-FS 5-Shot
K-MEANS	ResNet12	$72.73 \pm 0.23\%$	$84.05 \pm 0.14\%$	$87.35 \pm 0.19\%$	$92.31 \pm 0.10\%$	$78.39 \pm 0.24\%$	$85.73 \pm 0.16\%$
	ResNet18	$73.08 \pm 0.22\%$	$84.67 \pm 0.14\%$	$87.16 \pm 0.19\%$	$91.97 \pm 0.09\%$	$79.95 \pm 0.23\%$	$86.74 \pm 0.16\%$
	WRN	$76.67 \pm 0.22\%$	$86.73 \pm 0.13\%$	$88.28 \pm 0.19\%$	$92.37 \pm 0.10\%$	$83.69 \pm 0.22\%$	$89.19 \pm 0.15\%$
BMS (ours)	ResNet12	$77.62 \pm 0.28\%$	$86.95 \pm 0.15\%$	$90.14 \pm 0.19\%$	$94.30 \pm 0.10\%$	$81.65 \pm 0.25\%$	$88.38 \pm 0.16\%$
	ResNet18	$79.30 \pm 0.27\%$	$87.94 \pm 0.14\%$	$90.50 \pm 0.19\%$	$94.29 \pm 0.09\%$	$84.16 \pm 0.24\%$	$89.39 \pm 0.15\%$
	WRN	$82.07 \pm 0.25\%$	$89.51 \pm 0.13\%$	$91.01 \pm 0.18\%$	$94.60 \pm 0.09\%$	$86.93 \pm 0.23\%$	$91.18 \pm 0.15\%$
BMS* (ours)	ResNet12	$79.03 \pm 0.28\%$	$87.01 \pm 0.15\%$	$91.34 \pm 0.19\%$	$94.32 \pm 0.09\%$	$82.87 \pm 0.27\%$	$88.43 \pm 0.16\%$
	ResNet18	$80.56 \pm 0.27\%$	$87.98 \pm 0.14\%$	$91.39 \pm 0.19\%$	$94.31 \pm 0.09\%$	$85.17 \pm 0.25\%$	$89.42 \pm 0.16\%$
	WRN	$\mathbf{83.35 \pm 0.25}\%$	$\mathbf{89.53 \pm 0.13}\%$	$\mathbf{91.91 \pm 0.18}\%$	$\mathbf{94.62 \pm 0.09}\%$	$\mathbf{87.83 \pm 0.22}\%$	$\mathbf{91.20 \pm 0.15}\%$

Preprocessing impact: in Table 6, we compare our proposed feature preprocessing PEME with other preprocessing techniques such as batch normalization, which standardizes extracted feature values into $[0, 1]$ for a considered task, along with other ones being used in [35]. The experiment is conducted on miniImageNet (backbone: WRN). For all that is put into comparison, we run either an NCM classifier or BMS after preprocessing, depending on the settings. The obtained results clearly show the interest of PEME compared with existing alternatives, and we also observe that the power transform helps increase the accuracy on both inductive and transductive settings.

Table 6. Comparison of 1-shot and 5-shot accuracy on miniImageNet (backbone: WRN) when using various preprocessing steps on the extracted features. Best results are in bold.

	Inductive (NCM)		Transductive (BMS)	
Preprocessing	**1-Shot**	**5-Shot**	**1-Shot**	**5-Shot**
None	$55.30 \pm 0.21\%$	$78.34 \pm 0.15\%$	$77.62 \pm 0.26\%$	$87.96 \pm 0.13\%$
Batch Norm [60]	$66.81 \pm 0.20\%$	$83.57 \pm 0.13\%$	$73.74 \pm 0.21\%$	$88.07 \pm 0.13\%$
L2N [35]	$65.37 \pm 0.20\%$	$83.46 \pm 0.13\%$	$73.84 \pm 0.21\%$	$88.15 \pm 0.13\%$
CL2N [35]	$63.88 \pm 0.20\%$	$80.85 \pm 0.14\%$	$73.12 \pm 0.28\%$	$86.47 \pm 0.15\%$
EM_bE	$68.05 \pm 0.20\%$	$83.76 \pm 0.13\%$	$80.28 \pm 0.26\%$	$88.36 \pm 0.13\%$
PEM_bE	$\mathbf{68.43 \pm 0.20}\%$	$\mathbf{84.67 \pm 0.13}\%$	$82.01 \pm 0.26\%$	$89.50 \pm 0.13\%$
EM_nE	\	\	$80.14 \pm 0.27\%$	$88.39 \pm 0.13\%$
PEM_nE	\	\	$\mathbf{82.07 \pm 0.25}\%$	$\mathbf{89.51 \pm 0.13}\%$

Effect of power transform: we firstly conduct a Gaussian hypothesis test on each of the 640 coordinates of raw extracted features (backbone: WRN) for each of the 20 novel classes (dataset: miniImageNet). Following D'Agostino and Pearson's methodology [61,62] and $p = 1e-3$, only one of the $640 \times 20 = 12800$ tests return positive, suggesting a very low pass rate for raw features. However, after applying the power transform, we record a pass rate that surpasses 50%, suggesting a considerably increased number of positive results for Gaussian tests. This experiment shows the effect of power transform being able to adjust feature distributions into more Gaussian-like ones.

To better show the effect of this proposed technique on feature distributions, we depict in Figure 2 the distributions of an arbitrarily selected feature for three randomly selected novel classes of miniImageNet when using WRN, before and after applying the power transform. In addition, we also added to the figure the feature distributions after applying batch normalization for comparison purposes. We observe quite clearly that (1) raw features exhibit a positive distribution mostly concentrated around 0, a similar behavior is also observed for batch norm, and (2) power transform is able to reshape the feature distributions to close-to-Gaussian distributions. We observe similar behaviors with other datasets as well. Moreover, in order to visualize the impact of this technique with respect to the position of feature points, in Figure 3, we plot the feature vectors of three randomly selected classes from $\mathbf{D}_{novel}$. Note that all feature vectors in this experiment are reduced to 3-dimensional ones corresponding to their largest eigenvalues. From Figure 3, we can observe that the power transform, often followed by an L2-normalization, can help shape the class distributions to become more gathered and Gaussian-like [57].

Influence of the number of unlabeled samples: in order to better understand the gain in accuracy due to having access to more unlabeled samples, we depict in Figure 4 the evolution of accuracy as a function of q, when the number of classes $n = 5$ is fixed. Interestingly, the accuracy quickly reaches a close-to-asymptotical plateau, emphasizing the ability of the method to quickly exploit available information in the task.

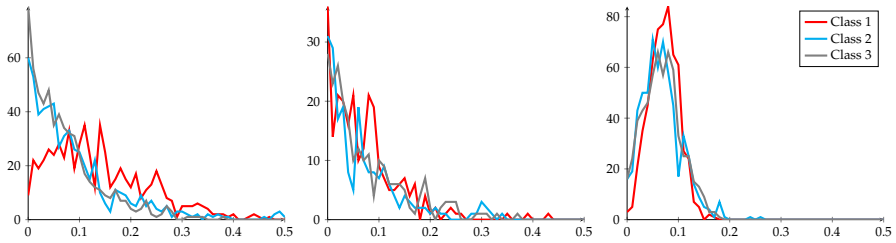


Figure 2. Distributions of an arbitrarily chosen feature for 3 novel classes with different preprocessing techniques: raw (**left**), batch norm (**middle**) and power transform (**right**).

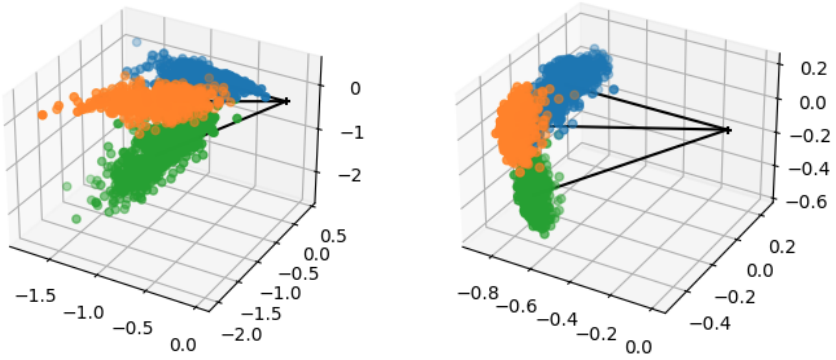


Figure 3. Plot of feature vectors (extracted from WRN) from 3 randomly selected classes (each with its own color). (**left**) Naive features. (**right**) Preprocessed features using power transform.

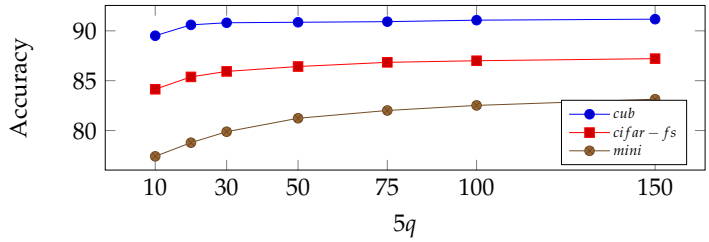


Figure 4. Accuracy of 5-way, 1-shot classification setting on miniImageNet, CUB and CIFAR-FS as a function of q.

Influence of hyperparameters: in order to test how much impact the hyperparameters could have on our proposed method in terms of prediction accuracy, here we select two important hyperparameters that are used in BMS and observe their impact. Namely, the number of training epochs e in logistic regression and the regulation parameter λ used for computing the prediction matrix $\mathbf{P}$. In Figure 5, we show the accuracy of our proposed method as a function of e (top) and λ (bottom). Results are reported for BMS* in 1-shot settings, and for BMS in 5-shot settings. From the figure, we can see a slight uptick of accuracy as e or λ increase, followed by a downhill when they become larger, implying an overfitting of the classifier. We chose our optimal parameters from these experiments. We note that, interestingly, the performance of the method appears quite robust to a non-optimal choice of these parameters.

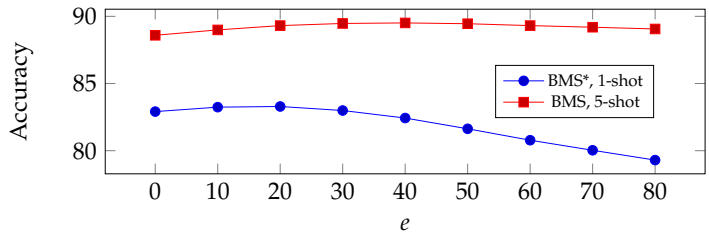


Figure 5. *Cont.*

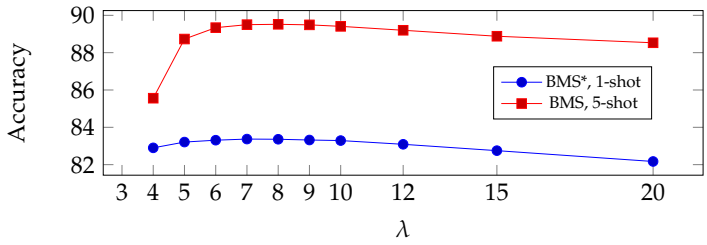


Figure 5. Accuracy of the proposed method on miniImageNet (backbone: WRN) as a function of training epoch e (**top**) and regulation parameter λ (**bottom**).

Convergence analysis: in this section, we discuss the convergence of the proposed method in Algorithm 2, namely the convergence of $\mathbf{P}$ as a function of the number of iteration step noted n_{step}. We conduct this experiment in a 5-way, 1-shot setting on miniImageNet (backbone: WRN). In Figure 6 (left), we depict $\|\Delta\mathbf{P}\|_2$ as a function of n_{step}, with $\|\Delta\mathbf{P}\|_2$ being defined as $\|\mathbf{P}(t+1) - \mathbf{P}(t)\|^2, 1 \leq t \leq n_{step}$, namely the Euclidean difference between the current $\mathbf{P}$ and the one computed in the previous step. Furthermore, we remind the reader that the goal of the proposed algorithm is to minimize the energy computed in Equation (5). Therefore, in Figure 6 (right), we depict the energy (value of Equation (5)) as a function of n_{step}. We can see that both $\|\Delta\mathbf{P}\|_2$ and energy tend to stabilize with more iteration steps.

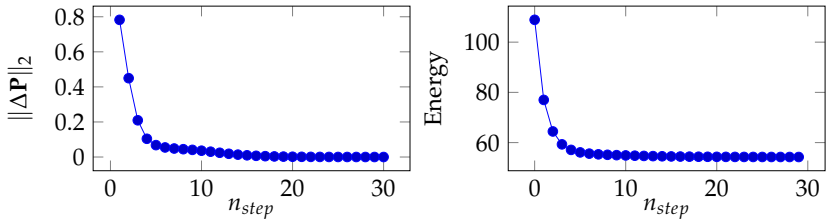


Figure 6. Convergence of BMS (1-shot on miniImagenet). (**left**) $\|\Delta\mathbf{P}\|_2$ as a function of n_{step}. (**right**) Energy (1-shot on miniImagenet) as a function of n_{step}.

Proposed method on backbones pre-trained with external data: in this experiment, we compare our proposed method BMS* with the work in [63] that pre-trains the backbone with the help of external illumination data for augmentation, followed by PT + MAP in [1] for class center estimation. Here, we use the same backbones as [63] and replace PT + MAP with our proposed BMS* under the same conditions. Results are presented in Table 7. Note that we also show the re-implemented results of [63], and our method reaches superior performance on all tested benchmarks using external data in [63].

Table 7. The proposed method on backbones pre-trained with external data. Note that *-re* denotes the re-implementation of an existing method. Best results are in bold.

Benchmark	Method	1-Shot	5-Shot
miniImageNet	Illu-Aug [63]	$82.99 \pm 0.23\%$	$89.14 \pm 0.12\%$
	Illu-Aug-*re*	$83.53 \pm 0.25\%$	$89.38 \pm 0.12\%$
	PEM$_n$E-BMS* (ours)	$\mathbf{83.85 \pm 0.25}\%$	$\mathbf{90.07 \pm 0.12}\%$
CUB	Illu-Aug [63]	$94.73 \pm 0.14\%$	$96.28 \pm 0.08\%$
	Illu-Aug-*re*	$94.63 \pm 0.15\%$	$96.06 \pm 0.08\%$
	PEM$_n$E-BMS* (ours)	$\mathbf{94.78 \pm 0.15}\%$	$\mathbf{96.43 \pm 0.07}\%$
CIFAR-FS	Illu-Aug [63]	$87.73 \pm 0.22\%$	$91.09 \pm 0.15\%$
	Illu-Aug-*re*	$87.76 \pm 0.23\%$	$91.04 \pm 0.15\%$
	PEM$_n$E-BMS* (ours)	$\mathbf{87.83 \pm 0.23}\%$	$\mathbf{91.49 \pm 0.15}\%$

Proposed method on Few-Shot Open-Set Recognition: Few-Shot Open-Set Recognition (FSOR) as a new trending topic deals with the fact that there are open data mixed in query set **Q** that do not belong to any of the supposed classes used for label predictions. Therefore, this often requires a robust classifier that is able to correctly classify the non-open data as well as rejecting the open ones. In Table 8, we apply our proposed PEME for feature preprocessing, followed by an NCM classifier and compare the results with other state-of-the-art alternatives. We observe that our proposed method is able to surpass the others in terms of accuracy and AUROC.

Table 8. Accuracy and AUROC of the proposed method for Few-Shot Open-Set Recognition. Best results are in bold.

	miniImageNet				**tieredImageNet**			
	1-Shot		**5-Shot**		**1-Shot**		**5-Shot**	
Method	**Acc**	**AUROC**	**Acc**	**AUROC**	**Acc**	**AUROC**	**Acc**	**AUROC**
ProtoNet [55]	64.01%	51.81%	80.09%	60.39%	68.26%	60.73%	83.40%	64.96%
FEAT [28]	67.02%	57.01%	82.02%	63.18%	70.52%	63.54%	84.74%	70.74%
NN [64]	63.82%	56.96%	80.12%	63.43%	67.73%	62.70%	83.43%	69.77%
OpenMax [65]	63.69%	62.64%	80.56%	62.27%	68.28%	60.13%	83.48%	65.51%
PEELER [66]	65.86%	60.57%	80.61%	67.35%	69.51%	65.20%	84.10%	73.27%
SnaTCHer [67]	67.60%	70.17%	82.36%	77.42%	70.85%	74.95%	85.23%	82.03%
PEM_bE-NCM (ours)	**68.43**%	**72.10**%	**84.67**%	**80.04**%	**71.87**%	**75.44**%	**87.09**%	**83.85**%

4.3. Proposed Method on Merged Features

In this section, we investigate the effect of our proposed method on merged features. Namely, we perform a direct concatenation of raw feature vectors extracted from multiple backbones at the beginning, followed by BMS. In Table 9, we chose the feature vectors from three backbones (WRN, ResNet18 and ResNet12) and evaluated the performance with different combinations. We observe that (1) a direct concatenation, depending on the backbones, can bring about 1% gain in both 1-shot and 5-shot settings compared with the results in Table 5 with feature vectors extracted from one single feature extractor. (2) BMS* reached new state-of-the-art results on few-shot learning benchmarks with feature vectors concatenated from WRN, ResNet18 and ResNet12, given that no external data are used.

Table 9. The 1-shot and 5-shot accuracy on miniImageNet, CUB and CIFAR-FS on our proposed PEM_nE-BMS with multi-backbones (backbone training procedure follows [2], '+' denotes a concatenation of backbone features).

	miniImageNet		**CUB**		**CIFAR-FS**	
Backbone	**1-Shot**	**5-Shot**	**1-Shot**	**5-Shot**	**1-Shot**	**5-Shot**
RN18 + RN12	80.32%	89.07%	92.31%	95.62%	85.44%	90.58%
WRN + RN12	82.63%	90.43%	92.69%	95.96%	87.11%	91.50%
WRN + RN18	83.05%	90.57%	92.66%	95.79%	87.53%	91.70%
WRN + RN18 + RN12	82.90%	90.64%	93.32%	96.31%	87.62%	91.84%
WRN + RN18 + RN12 *	84.37%	90.69%	**94.26**%	**96.32**%	**88.44**%	**91.86**%
6×WRN *	**85.54**%	**91.53**%	\	\	\	\

: BMS.

To further study the impact of the number of backbones on prediction accuracy, in Figure 7 we depict the performance of our proposed method as a function of the number of backbones. Note that, here, we operate on feature vectors of 6 WRN backbones (dataset: miniImageNet) concatenated one after another, which makes a total of 6 slots corresponding to a $640 \times 6 = 3840$ feature size. Each of them is trained the same way as in [2], and we randomly select the multiples of 640 coordinates within the slots to denote the number

of concatenated backbones used. The performance result is the average of 100 random selections, and we test with both BMS and BMS* for 1-shot, and BMS* for 5-shot. From Figure 7, we observe that, as the number of backbones increases, there is a relatively steady growth in terms of accuracy in multiple settings of our proposed method, indicating the interest of BMS in merged features.

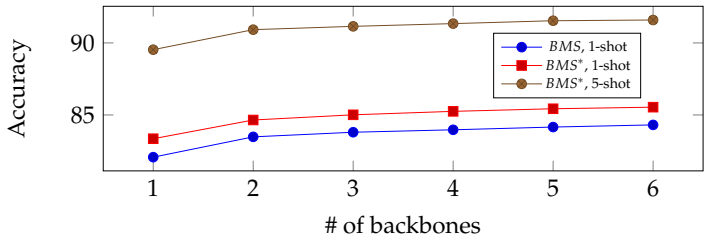


Figure 7. Accuracy of the proposed method in different settings as a function of the number of backbones (dataset: miniImageNet).

5. Conclusions

In this paper, we introduced a new pipeline to solve the few-shot classification problem. It comes with the two following assets: first, it is able to reach state-of-the-art accuracy on standardized benchmarks of the field, and second, it does not require any explicit priors about data distribution between classes, as opposed to many previous works in the domain. Using extensive experiments, we demonstrated that the proposed methodology can be used in a variety of settings, including cross-domain, multiple backbones, open-set recognition . . .. Using ablation tests, we showed the importance of the introduced steps in the methodology. The proposed methodology comes with only a few extra hyperparameters, on which our experiments suggest that a fine tuning is not necessarily required. Thus we believe that the proposed method is applicable to many practical engineering problems. In future work, we would like to better understand the fundamental reasons why the proposed preprocessing is able to boost performance. We would also like to find automatic ways to tune the hyperparameters.

Author Contributions: Conceptualization, Y.H., S.P., and V.G.; methodology, Y.H. and S.P.; software, S.P. and V.G.; validation, Y.H., S.P., and V.G.; formal analysis, Y.H., S.P., and V.G.; investigation, Y.H.; resources, S.P. and V.G.; data curation, Y.H.; writing—original draft preparation, Y.H.; writing—review and editing, Y.H., S.P., and V.G.; visualization, Y.H.; supervision, S.P. and V.G.; project administration, S.P. and V.G; funding acquisition, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Orange, France.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting reported results can be found at https://github.com/yhu01/BMS (accessed on 1 February 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, Y.; Gripon, V.; Pateux, S. Leveraging the feature distribution in transfer-based few-shot learning. In *Artificial Neural Networks and Machine Learning—ICANN 2021, Proceedings of the 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, 14–17 September 2021*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 487–499.
2. Mangla, P.; Kumari, N.; Sinha, A.; Singh, M.; Krishnamurthy, B.; Balasubramanian, V.N. Charting the right manifold: Manifold mixup for few-shot learning. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 2218–2227.
3. Wang, X.; Huang, T.E.; Gonzalez, J.; Darrell, T.; Yu, F. Frustratingly Simple Few-Shot Object Detection. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Virtual Event, 13–18 July 2020; Volume 119, pp. 9919–9928.

4. Wang, Y.; Bryan, N.J.; Cartwright, M.; Bello, J.P.; Salamon, J. Few-Shot Continual Learning for Audio Classification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, 6–11 June 2021; pp. 321–325.
5. Wolters, P.; Careaga, C.; Hutchinson, B.; Phillips, L. A Study of Few-Shot Audio Classification. *arXiv* **2020**, arXiv:2012.01573.
6. Zhang, S.; Qin, Y.; Sun, K.; Lin, Y. Few-Shot Audio Classification with Attentional Graph Neural Networks. In *Interspeech 2019, Proceedings of the 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019*; Kubin, G., Kacic, Z., Eds.; ISCA: Yuma, AZ, USA, 2019; pp. 3649–3653.
7. Bansal, T.; Jha, R.; McCallum, A. Learning to Few-Shot Learn Across Diverse Natural Language Classification Tasks. In Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain, 8–13 December 2020; Scott, D., Bel, N., Zong, C., Eds.; International Committee on Computational Linguistics: Barcelona, Spain, 2020; pp. 5108–5123.
8. Bansal, T.; Jha, R.; Munkhdalai, T.; McCallum, A. Self-Supervised Meta-Learning for Few-Shot Natural Language Classification Tasks. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020; Webber, B., Cohn, T., He, Y., Liu, Y., Eds.; Association for Computational Linguistics: Seattle, USA 2020; pp. 522–534.
9. Bao, Y.; Wu, M.; Chang, S.; Barzilay, R. Few-shot Text Classification with Distributional Signatures. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
10. Sun, L.; Li, C.; Ding, X.; Huang, Y.; Chen, Z.; Wang, G.; Yu, Y.; Paisley, J.W. Few-shot medical image segmentation using a global correlation network with discriminative embedding. *Comput. Biol. Med.* **2022**, *140*, 105067. [CrossRef] [PubMed]
11. Tang, H.; Liu, X.; Sun, S.; Yan, X.; Xie, X. Recurrent mask refinement for few-shot medical image segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 3918–3928.
12. Ouyang, C.; Biffi, C.; Chen, C.; Kart, T.; Qiu, H.; Rueckert, D. Self-supervision with superpixels: Training few-shot medical image segmentation without annotation. In *Computer Vision—ECCV 2020, Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23– 28 August 2020* ; Springer: Berlin/Heidelberg, Germany, 2020; pp. 762–780.
13. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1126–1135.
14. Ravi, S.; Larochelle, H. Optimization as a Model for Few-Shot Learning. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
15. Thrun, S.; Pratt, L. *Learning to Learn*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
16. Torrey, L.; Shavlik, J. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*; IGI Global: Hershey, PA, USA, 2010; pp. 242–264.
17. Das, D.; Lee, C.S.G. A Two-Stage Approach to Few-Shot Learning for Image Recognition. *IEEE Trans. Image Process.* **2020**, *29*, 3336–3350. [CrossRef] [PubMed]
18. Chen, W.; Liu, Y.; Kira, Z.; Wang, Y.F.; Huang, J. A Closer Look at Few-shot Classification. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
19. Hu, Y.; Gripon, V.; Pateux, S. Graph-based interpolation of feature vectors for accurate few-shot classification. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 8164–8171.
20. Lichtenstein, M.; Sattigeri, P.; Feris, R.; Giryes, R.; Karlinsky, L. Tafssl: Task-adaptive feature sub-space learning for few-shot classification. In *Computer Vision—ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 522–539.
21. Li, Z.; Zhou, F.; Chen, F.; Li, H. Meta-SGD: Learning to Learn Quickly for Few Shot Learning. *arXiv* **2017**, arXiv:1707.09835.
22. Antoniou, A.; Edwards, H.; Storkey, A.J. How to train your MAML. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
23. Bertinetto, L.; Henriques, J.F.; Torr, P.H.S.; Vedaldi, A. Meta-learning with differentiable closed-form solvers. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
24. Zhang, H.; Zhang, J.; Koniusz, P. Few-shot Learning via Saliency-guided Hallucination of Samples. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 2770–2779.
25. Chen, Z.; Fu, Y.; Wang, Y.X.; Ma, L.; Liu, W.; Hebert, M. Image deformation meta-networks for one-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 8680–8689.
26. Mensink, T.; Verbeek, J.; Perronnin, F.; Csurka, G. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *Computer Vision—ECCV 2012, Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012* ; Springer: Berlin/Heidelberg, Germany, 2012; pp. 488–501.
27. Chapelle, O.; Scholkopf, B.; Zien, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Trans. Neural Netw.* **2009**, *20*, 542–542. [CrossRef]
28. Ye, H.J.; Hu, H.; Zhan, D.C.; Sha, F. Few-shot learning via embedding adaptation with set-to-set functions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 8808–8817.
29. Zhang, C.; Cai, Y.; Lin, G.; Shen, C. DeepEMD: Few-Shot Image Classification With Differentiable Earth Mover's Distance and Structured Classifiers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 12203–12213.
30. Kang, D.; Kwon, H.; Min, J.; Cho, M. Relational Embedding for Few-Shot Classification. *arXiv* **2021**, arXiv:2108.09666.

31. Rizve, M.N.; Khan, S.H.; Khan, F.S.; Shah, M. Exploring Complementary Strengths of Invariant and Equivariant Representations for Few-Shot Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, 19–25 June 2021; pp. 10836–10846.
32. Rajasegaran, J.; Khan, S.H.; Hayat, M.; Khan, F.S.; Shah, M. Self-supervised Knowledge Distillation for Few-shot Learning. *arXiv* **2020**, arXiv:2006.09785.
33. Zhang, Z.; Sabuncu, M.R. Self-Distillation as Instance-Specific Label Smoothing. In *Advances in Neural Information Processing Systems 33, Proceedings of the Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, Virtual, 6–12 December 2020*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; MIT Press: Cambridge, MA, USA, 2020.
34. Tian, Y.; Wang, Y.; Krishnan, D.; Tenenbaum, J.B.; Isola, P. Rethinking Few-Shot Image Classification: A Good Embedding is All You Need? In *Computer Vision—ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12359, pp. 266–282. [CrossRef]
35. Wang, Y.; Chao, W.; Weinberger, K.Q.; van der Maaten, L. SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning. *arXiv* **2019**, arXiv:1911.04623.
36. Gripon, V.; Hacene, G.B.; Löwe, M.; Vermet, F. Improving Accuracy of Nonparametric Transfer Learning via Vector Segmentation. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2966–2970.
37. Yang, S.; Liu, L.; Xu, M. Free Lunch for Few-shot Learning: Distribution Calibration. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual, 3–7 May 2021.
38. Liu, Y.; Lee, J.; Park, M.; Kim, S.; Yang, E.; Hwang, S.J.; Yang, Y. Learning to Propagate Labels: Transductive Propagation Network for Few-Shot Learning. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
39. Agarap, A.F. Deep Learning using Rectified Linear Units (ReLU). *arXiv* **2018**, arXiv:1803.08375.
40. Tukey, J.W. *Exploratory Data Analysis*; Springer: Reading, MA, USA 1977; Volume 2.
41. Cinbis, R.G.; Verbeek, J.; Schmid, C. Approximate fisher kernels of non-iid image models for image categorization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 1084–1098. [CrossRef] [PubMed]
42. Villani, C. *Optimal Transport: Old and New*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008; Volume 338.
43. Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2013; pp. 2292–2300.
44. Solomon, J.; De Goes, F.; Peyré, G.; Cuturi, M.; Butscher, A.; Nguyen, A.; Du, T.; Guibas, L. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph. (TOG)* **2015**, *34*, 1–11. [CrossRef]
45. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–22.
46. Zagoruyko, S.; Komodakis, N. Wide Residual Networks. In Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, 19–22 September 2016; Wilson, R.C., Hancock, E.R., Smith, W.A.P., Eds.; BMVA Press: York, UK, 2016.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
48. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
49. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2016; pp. 3630–3638.
50. Liu, J.; Song, L.; Qin, Y. Prototype rectification for few-shot learning. In *Computer Vision—ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 741–756.
51. Ziko, I.; Dolz, J.; Granger, E.; Ayed, I.B. Laplacian regularized few-shot learning. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 12–18 July 2020; pp. 11660–11670.
52. Boudiaf, M.; Ziko, I.M.; Rony, J.; Dolz, J.; Piantanida, P.; Ayed, I.B. Transductive Information Maximization For Few-Shot Learning. *arXiv* **2020**, arXiv:2008.11297.
53. Kye, S.M.; Lee, H.; Kim, H.; Hwang, S.J. Transductive Few-shot Learning with Meta-Learned Confidence. *arXiv* **2020**, arXiv:2002.12017.
54. Rodríguez, P.; Laradji, I.; Drouin, A.; Lacoste, A. Embedding propagation: Smoother manifold for few-shot classification. In *Computer Vision—ECCV 2020, Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020* ; Springer: Berlin/Heidelberg, Germany, 2020; pp. 121–138.
55. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 4077–4087.
56. Rusu, A.A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; Hadsell, R. Meta-Learning with Latent Embedding Optimization. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
57. Chobola, T.; Vasata, D.; Kordík, P. Transfer learning based few-shot classification using optimal transport mapping from preprocessed latent space of backbone neural network. *arXiv* **2021**, arXiv:2102.05176.

58. Simon, C.; Koniusz, P.; Nock, R.; Harandi, M. Adaptive Subspaces for Few-Shot Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4136–4145.
59. Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6438–6447.
60. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
61. DIAgostino, R. An omnibus test of normality for moderate and large sample sizes. *Biometrika* **1971**, *58*, 341–348. [CrossRef]
62. D'AGOSTINO, R.; Pearson, E.S. Tests for departure from normality. Empirical results for the distributions of b 2 and $\sqrt{b}$. *Biometrika* **1973**, *60*, 613–622. [CrossRef]
63. Zhang, H.; Cao, Z.; Yan, Z.; Zhang, C. Sill-Net: Feature Augmentation with Separated Illumination Representation. *arXiv* **2021**, arXiv:2102.03539.
64. Júnior, P.R.M.; De Souza, R.M.; Werneck, R.d.O.; Stein, B.V.; Pazinato, D.V.; de Almeida, W.R.; Penatti, O.A.; Torres, R.d.S.; Rocha, A. Nearest neighbors distance ratio open-set classifier. *Mach. Learn.* **2017**, *106*, 359–386. [CrossRef]
65. Bendale, A.; Boult, T.E. Towards open set deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1563–1572.
66. Liu, B.; Kang, H.; Li, H.; Hua, G.; Vasconcelos, N. Few-shot open-set recognition using meta-learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 8798–8807.
67. Jeong, M.; Choi, S.; Kim, C. Few-shot Open-set Recognition by Transformation Consistency. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12566–12575.

4.3 Discussions

Thanks to the relatively large improvement that "PT+MAP" has brought in transductive few-shot classification, the performance of our proposed method stayed state-of-the-art for a long time on Papers With Code [1] [2]. Our paper **Leveraging the Feature Distribution in Transfer-Based Few-Shot Learning** has been frequently cited and the proposed method has been reused, studied and extended by other works such as [LSA21; CVK21; Ben+22c; ZK22; Hu+22; Wal+22], along with many other methods that are also based on Optimal Transport. Moreover, our other paper **Squeezing Backbone Feature Distributions to the Max for Efficient Few-Shot Learning** further improves the prediction accuracy based on a logistic regression algorithm that makes use of the pseudo labels on query set samples. In this section we will address more details of our proposed method.

4.3.1 Importance of feature preprocessing

From the contributions presented above, we observe that feature preprocessing is crucial for our proposed method to obtain maximum increase in accuracy. The goal is to reshape the feature distributions to be close to Gaussian so that the designed classifier yields its optimal effect. From Figure 2 in the paper, the initial feature distribution coming out of ReLU does not look like Gaussian at all, but rather a Log-normal distribution. The proposed Power Transform (PT) is able to adjust the distributions into Gaussian-like, with more than half of the the reshaped features passing the Gaussian hypothesis test. Furthurmore, work in [CVK21] managed to obtain slight improvement by adding a hyperparameter on the feature normalization in order for a finer adjustment.

4.3.2 Logistic regression classifier

Another interesting observation in our proposed method is that the logistic regression applied in [HPG22a] brings larger gain in 5-shot setting than in 1 shot compared with PT+MAP, raising the question of how can it influence the performance as the number of labeled samples become larger, i.e. in the case of 10-shot or 20-shot scenarios [Vei+21]. Note that here we apply logistic regression on all samples, including unlabeled ones with their soft class assignments. We believe to be the first to integrate such a process into an EM framework for better cluster estimations. Moreover, according to the loss function provided in Eq. 12, we see that it also corresponds to a self-distillation process that is presented in [ZS20], with a scaling parameter that attempts to adjust the logits to be close to the soft class assignments from OT. In other words, similar to the distillation process where the student mimics the teacher in terms of logits, here in our method the logits are computed using OT and served as the teacher for the logistic regression model to learn as a student.

[1] https://paperswithcode.com/sota/few-shot-image-classification-on-mini-2
[2] https://paperswithcode.com/sota/few-shot-image-classification-on-mini-3

4.3.3 Limitations on OT

Although methods based on OT obtain relatively large increase in accuracy, they all require the prior knowledge about the distribution of query set over targeted classes [Vei+21], namely the number of unlabeled samples per class. Therefore, in a balanced setting where the query set distribution is uniform, OT based methods have the major advantage on the class assignments. However, in an unbalanced setting where we are given a fixed total number of samples to predict labels but do not know the number of samples for a class, these methods do not perform well [Vei+21].

4.3.4 Perspectives

In our first attempt to address the problem of imbalance, in this work we proposed a modified version of OT to try to reduce the effect of priors by only normalizing columns that have less sums than the predicted minimum number [Lic+20]. However, without knowing the exact proportion of unlabeled samples over classes, the algorithm tends to equalise those unlabeled samples and thus still results in a significant decrease in accuracy in the unbalanced setting. Indeed, afterwards, we believe that the method proposed in [HSS18] is not well suited to deal with unbalanced cases. Therefore, in summary, given that in most real world scenarios we do not know how a test set is allocated, OT based methods tend to be less practical.

In the next paper we propose a variation Bayesian method that is more desirable for the real world situations and obtain state-of-the-art performance in the unbalanced setting.

Chapter 5

Adaptive Dimension Reduction and Variational Inference for Transductive Few-Shot Classification

In previous chapter we present our contribution mainly tackling the classifier design step of the pipeline. This chapter continues the improvement on the classifier design, and we present our work that addresses the limitations of previous work. In this chapter we present the context, the paper with our proposed method, discussions with additional experiments, limitations and perspectives.

5.1 Context

As previously stated, methods based on OT require the prior knowledge about the query data, without which the OT algorithms would tend towards a balanced distribution and thus do not perform well in the unbalanced setting [Vei+21]. Therefore, we need algorithms that better estimate the targeted classes in a data-thrifty and unbalanced situation where no prior is required.

Current methods for transductive FSC have been focusing more and more on cluster estimations [YLX21; Bat+20; Bat+22], i.e. class means and class covariances under Gaussian Mixture Models. Although algorithms operated under the EM framework can bring gains in accuracy, this remains to be a challenging task given that there are too few labeled data in a test set to estimate in a way that can accurately describe a cluster, especially the randomness and uncertainty that have to be dealt with due to data thrifty.

There exists other techniques that operate under EM framework, such as Variational Bayesian (VB) inference [HG08; FR12] that realises the clustering as well. Compared with EM algorithm that estimates cluster parameters directly, VB inference regards these parameters as hidden variables and thus introduces more other parameters for estimations. The goal is to approximate the posterior of the variables by a variational distribution. In the case of transductive few-shot classification, there are two main difficulties to be considered when applying VB

inference: 1) given the semi-supervised feature of the transductive few-shot setting, adaptations need to be made on VB inference which is unsupervised in nature; 2) limited data samples in the test set with high feature dimensions (typically 512 or 640) may contain too much noise for estimations and cause a VB model to collapse or have coarse results.

With all of the problems presented above, and considering the fact that VB models take into account the mixture parameters of clusters that follow the Dirichlet distribution law, which corresponds to the unbalanced setting proposed in [Vei+21], in our paper **Adaptive Dimension Reduction and Variational Inference for Transductive Few-Shot Classification** [HPG22b] we propose a new clustering method that is based on Variational Bayesian (VB) inference, which can be seen as an extension of EM algorithm. Our proposed VB model 1) makes use of the labeled data in the support set for prototype estimations; 2) takes into account the uncertainty in estimation by viewing class prototypes as random variables, which would inject more flexibility into the model; and 3) utilizes Probabilistic Linear Discriminant Analysis (PLDA) to reduce feature dimension so that the proposed model is less complex and more stable, while maximizing the inter/intra-class distance ratio.

5.2 Paper on using Variational inference and Adaptive Dimension Reduction to reach SOTA performance

Adaptive Dimension Reduction and Variational Inference for Transductive Few-Shot Classification

Yuqing Hu
Orange Labs, Cesson-Sévigné, France
IMT-Atlantique, Brest, France
yuqing.hu@imt-atlantique.fr

Stéphane Pateux
Orange Labs, Cesson-Sévigné, France
stephane.pateux@orange.com

Vincent Gripon
IMT-Atlantique, Brest, France
vincent.gripon@imt-atlantique.fr

Abstract

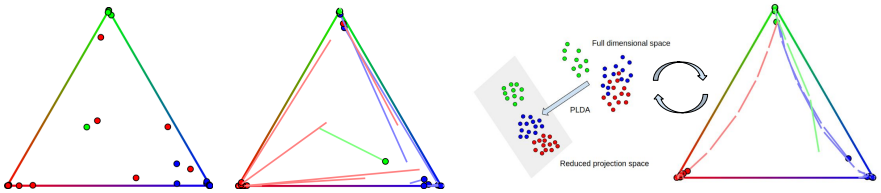
Transductive Few-Shot learning has gained increased attention nowadays considering the cost of data annotations along with the increased accuracy provided by unlabelled samples in the domain of few shot. Especially in Few-Shot Classification (FSC), recent works explore the feature distributions aiming at maximizing likelihoods or posteriors with respect to the unknown parameters. Following this vein, and considering the parallel between FSC and clustering, we seek for better taking into account the uncertainty in estimation due to lack of data, as well as better statistical properties of the clusters associated with each class. Therefore in this paper we propose a new clustering method based on Variational Bayesian inference, further improved by Adaptive Dimension Reduction based on Probabilistic Linear Discriminant Analysis. Our proposed method significantly improves accuracy in the realistic unbalanced transductive setting on various Few-Shot benchmarks when applied to features used in previous studies, with a gain of up to 6% in accuracy. In addition, when applied to balanced setting, we obtain very competitive results without making use of the class-balance artefact which is disputable for practical use cases. We also provide the performance of our method on a high performing pretrained backbone, with the reported results further surpassing the current state-of-the-art accuracy, suggesting the genericity of the proposed method.

1 Introduction

Few-shot learning, and in particular Few-Shot Classification, has become a subject of paramount importance in the last years with a large number of methodologies and discussions. Where large datasets continuously benefit from improved machine learning architectures, the ability to transfer this performance to the low-data regime is still a challenge due to the high uncertainty posed using few labels. In more details, there are two main types of FSC tasks. In *inductive* FSC [1, 36, 46, 33], the situation comes to its extremes with only a few data samples available for each class, leading sometimes to completely intractable settings, such as when facing a black dog on the one hand and a white cat on the other hand. In *transductive* FSC, additional unlabelled samples are available for prediction, leading to improved reliability and more elaborate solutions [24, 23, 2].

Inductive FSC is likely to occur when data acquisition is difficult or expensive, or when categories of interest correspond to rare events. Transductive FSC is more likely encountered when data labeling is expensive, for fast prototyping of solutions, or when the categories of interest are rare and hard to detect. Since the latter correspond to situations where it is possible to exploit, at least partially,

Preprint. Under review.



(a) Few-Shot task (b) Initialization (c) PLDA and VB inference

Figure 1: Summary of the proposed method. Here we illustrate a 3-way classification task in a standard 2-simplex using soft-classification probabilities. Trajectories show the evolution across iterations. For a given Few-Shot task which nearest-class-mean probabilities are depicted in (a), a Soft-KMEANS clustering method is performed in (b) to initialize o_{nk} (see Alg. 1). Then in (c) an iteratively refined Variational Bayesian (VB) model with Adaptive Dimension Reduction using Probabilistic Linear Discriminant Analysis (PLDA) is applied to obtain the final class predictions.

the distribution of unlabelled samples, the trend evolved to using potentially varying parts of this additional source of information. With most standardized benchmarks using very limited scope of variability in the generated Few-Shot tasks, this even came to the point the best performing methods are often relying on questionable information, such as equidistribution between the various classes among the unlabelled samples, that is unlikely realistic in applications.

This limitation of benchmarking for transductive FSC has recently been discussed in [39]. In this paper, the authors propose a new way of generating transductive FSC benchmarks where the distribution of samples among classes can drastically change from a Few-Shot generated task to the next one. Interestingly, they showed the impact of generating class imbalance on the performance on various popular methods, resulting in some cases in drops in average accuracy of more than 10%.

A simple way to reach state-of-the-art performance in transductive FSC consists in extracting features from the available samples using a pretrained backbone deep learning architecture, and then using semi-supervised clustering routines to estimate samples distribution among classes. Due to the very limited number of available samples, distribution-agnostic clustering algorithms are often preferred, such as K-MEANS or its variants [29, 25, 32] or mean-shift [9] for instance.

In this paper, we are interested in showing it is possible to combine data reduction with statistical inference through a Variational Bayesian (VB) [13] approach. Here, data reduction helps considerably reduce the number of parameters to infer, while VB provides more flexibility than the usual K-Means methods. Interestingly, the proposed approach can easily cope with standard equidistributed Few-Shot tasks or the unbalanced ones proposed in [39], defining a new state-of-the-art for five popular transductive Few-Shot vision classification benchmarks.

Our claims are the following:

- We introduce a novel semi-supervised clustering algorithm based on VB inference and Probabilistic Linear Discriminant Analysis (PLDA),
- We demonstrate the general utility of our proposed method being able to improve accuracy in a variety of deep learning models and settings,
- We show the ability of the proposed method to reach state-of-the-art transductive FSC performance on multiple vision benchmarks (balanced and unbalanced).

2 Related work

There are two main frameworks in the field of FSC: 1) only one unlabelled sample is processed at a time for class predictions, which is called inductive FSC, and 2) the entire unlabelled samples are available for further estimations, which is called transductive FSC. Inductive methods focus on training a feature extractor that generalizes well the embedding in a feature sub-space, they include meta learning methods such as [12, 26, 2, 40, 30, 37] that train a model in an episodic manner, and transfer learning methods [8, 28, 48, 5, 3, 33] that train a model with a set of mini-batches. Recent

state-of-the-art works on inductive FSC [46, 47, 43, 19] combine the above two strategies and propose a transfer based training used as model initialization, followed by an episodic training that adapts the model to better fit the Few-Shot tasks.

Transductive methods are becoming more and more popular thanks to their better performance due to the use of unlabelled data, as well as their utility in situations where data annotation is costly. Early literature of this branch operates on a class-balanced setting where unlabelled instances are evenly distributed among targeted classes. Graph-based methods [14, 7, 44, 21] make use of the affinity among features and propose to group those that belong to the same class. More recent works such as [16] propose methods based on Optimal Transport that realizes sample-class allocation with a minimum cost. While effective, these methods often require class-balanced priors to work well, which is not realistic due to the arbitrary unknown query set. In [39] the authors put forward a novel unbalanced setting that composes a query set with unlabelled instances sampled following a Dirichlet distribution, injecting more imbalance for predictions.

In this paper we propose a clustering method to solve transductive FSC, where the aim is to estimate cluster parameters giving high predictions for unlabelled samples. Under Gaussian assumptions, previous works [25, 32] have utilised algorithms such as Expectation Maximization [10] (EM), with the goal of maximizing likelihoods or posteriors with respect to the parameters for a cluster, with the hidden variables marginalized. However, this may not be the most suitable way due to the scarcity of available data in a given Few-Shot task, which increases the level of uncertainty for cluster estimations. Therefore, in this paper we propose a Variational Bayesian (VB) approach, in which we regard some unknown parameters as hidden variables in order to inject more flexibility into the model, and we try to approximate the posterior of the hidden variables by a variational distribution.

As models with too few labelled samples often give too much randomness for a cluster to be stably reckoned, they often require the use of feature dimension reduction techniques to stabilize cluster estimations. Previous literature such as [25] applies a PCA method that reduces dimension in a non-supervised manner, and [6] proposes a modified LDA during backbone training that maximizes the ratio of inter/intra-class distance. In this paper we propose to use Probabilistic Linear Discriminant Analysis [17] (PLDA) that 1) is applied on extracted features, 2) fits data more desirably into distribution assumptions, and 3) is semi-supervised in combination of a VB model. We integrate PLDA into the VB model in order to refine the reduced space through iterations.

3 Methodology

In this section, we firstly present the standard setting in transductive FSC, including the latest unbalanced setting proposed by [39] where unlabelled samples are non-uniformly distributed among classes. Then we present our proposed method combining PLDA and VB inference.

3.1 Problem formulation

Following other works in the domain, our proposed method is operated on a feature space obtained from a pre-trained backbone. Namely, we are given the extracted features of 1) a generic base class dataset $\mathcal{D}_{base} = \{\boldsymbol{x}_i^{base}\}_{i=1}^{N_{base}} \in \mathcal{C}_{base}$ that contains N_{base} labelled samples where each sample $\boldsymbol{x}_i^{base}$ is a column vector of length D, and $\mathcal{C}_{base}$ is the set of base classes to which these samples belong. These base classes have been used to train the backbone. And similarly, 2) a novel class dataset $\mathcal{D}_{novel} = \{\boldsymbol{x}_n^{novel}\}_{n=1}^{N}$ containing N samples belonging to a set of K novel classes $\mathcal{C}_{novel}$ ($\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$). On this novel dataset, only a few elements are labelled, and the aim is to predict the missing labels. Denote $\mathbf{X}$ the matrix obtained by aggregating elements in $\mathcal{D}_{novel}$ row-wise.

When benchmarking transductive FSC methods, it is common to randomly generate Few-Shot tasks by sampling $\mathcal{D}_{novel}$ from a larger dataset. These tasks are generated by sampling K distinct classes, L distinct labelled elements for each class (called support set) and Q total unlabelled elements without repetition and distinct from the labelled ones (called query set). All these unlabelled elements belong to one of the selected classes. We obtain a total of $N = KL + Q$ elements in the task, and compute the accuracy on the Q unlabelled ones. Depending on how unlabelled instances are distributed among selected classes within a task, we further distinguish a balanced setting where the query set is evenly distributed among the K classes, from an unbalanced setting where it can vary from class to class. An automatic way to generate such unbalanced Few-Shot tasks has been proposed in [39]

where the number of elements to draw from each class is determined using a Dirichlet distribution parameterized by $\alpha_o^* \mathbf{1}$, where $\mathbf{1}$ is the all-one vector. To solve a transductive FSC task, our method is composed of PLDA and VB inference, that we introduce in the next paragraphs.

3.2 Probabilistic Linear Discriminant Analysis (PLDA)

In our work, PLDA [17] is mainly used to reduce feature dimensions. For a Few-Shot task $\mathbf{X}$, let $\mathbf{\Phi}_w$ be a positive definite matrix representing the estimated shared within-class covariance of a given class, and $\mathbf{\Phi}_b$ be a positive semi-definite matrix representing the estimated between-class covariance that generates class variables. The goal of PLDA is to project data onto a subspace while maximizing the signal-to-noise ratio for class labelling. In details, we obtain a projection matrix $\mathbf{W}$ that diagonalizes both $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_b$ and yield the following equations:

$$\mathbf{W}^T \mathbf{\Phi}_w \mathbf{W} = \mathbf{I}, \quad \mathbf{W}^T \mathbf{\Phi}_b \mathbf{W} = \mathbf{\Psi} \tag{1}$$

where $\mathbf{I}$ is an identity matrix and $\mathbf{\Psi}$ is a diagonal matrix. In this paper, we assume a similar distribution between the pre-trained base classes and the transferred novel classes [45]. Therefore we propose to estimate $\mathbf{\Phi}_w$ to be the within-class scatter matrix of $\mathcal{D}_{base}$, denoted as $\mathbf{S}_w^{bass}$. In practice we implement PLDA by firstly transforming $\mathbf{X}$ using a rotation matrix $\mathbf{R} \in \mathbb{R}^{D \times D}$ and a set of scaling values $\boldsymbol{s} \in \mathbb{R}^D$ obtained from $\mathbf{S}_w^{base}$. Note that we clamp the scaling values to be no larger than an upper-bound s_{max} in order to prevent too large values, s_{max} is a hyper-parameter. Then we project the transformed data onto their estimated class centroids space, in accordance with the d largest eigenvalues of $\mathbf{\Psi}$, and obtain dimension-reduced data $\mathbf{U} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_n, ..., \boldsymbol{u}_N]^T \in \mathbb{R}^{N \times d}$ where $\boldsymbol{u}_n = \mathbf{W}^T \boldsymbol{x}_n$ and $d = K - 1$. More detailed implementation can be found in Appendix.

3.3 Variational Bayesian (VB) Inference

During VB inference, we operate on a reduced d-dimensional space obtained after applying PLDA. Considering a Gaussian mixture model for a given task $\mathbf{U} \in \mathbb{R}^{N \times d}$ in reduced space, let θ be the unknown variables of the model. In VB we attempt to find a probability distribution $q(\theta)$ that approximates the true posterior $p(\theta|\mathbf{U})$, i.e. maximizes the ELBO (see Appendix for more details). In our case, we define $\theta = \{\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}\}$ where $\boldsymbol{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$ is a set of latent variables used as class indicators, each latent variable $\boldsymbol{z}_n$ has an one-of-K representation, $\boldsymbol{\pi}$ is a K-dimensional vector representing mixing ratios between the classes, and $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1}^K$ where $\boldsymbol{\mu}_k$ is the centroid for class k. Note that 1) contrary to EM where $\boldsymbol{\pi}, \boldsymbol{\mu}$ are seen as parameters that can be estimated directly, in VB they are deemed as hidden variables following certain distribution laws. 2) This is not a full VB model due to the lack of precision matrix (i.e. the inverse of covariance matrix) as a variable in θ. Although a VB model frees up more parameters for the unknown variables, it also increases the instability in estimations so that the model becomes too sensible. Therefore, in this paper we impose an assumption that all classes in $\mathbf{U}$ share the same precision matrix and it is fixed during VB iterations. Namely we define $\mathbf{\Lambda}_k = \mathbf{\Lambda} = T_{vb}\mathbf{I}$ for $k = 1, ..., K$, where T_{vb} is a hyper-parameter in order to compensate the variation between base and estimated novel class distributions.

In order for a model to be in a variational bayesian setting, we define priors and likelihoods on the unknown variables, with several initialization parameters attached:

$$\begin{aligned} \text{priors}: \quad & p(\boldsymbol{\pi}) = Dir(\boldsymbol{\pi}|\alpha_o), \quad p(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_o, (\beta_o \mathbf{\Lambda})^{-1}), \\ \text{likelihoods}: \quad & p(\boldsymbol{Z}|\boldsymbol{\pi}) = \prod_{n=1}^{N} Categorical(\boldsymbol{z}_n|\boldsymbol{\pi}), \quad p(\boldsymbol{U}|\boldsymbol{Z}, \boldsymbol{\mu}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{u}_n|\boldsymbol{\mu}_k, \mathbf{\Lambda}^{-1})^{z_{nk}} \end{aligned} \tag{2}$$

where $\boldsymbol{\pi}$ follows a K-dimensional symmetric Dirichlet distribution, with α_o being the prior of component weight for each class, which we set to 2.0 in accordance with [39], i.e. the same value as the Dirichlet distribution parameter α_o^* that is used to generate Few-Shot tasks. The vector $\boldsymbol{m}_o$ is the prior about the class centroid variables, we let it to be $\mathbf{0}$. And β_o stands for the prior about the moving range of class centroid variables: the larger it is, the closer the centroids are to $\boldsymbol{m}_o$. We empirically found that $\beta_o = 10.0$ gives consistent good results across datasets and FSC problems.

As previously stated, we approximate a variable distribution to the true posterior. To further simplify, we follow the Mean-Field assumption [31, 18] and assume that the unknown variables are independent from one another. Therefore we let $q(\theta) = q(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}) = q(\boldsymbol{Z})q(\boldsymbol{\pi})q(\boldsymbol{\mu}) \approx p(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}/\mathbf{U})$ and solve for each term. The explicit formulation for these marginals is provided in Eq. 3, 4 (see Appendix for more details). The estimation of the various parameters is then classically performed through an iterative EM framework as presented further.

Denote $\boldsymbol{o}_n = [o_{n1}, ..., o_{nk}, ..., o_{nK}]$ as the soft class assignment for $\boldsymbol{u}_n$ ($o_{nk} \geq 0,\ \sum_{k=1}^{K} o_{nk} = 1$), and o_{nk} represents the portion of nth sample allocated to kth class.

M step: In this step we estimate $q(\boldsymbol{\pi})$ and $q(\boldsymbol{\mu})$ in use of the class assignments o_{nk}:

$$\begin{aligned} p(\boldsymbol{\pi}) = Dir(\boldsymbol{\pi}|\alpha_o) \implies q^*(\boldsymbol{\pi}) = Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}) \quad \text{with} \quad \alpha_k = \alpha_o + N_k, \\ p(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_o, (\beta_o\boldsymbol{\Lambda})^{-1}) \implies q^*(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\beta_k\boldsymbol{\Lambda})^{-1}) \\ \text{with} \quad \beta_k = \beta_o + N_k,\ \boldsymbol{m}_k = \frac{1}{\beta_k}(\beta_o\boldsymbol{m}_o + \sum_{n=1}^{N} o_{nk}\boldsymbol{u}_n), \end{aligned} \tag{3}$$

where $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_k, ..., \alpha_K]$ are the estimated component weights for classes, and $N_k = \sum_{n=1}^{N} o_{nk}$ is the sum of the soft assignments for all samples in class k. We also estimate the moving range parameter β_k and the centroid $\boldsymbol{m}_k$ for each class centroid variable. We observe that the posteriors take the same forms as the priors. Demonstration of these results is presented in Appendix.

E step: In this step we estimate $q(\boldsymbol{Z})$ by updating o_{nk}, using the current values of all other parameters computed in the M-step, i.e. α_k, β_k and $\boldsymbol{m}_k$.

$$p(\boldsymbol{Z}|\boldsymbol{\pi}) = \prod_{n=1}^{N} Categorical(\boldsymbol{z}_n|\boldsymbol{\pi}) \implies q^*(\boldsymbol{Z}) = \prod_{n=1}^{N} Categorical(\boldsymbol{z}_n|\boldsymbol{o}_n) \tag{4}$$

where each element o_{nk} can be computed as $o_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^{K} \rho_{nj}}$ in which:

$$\log \rho_{nk} = \psi(\alpha_k) - \psi(\sum_{j=1}^{K} \alpha_j) + \frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{d}{2}\log 2\pi - \frac{1}{2}[d\beta_k^{-1} + (\boldsymbol{u}_n - \boldsymbol{m}_k)^T\boldsymbol{\Lambda}(\boldsymbol{u}_n - \boldsymbol{m}_k)], \tag{5}$$

with $\psi(\cdot)$ being the logarithmic derivative of the gamma function (also known as the digamma function). We observe that $q^*(\boldsymbol{Z})$ follows the same categorical distribution as the likelihood, and it is parameterized by o_{nk}. More details can be found in Appendix.

Proposed algorithm The proposed method combines PLDA and VB inference which leads to an Efficiency Guided Adaptive Dimension Reduction for VAriational BAyesian inference. We thus name our proposed method "BAVARDAGE", and the detailed description is presented in Algorithm 1. Given a Few-Shot task $\mathbf{X}$ and a within-class scatter matrix $\mathbf{S}_w^{base}$, we initialize o_{nk} using EM algorithm with an assumed covariance matrix, adjusted by a temperature hyper-parameter T_{km}, for all classes. Note that this is equivalent to Soft-KMEANS [20] algorithm. And for each iteration we update parameters: in M step we update α_k, β_k and centroids $\boldsymbol{m}_k$, in E step we only update o_{nk}, and we apply PLDA with the updated o_{nk} to reduce feature dimensions. Finally, predicted labels are obtained by selecting the class that corresponds to the largest value in o_{nk}.

The illustration of our proposed method is presented in Figure 1. For a Few-Shot task that has three classes (red, blue and green) with unlabelled samples depicted on the probability simplex, we firstly initialize o_{nk} with Soft-KMEANS which directs some data points to their belonging classes while further distancing some points from their targeted classes. Then we apply the proposed VB inference integrated with PLDA, resulting in additional points moving towards their corresponding classes.

Algorithm 1 BAVARDAGE

Inputs: $\mathbf{X} \in \mathbb{R}^{N \times D}$, $\mathbf{S}_w^{base} \in \mathbb{R}^{D \times D}$
Hyper-parameters: $T_{km},\ T_{vb},\ s_{max}$
Priors for VB: $\alpha_o = 2.0,\ \beta_o = 10.0,\ \boldsymbol{m}_o = 0,\ \boldsymbol{\Lambda} = T_{vb} \cdot \mathbf{I}$
Initializations: $o_{nk} = \text{EM}\ (\mathbf{X},\ T_{km})$
for $i = 1$ **to** n_{step} **do**
  $\mathbf{U} = \text{PLDA}\ (\mathbf{X},\ \mathbf{S}_w^{base},\ s_{max},\ o_{nk})$ # See more details in Appendix.
  VB (M step):
  $\alpha_k = \alpha_o + \sum_{n=1}^{N} o_{nk}$
  $\beta_k = \beta_o + \sum_{n=1}^{N} o_{nk}$
  $\boldsymbol{m}_k = \frac{1}{\beta_k}(\beta_o \boldsymbol{m}_o + \sum_{n=1}^{N} o_{nk} \boldsymbol{u}_n)$
  VB (E step):
  $\log \rho_{nk} = \psi(\alpha_k) - \psi(\sum_{j=1}^{K} \alpha_j) + \frac{1}{2} \log |\boldsymbol{\Lambda}| - \frac{d}{2} \log 2\pi - \frac{1}{2}[d\beta_k^{-1} + (\boldsymbol{u}_n - \boldsymbol{m}_k)^T \boldsymbol{\Lambda} (\boldsymbol{u}_n - \boldsymbol{m}_k)]$
  $o_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^{K} \rho_{nj}}$
end for
return $\hat{\ell}(\boldsymbol{x}_n) = \arg\max_k (o_{nk})$

4 Experiments

In this section we provide details on the standard transductive Few-Shot classification settings, and we evaluate the performance of our proposed method.

Benchmarks We test our method on standard Few-Shot benchmarks: *mini*-Imagenet [35], *tiered*-Imagenet [32] and caltech-ucsd birds-200-2011 (CUB) [41]. *mini*-Imagenet is a subset of ILSVRC-12 [35] dataset, it contains a total of $60,000$ images of size 84×84 belonging to 100 classes (600 images per class) and follows a 64-16-20 split for base, validation and novel classes. *tiered*-Imagenet is a larger subset of ILSVRC-12 containing 608 classes with $779,165$ images of size 84×84 in total, and we use the standard 351-97-160 split, and CUB is composed of 200 classes following a 100-50-50 split (Image size: 84×84). In Appendix we also show the performance of our proposed method on other well-known Few-Shot benchmarks such as FC100 [30] and CIFAR-FS [4].

Settings Following previous works [25, 34, 39], our proposed method is evaluated on 1-shot 5-way ($K = 5,\ L = 1$), and 5-shot 5-way ($K = 5,\ L = 5$) scenarios. As for the query set, we set a total number of $Q = 75$ unlabelled samples, from which we further define two settings: 1) a balanced setting where unlabelled instances are evenly distributed among K classes, and 2) an unbalanced setting where the query set is randomly distributed, following a Dirichlet distribution parameterized by α_o^*. In our paper we follow the same setting as [39] and set $\alpha_o^* = 2.0$, further experiments with different values are conducted in the next sections. The performance of our proposed method is evaluated by computing the averaged accuracy over $10,000$ Few-Shot tasks.

Implementation details In this paper we firstly compare our proposed algorithm with the other state-of-the-art methods using the same pretrained backbones and benchmarks provided in [39]. Namely we extract the features using the same ResNet-18 (RN18) and WideResNet28_10 (WRN) neural models, and present the performance on *mini*-Imagenet, *tiered*-Imagenet and CUB datasets. In our proposed method, the raw features are preprocessed following [42]. As for the hyper-parameters, we set $T_{km} = 10, T_{vb} = 50, s_{max} = 2$ for the balanced setting; $T_{km} = 50, T_{vb} = 50, s_{max} = 1$ for the unbalanced setting, and we use the same VB priors for all settings. To further show the functionality of our proposed method on different backbones and other benchmarks, we tested BAVARDAGE on a recent high performing feature extractor trained on a ResNet-12 (RN12) neural model [28, 3], and we report the accuracy in Table 1 and in Appendix with various settings.

4.1 Main results

The main results on the relevant settings are presented in Table 1. Note that we report the accuracy of other methods following [39], and add the performance of our proposed method in comparison, using the same pretrained RN18 and WRN feature extractors, and we also report the result of a RN12 backbone pretrained following [3]. We observe that our proposed algorithm reaches state-of-the-art performance for nearly all referenced datasets in the unbalanced setting, surpassing previous methods by a noticeable margin especially on 1-shot. In the balanced setting we also reach competitive accuracy compared with [16] along with other works that make use of a perfectly balanced prior on unlabelled samples, while our proposed method suggests no such prior. In addition, we provide results on the other Few-Shot benchmarks with different settings in Appendix.

Table 1: Comparisons of the state-of-the-art methods on *mini*-Imagenet, *tiered*-Imagenet and CUB datasets using the same pretrained backbones as [39], along with the accuracy of our proposed method on a ResNet-12 backbone pretrained following [3].

***mini*-Imagenet**		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
MAML [12]		47.6/−	64.5/−	51.4/−	69.5/−
Versa [15]		47.8/−	61.9/−	50.0/−	65.6/−
Entropy-min [11]		58.5/60.4	74.8/76.2	63.6/66.1	82.1/84.2
PT-MAP [16]		60.1/60.6	67.1/66.8	**76.9**/**78.9**	**85.3**/86.6
LaplacianShot [48]	RN18/WRN [39]	65.4/70.0	81.6/83.2	70.1/72.9	82.1/83.8
BD-CSPN [26]		67.0/70.4	80.2/82.3	69.4/72.5	82.0/83.7
TIM [5]		67.3/69.8	79.8/81.6	71.8/74.6	83.9/85.9
α-TIM [39]		67.4/69.8	82.5/84.8	−/−	−/−
BAVARDAGE (ours)		**71.0**/**74.1**	**83.6**/**85.5**	75.1/78.5	84.5/**87.4**
BAVARDAGE (ours)	RN12 [3]	**77.8**	**88.0**	**82.7**	**89.5**
***tiered*-Imagenet**		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Entropy-min [11]		61.2/62.9	75.5/77.3	67.0/68.9	83.1/84.8
PT-MAP [16]		64.1/65.1	70.0/71.0	**82.9**/**84.6**	**88.8**/**90.0**
LaplacianShot [48]		72.3/73.5	85.7/86.8	77.1/78.8	86.2/87.3
BD-CSPN [26]	RN18/WRN [39]	74.1/75.4	84.8/85.9	76.3/77.7	86.2/87.4
TIM [5]		74.1/75.8	84.1/85.4	78.6/80.3	87.7/88.9
α-TIM [39]		74.4/76.0	**86.6**/**87.8**	−/−	−/−
BAVARDAGE (ours)		**76.6**/**77.5**	86.5/87.5	80.3/81.5	87.1/88.3
BAVARDAGE (ours)	RN12 [3]	**79.4**	**88.0**	**83.5**	**89.0**
CUB		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
PT-MAP [16]		65.1	71.3	85.5	91.3
Entropy-min [11]		67.5	82.9	72.8	88.9
LaplacianShot [48]		73.7	87.7	78.9	88.8
BD-CSPN [26]	RN18 [39]	74.5	87.1	77.9	88.9
TIM [5]		74.8	86.9	80.3	90.5
α-TIM [39]		75.7	89.8	−	−
BAVARDAGE (ours)		**82.0**	**90.7**	**85.6**	**91.4**
BAVARDAGE (ours)	RN12 [3]	**83.1**	**90.8**	**87.4**	**92.0**

4.2 Ablation studies

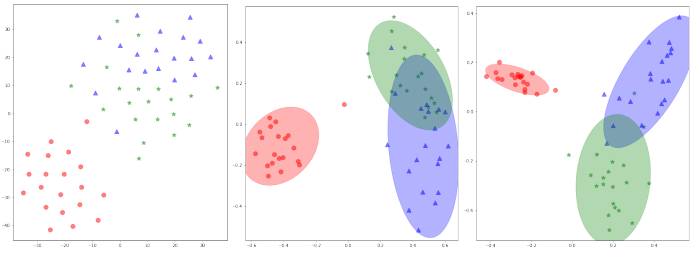
Analysis on the elements of BAVARDAGE In this experiment we dive into our proposed method and conduct an ablation study on the impact of each element. Namely, we report the performance in the following 3 scenarios: 1) only run Soft-KMEANS on the extracted features to obtain a baseline accuracy; 2) run the VB model with o_{nk} initialized by Soft-KMEANS, without reducing the feature

space; and 3) integrate PLDA into VB iterations. From Table 2 we observe only a slight increase of accuracy compared with baseline when no dimensionality reduction is applied. This is due to the fact that high feature dimensions increase uncertainty in the estimations, making the model sensitive to parameters. With our implementation of PLDA iteratively applied in the VB model, we can see from the table that the performance increases by a relatively large margin, suggesting the effectiveness of our proposed adaptive dimension reduction method.

Table 2: Ablation study on the elements of our proposed method, with results tested on *mini*-Imagenet (backbone: WRN) and CUB (backbone: RN18) in the unbalanced setting.

			***mini*-Imagenet**		**CUB**	
Soft-KMEANS	VB	PLDA	**1-shot**	**5-shot**	**1-shot**	**5-shot**
✓			71.4	82.4	77.5	86.7
✓	✓		71.8	82.5	77.8	87.2
✓	✓	✓	**74.1**	**85.5**	**82.0**	**90.7**

Visualization of features for different projections To further showcase the effect of proposed PLDA, in Fig. 2 we visualize the extracted features of a 3-way Few-Shot task in the following 3 scenarios: (a) features in the original space, using T-SNE [38] for visualization purpose; (b) features that are projected directly onto their centroids space, and finally (c) features projected using PLDA. The ellipses drawn in (b) and (c) are the cluster estimations computed using the real labels of data samples, and we can thus observe a larger separation of different clusters with PLDA projection for the task in which the original features overlap heavily between clusters in blue and green.



(a) T-SNE (b) Centroids projection (c) PLDA

Figure 2: Visualization of extracted features of a Few-Shot task using different projection methods (dataset: *mini*-Imagenet, backbone: WRN), we report a 86.7%, 90.0% and 95.0% prediction accuracy corresponding to each projection.

Robustness against imbalance In Table 1 we show the accuracy of our proposed method using VB priors introduced in Section 3.3, in which α_o is set to be equal to the Dirichlet's parameter α_o^* for the level of imbalance in the query set. Therefore, in this experiment we test the robustness of BAVARDAGE, namely in Fig. 3 we alter α_o and report the accuracy on different imbalance levels (varying α_o^*) in both 1-shot and 5-shot settings. Note that the proposed model becomes slightly more sensitive to α_o when the level of imbalance increases (smaller α_o^*), with an approximate 1% drop of accuracy when increasing α_o in the case of $\alpha_o^* = 1$.

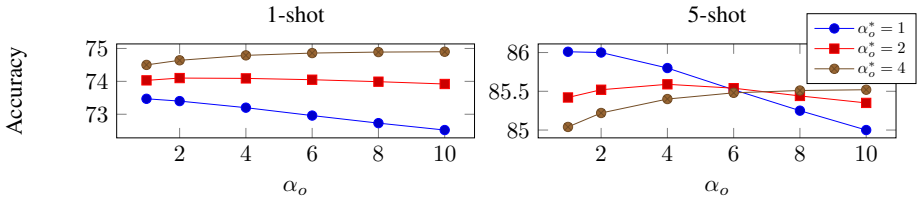


Figure 3: 1-shot and 5-shot accuracy on different imbalance levels (varying α_o^*) as a function of VB priors α_o (dataset: *mini*-Imagenet, backbone: WRN).

Varying Few-Shot settings In this experiment we observe the performance of BAVARDAGE on different Few-Shot settings, namely we vary the number of labelled samples per class L as well as the total number of unlabelled samples Q in a task, for further comparison we also report the accuracy using only Soft-KMEANS algorithm. In Fig. 4 we can observe constant higher accuracy of our proposed method, and a slightly larger difference gap when Q increases.

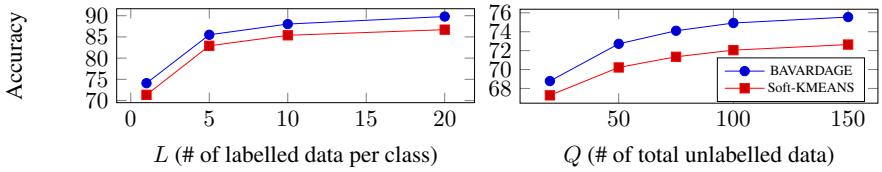


Figure 4: Accuracy as a funtion of L and Q in comparison with Soft-KMEANS (dataset: *mini*-Imagenet, backbone: WRN).

5 Conclusion

In this paper we proposed a clustering method based on Variational Bayesian Inference and Probabilistic Linear Discriminant Analysis for transductive Few-Shot Classification. BAVARDAGE has reached state-of-the-art accuracy on nearly all Few-Shot benchmarks in the realistic unbalanced setting, as well as competitive performance in the balanced setting without using a perfectly class-balanced prior. As our proposed method assumes a shared isotropic covariance matrix for all clusters, the estimations in VB models could be limited. Therefore the future work could study a better estimation of covariance matrices associated with each cluster. An interesting asset of the proposed method is that it performs most of its processing in a reduced $(K-1)$-dimensional space, where K is the number of classes, suggesting interests for visualization and suitability for more elaborate statistical machine learning methods. As in [39], we encourage the community to rethink the works in transductive settings to provide fairer grounds of comparison between the various proposed approaches.

6 Appendix

6.1 Implementation details on the proposed PLDA

In this section we present more details on our implementation of PLDA proposed in section 3.2 in the paper. Given $\mathbf{X} \in \mathbb{R}^{N \times D}$, we estimate its within-class covariance matrix to be $\mathbf{S}_w^{base}$ calculated from $\mathbf{D}_{base}$. Denote $\mathcal{I}_c^{base}$ as the set of samples belonging to base class c where $c \in 1, ..., |\mathcal{C}_{base}|$, therefore $\mathbf{\Phi}_w$ is approximated as follows:

$$\mathbf{\Phi}_w \approx \mathbf{S}_w^{base} = \frac{\sum_c \sum_{i \in \mathcal{I}_c^{base}} (\boldsymbol{x}_i^{base} - \boldsymbol{m}_c^{base})(\boldsymbol{x}_i^{base} - \boldsymbol{m}_c^{base})^T}{N_{base}}, \tag{6}$$

where $\boldsymbol{m}_c^{base} = \frac{1}{|\mathcal{I}_c^{base}|} \sum_{i \in \mathcal{I}_c^{base}} \boldsymbol{x}_i^{base}$ is the mean of c-th base class. Let $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_i, ..., \lambda_D] \in \mathbb{R}^D$ be the eigenvalues of $\mathbf{S}_w^{base}$ in descending order, and we set $\mathbf{R} = [\boldsymbol{r}_1, ..., \boldsymbol{r}_i, ..., \boldsymbol{r}_D] \in \mathbb{R}^{D \times D}$

Algorithm 2 Proposed PLDA

Fonction PLDA ($\mathbf{X},\ \mathbf{S}_w^{base},\ s_{max},\ o_{nk}$)
  Sphere $\mathbf{X}$ using $\mathbf{T}$ (Eq. 7), obtain $\mathbf{X}'$.
  Estimate centroids $\boldsymbol{m}'_k$ using o_{nk} (Eq. 8).
  Compute $\boldsymbol{\Psi}$ using $\boldsymbol{m}'_k$ (Eq. 9).
  Project $\mathbf{X}'$ onto the centroids space, obtain $\mathbf{U}$.
Return $\mathbf{U}$

to be the corresponding eigenvectors. In this paper we define a transformation matrix $\mathbf{T} = \mathbf{SR}$ where $\mathbf{S}$ is a diagonal matrix with diagonal values being the square root of multiplicative inverse of $\boldsymbol{\lambda}$, clamped to an upper bound s_{max}. Namely, $\boldsymbol{s} = diag(\mathbf{S})$ where $\boldsymbol{s} = [s_1, ...s_i, ...s_D] \in \mathbb{R}^D$ is a D-length vector containing the scaling value for each dimension, and we set s_i to be as follows:

$$s_i = \begin{cases} \lambda_i^{-0.5} & \text{if} \quad \lambda_i^{-0.5} \leq s_{max} \\ s_{max} & otherwise \end{cases} . \tag{7}$$

We can see from Eq. 7 that $\mathbf{T}$ is composed of a rotation matrix and scaling values on feature dimensions that help morph the within-class distribution into an identity covariance matrix. This corresponds to a data sphering/whitening process in which we decorrelate samples in each of the dimensions. In our implementation we transform $\mathbf{X}$ by multiplying it with $\mathbf{T}$. Therefore the sphered data samples, denoted as $\mathbf{X}' = [\boldsymbol{x}'_1, ...\boldsymbol{x}'_n, ...\boldsymbol{x}'_N]^T \in \mathbb{R}^{N \times D}$, are obtained from $\boldsymbol{x}'_n = \mathbf{T}\boldsymbol{x}_n$.

Next, we project $\mathbf{X}'$ onto a subspace that corresponds to the $K-1$ largest eigenvalues of its between-scatter matrix. Denote $\boldsymbol{m}'_k$ as the estimated centroid for class k, given soft class assignments o_{nk} ($1 \leq n \leq N, 1 \leq k \leq K$), $\boldsymbol{m}'_k$ is computed as:

$$\boldsymbol{m}'_k = \frac{\sum_{n=1}^N o_{nk}\boldsymbol{x}'_n}{\gamma + N_k},\ N_k = \sum_{n=1}^N o_{nk} , \tag{8}$$

where γ is used as an offset indicating how close the centroids are to 0, in this paper we set it to 10.0, same as β_o in the VB model in reduced space. Therefore, the between-class scatter matrix $\boldsymbol{\Psi}$ of sphered samples can be calculated as:

$$\boldsymbol{\Psi} = \sum_{k=1}^K (\boldsymbol{m}'_k - \boldsymbol{m}')(\boldsymbol{m}'_k - \boldsymbol{m}')^T , \tag{9}$$

where $\boldsymbol{m}' = \frac{1}{K}\sum_{k=1}^K \boldsymbol{m}'_k$ is the mean of estimated class centroids. Then we project $\mathbf{X}'$ onto a d-length subspace, where $d = K-1$. In details, denote $\mathbf{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_i, ..., \boldsymbol{v}_d] \in \mathbb{R}^{D \times d}$ to be the eigenvectors corresponding to the d largest eigenvalues of $\boldsymbol{\Psi}$, the projected data $\mathbf{U}$ are obtained as $\boldsymbol{u}_n = \mathbf{V}^T\boldsymbol{x}'_n$ for each sample. Note that the formulation of $\boldsymbol{\Psi}$ in Eq. 9 allows at most $K-1$ non-zero eigenvalues, therefore the resulting subspace projection using these eigenvectors is equivalent to a projection onto the affine subspace containing the centroids $\boldsymbol{m}'_k$. Furthermore, according to Eq. 1 in the paper, we can further deduce the projection matrix $\mathbf{W}$ to be as follows:

$$\begin{aligned} \boldsymbol{u}_n = \mathbf{W}^T\boldsymbol{x}_n = \mathbf{V}^T\boldsymbol{x}'_n = \mathbf{V}^T\mathbf{T}\boldsymbol{x}_n = \mathbf{V}^T\mathbf{SR}\boldsymbol{x}_n, \\ \implies \mathbf{W} = (\mathbf{V}^T\mathbf{SR})^T = \mathbf{R}^T\mathbf{SV}. \end{aligned} \tag{10}$$

The entire process is described in Algorithm 2.

6.2 Implementation details on the proposed VB model

In this section we provide more detailed explanation of our proposed VB model. Given a posterior $p(\theta|\mathbf{U})$, we approximate it with a function variational distribution $q(\theta)$ by minimizing the Kullback-

Leibler divergence:

$$\begin{aligned} q^*(\theta) &= \arg\min_q\{D_{KL}(q||p)\} \\ &= \arg\min_q\{\log p(\mathbf{U}) - \mathcal{L}(q)\} \\ &= \arg\max_q\{\mathcal{L}(q)\} \end{aligned} \tag{11}$$

where the evidence $\log p(\mathbf{U})$ is considered fixed, and $\mathcal{L}(q) = \int q(\theta)\log\frac{p(\theta,\mathbf{U})}{q(\theta)}d\theta$ stands for Evidence Lower BOund (ELBO) providing "evidence" that we have chosen the right model. We can see that minimizing the Kullback-Leibler divergence is equivalent to maximizing the ELBO. Suppose $\theta = \{\theta_1, ..., \theta_m, ..., \theta_M\}$, we firstly factorize $q(\theta) = \prod_{m=1}^{M} q(\theta_m)$ according to the Mean-Field assumption, then we solve each term individually:

$$\begin{aligned} \mathcal{L}(q) &= \int q(\theta)\log\frac{p(\theta,\mathbf{U})}{q(\theta)}d\theta \\ &= \int\left(\prod_{m=1}^{M} q(\theta_m)\right)\left(\log p(\theta,\mathbf{U}) - \sum_{m=1}^{M}\log q(\theta_m)\right)d\theta_1 d\theta_2 ... d\theta_M \\ &= \sum_{m=1}^{M}\left(\int q(\theta_m)\left(\int q(\theta_{-m})\log p(\theta,\mathbf{U})d\theta_{-m}\right)d\theta_m - \int q(\theta_m)\log q(\theta_m)d\theta_m\right), \end{aligned} \tag{12}$$

and the ELBO is maximized when:

$$\log q^*(\theta_m) = \mathbb{E}_{\theta_{-m}}[\log p(\theta,\mathbf{U})] + const, \tag{13}$$

where $\mathbb{E}_{\theta_{-m}}[\cdot]$ stands for the expectation with respect to all variables in θ except θ_m. In our method we define $\theta = \{\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}\}$, the detailed formula of some variables are presented as follows:

$$\begin{aligned} \boldsymbol{z}_n &= [z_{n1}, ..., z_{nk}, ..., z_{nK}] \in \{0,1\}^K, \quad \sum_{k=1}^{K} z_{nk} = 1, \\ \boldsymbol{\pi} &= [\pi_1, ..., \pi_k, ..., \pi_K], \quad \pi_k \geq 0, \quad \sum_{k=1}^{K}\pi_k = 1. \end{aligned} \tag{14}$$

According to Bayes' theorem, we rewrite the posterior to be:

$$\begin{aligned} p(\theta|\mathbf{U}) = p(\boldsymbol{Z},\boldsymbol{\pi},\boldsymbol{\mu}|\mathbf{U}) &= \frac{p(\boldsymbol{Z},\boldsymbol{\pi},\boldsymbol{\mu},\mathbf{U})}{p(\mathbf{U})} \\ &= \frac{p(\mathbf{U}|\boldsymbol{Z},\boldsymbol{\mu})p(\boldsymbol{Z}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\mu})}{p(\mathbf{U})}, \end{aligned} \tag{15}$$

in which:

$$\begin{aligned} p(\mathbf{U}|\boldsymbol{Z},\boldsymbol{\mu}) &= \prod_{n=1}^{N}\prod_{k=1}^{K}\mathcal{N}(\boldsymbol{u}_n|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}^{-1})^{z_{nk}}, \\ p(\boldsymbol{Z}|\boldsymbol{\pi}) &= \prod_{n=1}^{N} Categorical(\boldsymbol{z}_n|\boldsymbol{\pi}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}, \\ p(\boldsymbol{\pi}) &= Dir(\boldsymbol{\pi}|\alpha_o) = \frac{\Gamma(\sum_{k=1}^{K}K\alpha_o)}{\prod_{k=1}^{K}\Gamma(\alpha_o)}\prod_{k=1}^{K}\pi_k^{\alpha_o-1} = C(\alpha_o)\prod_{k=1}^{K}\pi_k^{\alpha_o-1}, \\ p(\boldsymbol{\mu}) &= \prod_{k=1}^{K}\mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_o,(\beta_o\boldsymbol{\Lambda})^{-1}). \end{aligned} \tag{16}$$

According to Eq. 13, $q^*(\boldsymbol{\pi})$ can be computed as follows:

$$
\begin{aligned}
\log q^*(\boldsymbol{\pi}) &= \mathbb{E}_{\boldsymbol{Z},\boldsymbol{\mu}}[\log p(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{U})] + const \\
&= \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{Z}|\boldsymbol{\pi})] + \log p(\boldsymbol{\pi}) + const \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{Z}}[z_{nk}] \log \pi_k + \sum_{k=1}^{K}(\alpha_o - 1)\log \pi_k + const \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N} o_{nk} \log \pi_k + \sum_{k=1}^{K}(\alpha_o - 1)\log \pi_k + const \\
&= \sum_{k=1}^{K}(N_k + \alpha_o - 1)\log \pi_k + const, \\
\implies q^*(\boldsymbol{\pi}) &= \prod_{k=1}^{K} \pi_k^{N_k + \alpha_o - 1} + const \\
&= \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} + const \\
&= Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}).
\end{aligned}
\tag{17}
$$

Similarly for $q^*(\boldsymbol{\mu})$ we can compute it as shown below:

$$
\begin{aligned}
\log q^*(\boldsymbol{\mu}) &= \mathbb{E}_{\boldsymbol{Z},\boldsymbol{\pi}}[\log p(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{U})] + const \\
&= \mathbb{E}_{\boldsymbol{Z}}[\log p(\mathbf{U}|\boldsymbol{Z}, \boldsymbol{\mu})] + \log p(\boldsymbol{\mu}) + const \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{Z}}[z_{nk}] \log \mathcal{N}(\boldsymbol{u}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}^{-1}) + \sum_{k=1}^{K} \log \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_o, (\beta_o \boldsymbol{\Lambda}^{-1}) + const \\
&= \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} o_{nk} \log |\boldsymbol{\Lambda}| - \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} o_{nk}(\boldsymbol{u}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Lambda}(\boldsymbol{u}_n - \boldsymbol{\mu}_k) \\
&+ \frac{1}{2}\sum_{k=1}^{K} \log |\beta_o \boldsymbol{\Lambda}| - \frac{1}{2}\sum_{k=1}^{K}(\boldsymbol{\mu}_k - \boldsymbol{m}_o)^T \beta_o \boldsymbol{\Lambda}(\boldsymbol{\mu}_k - \boldsymbol{m}_o).
\end{aligned}
\tag{18}
$$

To compute β_k, we gather the quadratic terms that contain $\boldsymbol{\mu}_k$ in Eq. 18:

$$
\begin{aligned}
(quad) &= -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} o_{nk} \boldsymbol{\mu}_k^T \boldsymbol{\Lambda} \boldsymbol{\mu}_k - \frac{1}{2}\sum_{k=1}^{K} \boldsymbol{\mu}_k^T \beta_o \boldsymbol{\Lambda} \boldsymbol{\mu}_k \\
&= -\frac{1}{2}\sum_{k=1}^{K} \boldsymbol{\mu}_k^T (N_k \boldsymbol{\Lambda} + \beta_o \boldsymbol{\Lambda}) \boldsymbol{\mu}_k \\
&= -\frac{1}{2}\sum_{k=1}^{K} \boldsymbol{\mu}_k^T (\beta_o + N_k) \boldsymbol{\Lambda}_k \boldsymbol{\mu}_k, \\
&\implies \beta_k = \beta_o + N_k.
\end{aligned}
\tag{19}
$$

As for $\boldsymbol{m}_k$, we gather the linear terms that contain $\boldsymbol{\mu}_k$ in Eq. 18:

$$
\begin{aligned}
(linear) &= \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} o_{nk}\boldsymbol{\mu}_k^T\boldsymbol{\Lambda}\boldsymbol{u}_n + \frac{1}{2}\sum_{k=1}^{K}\boldsymbol{\mu}_k^T\beta_o\boldsymbol{\Lambda}\boldsymbol{m}_o \\
&= \frac{1}{2}\sum_{k=1}^{K}\boldsymbol{\mu}_k^T\boldsymbol{\Lambda}(\beta_o\boldsymbol{m}_o + \sum_{n=1}^{N} o_{nk}\boldsymbol{u}_n) \\
&= \frac{1}{2}\sum_{k=1}^{K}\boldsymbol{\mu}_k^T\beta_k\boldsymbol{\Lambda}\boldsymbol{m}_k, \\
&\Longrightarrow \boldsymbol{m}_k = \frac{1}{\beta_k}(\beta_o\boldsymbol{m}_o + \sum_{n=1}^{N} o_{nk}\boldsymbol{u}_n).
\end{aligned} \tag{20}
$$

Therefore $q^*(\boldsymbol{\mu})$ can be reformulated as:

$$
q^*(\boldsymbol{\mu}) = \prod_{k=1}^{K} q^*(\boldsymbol{\mu}_k) = \prod_{k=1}^{K}\mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\beta_k\boldsymbol{\Lambda})^{-1}). \tag{21}
$$

We also provide a more detailed calculation of $q^*(\mathbf{Z})$:

$$
\begin{aligned}
\log q^*(\boldsymbol{Z}) &= \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\mu}}[\log p(\boldsymbol{Z},\boldsymbol{\pi},\boldsymbol{\mu},\mathbf{U})] + const \\
&= \mathbb{E}_{\boldsymbol{\pi}}[\log p(\boldsymbol{Z}|\boldsymbol{\pi})] + \mathbb{E}_{\boldsymbol{\mu}}[\log p(\mathbf{U}|\boldsymbol{Z},\boldsymbol{\mu})] + const \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\left(\mathbb{E}_{\boldsymbol{\pi}}[\log\pi_k] + \mathbb{E}_{\boldsymbol{\mu}}[\log\mathcal{N}(\boldsymbol{u}_n|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}^{-1})]\right) + const \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\log\rho_{nk} + const,
\end{aligned} \tag{22}
$$

where

$$
\begin{aligned}
\log\rho_{nk} &= \mathbb{E}_{\boldsymbol{\pi}}[\log\pi_k] + \mathbb{E}_{\boldsymbol{\mu}}[\log\mathcal{N}(\boldsymbol{u}_n|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}^{-1})] \\
&= \mathbb{E}_{\boldsymbol{\pi}}[\log\pi_k] + \frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{d}{2}\log 2\pi - \frac{1}{2}\mathbb{E}_{\boldsymbol{\mu}}[(\boldsymbol{u}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}(\boldsymbol{u}_n-\boldsymbol{\mu}_k)].
\end{aligned} \tag{23}
$$

Therefore $q^*(\mathbf{Z})$ can be expressed as:

$$
q^*(\boldsymbol{Z}) = \prod_{n=1}^{N}\prod_{k=1}^{K} o_{nk}^{z_{nk}} = \prod_{n=1}^{N} Categorical(\boldsymbol{z}_n|\boldsymbol{o}_n), \quad o_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^{K}\rho_{nj}}, \tag{24}
$$

we can see that the variable follows a categorical distribution, parameterized by o_{nk}, and $o_{nk} = \mathbb{E}_{\boldsymbol{Z}}[z_{nk}]$. As for Eq. 23, more details are shown as follows:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\pi}}[\log\pi_k] &= \psi(\alpha_k) - \psi(\sum_{j=1}^{K}\alpha_j), \\
\mathbb{E}_{\boldsymbol{\mu}}[(\boldsymbol{u}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}(\boldsymbol{u}_n-\boldsymbol{\mu}_k)] &= \int(\boldsymbol{u}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}(\boldsymbol{u}_n-\boldsymbol{\mu}_k)q^*(\boldsymbol{\mu}_k)d\boldsymbol{\mu}_k \\
&= (\boldsymbol{u}_n-\boldsymbol{m}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{u}_n-\boldsymbol{m}_k) + \mathrm{Tr}[\boldsymbol{\Lambda}\cdot(\beta_k\boldsymbol{\Lambda})^{-1}] \\
&= d\beta_k^{-1} + (\boldsymbol{u}_n-\boldsymbol{m}_k)^T\boldsymbol{\Lambda}(\boldsymbol{u}_n-\boldsymbol{m}_k),
\end{aligned} \tag{25}
$$

$\psi(\cdot)$ is the logarithmic derivative of the gamma function, and the distribution for π_k and $\boldsymbol{\mu}_k$ follows Eq. 17 and 21. Therefore:

$$
\log\rho_{nk} = \psi(\alpha_k) - \psi(\sum_{j=1}^{K}\alpha_j) + \frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{d}{2}\log 2\pi - \frac{1}{2}[d\beta_k^{-1} + (\boldsymbol{u}_n-\boldsymbol{m}_k)^T\boldsymbol{\Lambda}(\boldsymbol{u}_n-\boldsymbol{m}_k)]. \tag{26}
$$

From the above equations we observe a dependency between priors and posteriors, which can be estimated iteratively depending on the class allocations. Therefore in this paper we propose to solve it under a basic Expectation Maximization framework where we estimate o_{nk} in the E-step, while updating α_k, β_k and $\boldsymbol{m}_k$ in the M-step.

6.3 Hyperparameter tuning

In this section we detail about how the hyperparameters in our proposed method are obtained. Namely, for a standard Few-Shot benchmark that has been split into base-validation-novel class set, we firstly tune our model using validation set and choose the hyperparameters accordingly before applying to the novel set. For example in Figure 5 we tune two temperature parameters T_{km}, T_{vb}, the scaling up-bound parameter s_{max} and the VB prior β_o that are used in our proposed BAVARDAGE. The blue curves show the performance on validation set while the red curves show the accuracy on the novel set (benchmark: *mini*-Imagenet). From the figure we see a similar behavior between two sets in terms of performance, T_{km} has little impact on the accuracy, same for T_{vb} when it is large. For s_{max} we observe an uptick when it is around 1, followed by a slowing decrease and finally stabilizing to the same accuracy when it becomes larger. In this paper we tune hyperparameters for each benchmark in the same way. For *tiered*-Imagenet we set T_{km}, T_{vb} and s_{max} to be 10, 100, 2 in the balanced setting, 100, 100, 1 in the unbalanced setting; for CUB we set them to be 10, 4, 5 in both balanced and unbalanced settings; and for FC100 and CIFAR-FS we set the hyperparameters to be the same as *mini*-Imagenet. As for β_o we set it to be 10 across datasets since it gives the best performance.

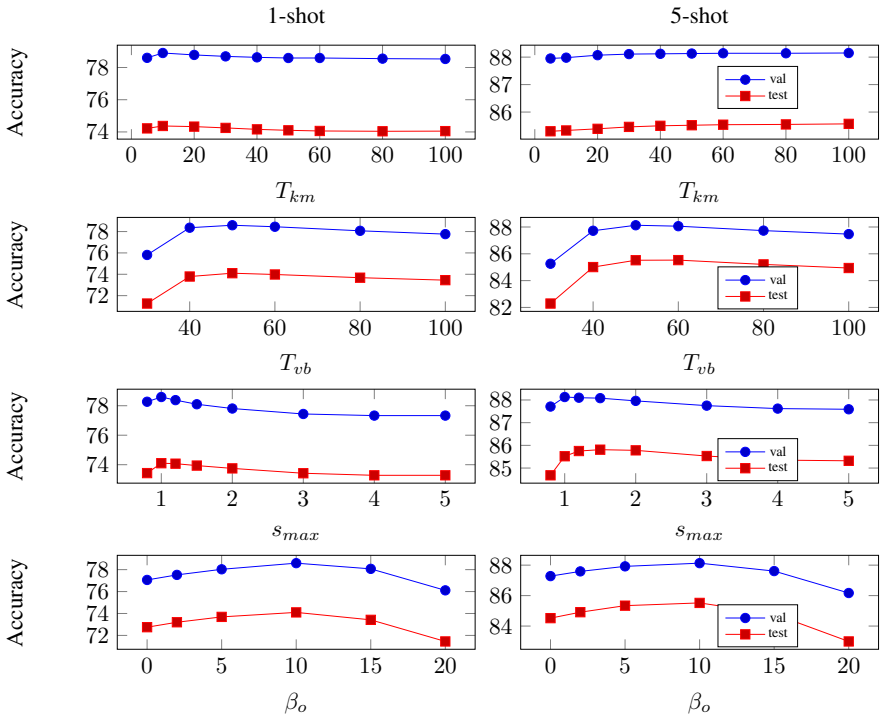


Figure 5: Hyperparameter tuning of our proposed method. Here we tune 4 hyperparameters of BAVARDAGE on *mini*-Imagenet (backbone: WRN) in the unbalanced setting.

Table 3: Detailed results of BAVARDAGE with confidence interval of 95% on the Few-Shot benchmarks, along with a baseline accuracy using Soft-KMEANS. We use RN18 and WRN pretrained from [39], RN12 and RN12* pretrained from [3].

***mini*-Imagenet**		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Soft-KMEANS	RN18 [39]	$68.82 \pm 0.27\%$	$81.27 \pm 0.17\%$	$73.47 \pm 0.26\%$	$83.04 \pm 0.15\%$
	WRN [39]	$71.35 \pm 0.27\%$	$82.41 \pm 0.16\%$	$75.70 \pm 0.25\%$	$84.42 \pm 0.14\%$
	RN12 [3]	$75.65 \pm 0.25\%$	$86.35 \pm 0.14\%$	$80.81 \pm 0.24\%$	$87.92 \pm 0.12\%$
	RN12* [3]	$77.51 \pm 0.26\%$	$87.78 \pm 0.14\%$	$82.14 \pm 0.24\%$	$89.08 \pm 0.12\%$
BAVARDAGE	RN18 [39]	$71.01 \pm 0.31\%$	$83.60 \pm 0.17\%$	$75.07 \pm 0.28\%$	$84.49 \pm 0.14\%$
	WRN [39]	$74.10 \pm 0.30\%$	$85.52 \pm 0.16\%$	$78.51 \pm 0.27\%$	$87.41 \pm 0.13\%$
	RN12 [3]	$77.85 \pm 0.28\%$	$88.02 \pm 0.14\%$	$82.67 \pm 0.25\%$	$89.50 \pm 0.11\%$
	RN12* [3]	$79.76 \pm 0.29\%$	$89.85 \pm 0.13\%$	$84.80 \pm 0.25\%$	$91.65 \pm 0.10\%$
***tiered*-Imagenet**		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Soft-KMEANS	WRN [39]	$73.92 \pm 0.28\%$	$85.02 \pm 0.18\%$	$78.59 \pm 0.27\%$	$85.76 \pm 0.16\%$
	RN18 [39]	$73.79 \pm 0.28\%$	$84.65 \pm 0.18\%$	$78.34 \pm 0.27\%$	$85.52 \pm 0.17\%$
	RN12 [3]	$78.15 \pm 0.27\%$	$87.65 \pm 0.17\%$	$83.11 \pm 0.25\%$	$88.80 \pm 0.15\%$
	RN12* [3]	$79.62 \pm 0.27\%$	$88.61 \pm 0.16\%$	$84.08 \pm 0.24\%$	$89.56 \pm 0.14\%$
BAVARDAGE	WRN [39]	$77.45 \pm 0.31\%$	$87.48 \pm 0.18\%$	$81.47 \pm 0.28\%$	$88.27 \pm 0.16\%$
	RN18 [39]	$76.55 \pm 0.31\%$	$86.46 \pm 0.19\%$	$80.32 \pm 0.28\%$	$87.14 \pm 0.16\%$
	RN12 [3]	$79.38 \pm 0.29\%$	$88.04 \pm 0.18\%$	$83.52 \pm 0.26\%$	$89.03 \pm 0.15\%$
	RN12* [3]	$81.17 \pm 0.29\%$	$89.63 \pm 0.17\%$	$85.20 \pm 0.25\%$	$90.41 \pm 0.14\%$
CUB		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Soft-KMEANS	RN18 [39]	$77.54 \pm 0.26\%$	$86.70 \pm 0.14\%$	$82.67 \pm 0.24\%$	$89.04 \pm 0.11\%$
	RN12 [3]	$81.24 \pm 0.25\%$	$87.27 \pm 0.14\%$	$84.87 \pm 0.22\%$	$89.64 \pm 0.11\%$
	RN12* [3]	$82.40 \pm 0.24\%$	$89.40 \pm 0.13\%$	$87.38 \pm 0.20\%$	$91.29 \pm 0.10\%$
BAVARDAGE	RN18 [39]	$82.00 \pm 0.28\%$	$90.67 \pm 0.12\%$	$85.64 \pm 0.25\%$	$91.42 \pm 0.10\%$
	RN12 [3]	$83.12 \pm 0.26\%$	$90.81 \pm 0.12\%$	$87.41 \pm 0.22\%$	$92.03 \pm 0.09\%$
	RN12* [3]	$86.96 \pm 0.24\%$	$92.84 \pm 0.10\%$	$90.42 \pm 0.20\%$	$93.50 \pm 0.08\%$
FC100		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Soft-KMEANS	RN12 [3]	$51.24 \pm 0.27\%$	$64.70 \pm 0.22\%$	$54.59 \pm 0.26\%$	$66.37 \pm 0.20\%$
	RN12* [3]	$51.64 \pm 0.27\%$	$65.26 \pm 0.22\%$	$54.87 \pm 0.26\%$	$66.89 \pm 0.20\%$
BAVARDAGE	RN12 [3]	$52.60 \pm 0.32\%$	$65.35 \pm 0.25\%$	$56.66 \pm 0.28\%$	$69.69 \pm 0.21\%$
	RN12* [3]	$53.78 \pm 0.30\%$	$68.75 \pm 0.24\%$	$57.27 \pm 0.29\%$	$70.60 \pm 0.21\%$
CIFAR-FS		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Soft-KMEANS	RN12 [3]	$80.72 \pm 0.25\%$	$88.31 \pm 0.17\%$	$85.47 \pm 0.22\%$	$89.36 \pm 0.15\%$
	RN12* [3]	$81.75 \pm 0.25\%$	$88.92 \pm 0.17\%$	$86.07 \pm 0.22\%$	$89.85 \pm 0.15\%$
BAVARDAGE	RN12 [3]	$82.68 \pm 0.27\%$	$88.97 \pm 0.18\%$	$86.20 \pm 0.23\%$	$89.58 \pm 0.15\%$
	RN12* [3]	$83.82 \pm 0.27\%$	$89.84 \pm 0.18\%$	$87.35 \pm 0.23\%$	$90.63 \pm 0.16\%$

6.4 Additional experiments on other Few-Shot benchmarks

In Section 4 in the paper we tested our proposed method on three standard Few-Shot benchmarks: *mini*-Imagenet[1], *tiered*-Imagenet[2] and CUB[3], following the same setting as presented in `https://github.com/oveilleux/Realistic_Transductive_Few_Shot`. In this section we further conduct experiments on two other well-known Few-Shot datasets: 1) FC100 (`https://github.com/ElementAI/TADAM`) is a recent split dataset based on CIFAR-100 [22] that contains 60 base classes for training, 20 classes for validation and 20 novel classes for evaluation, each class is composed of 600 images of size 32x32 pixels; 2) CIFAR-FS (`https://github.com/bertinetto/r2d2`) is also sampled from CIFAR-100 and shares the same quantity of classes in the base-validation-

[1] `https://github.com/yaoyao-liu/mini-imagenet-tools`

[2] `https://github.com/yaoyao-liu/tiered-imagenet-tools`

[3] `http://www.vision.caltech.edu/datasets/cub_200_2011`

novel splits as for *mini*-Imagenet. Each class contains 600 images of size 32x32 pixels. In Table 3 below we report the accuracy of our proposed method on all benchmarks, note that for FC100 and CIFAR-FS we believe to be among the first to conduct experiments in the unbalanced setting.

In Table 3 we also show the results using WRN and RN18 pretrained from [39] and RN12 pretrained from [3], same as Table 1 in the paper, with a confidence interval of 95% added next to the accuracy. In addition, given that some works [27, 47] in the field utilize data augmentation techniques to extract features based on images in original dimensions instead of reduced ones, here we apply our BAVARDAGE following the same setting and report the accuracy on a pretrained RN12 feature extractor [3] with data augmentation (denote RN12*). For comparison purpose we also provide a baseline accuracy on each Few-Shot benchmark using Soft-KMEANS algorithm.

With BAVARDAGE, we observe a clear increase of accuracy for all datasets compared with Soft-KMEANS in both balanced and unbalanced settings, suggesting the genericity of the proposed method. As for the computational time, we evaluate an average of 1.72 seconds per accuracy (on 10,000 Few-Shot tasks) using a GeForce RTX 3090 GPU.

References

[1] A. Antoniou, H. Edwards, and A. J. Storkey. How to train your MAML. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[2] S. Baik, J. Choi, H. Kim, D. Cho, J. Min, and K. M. Lee. Meta-learning with task-adaptive loss function for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9465–9474, 2021.

[3] Y. Bendou, Y. Hu, R. Lafargue, G. Lioi, B. Pasdeloup, S. Pateux, and V. Gripon. Easy: Ensemble augmented-shot y-shaped learning: State-of-the-art few-shot classification with simple ingredients. *arXiv preprint arXiv:2201.09699*, 2022.

[4] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[5] M. Boudiaf, I. Ziko, J. Rony, J. Dolz, P. Piantanida, and I. Ben Ayed. Information maximization for few-shot learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2445–2457. Curran Associates, Inc., 2020.

[6] T. Cao, M. T. Law, and S. Fidler. A theoretical analysis of the number of shots in few-shot learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[7] C. Chen, K. Li, W. Wei, J. T. Zhou, and Z. Zeng. Hierarchical graph neural networks for few-shot learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1):240–252, 2021.

[8] W. Chen, Y. Liu, Z. Kira, Y. F. Wang, and J. Huang. A closer look at few-shot classification. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[9] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[11] G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto. A baseline for few-shot image classification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

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

[13] C. W. Fox and S. J. Roberts. A tutorial on variational bayesian inference. *Artificial intelligence review*, 38(2):85–95, 2012.

[14] S. Gidaris and N. Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21–30, 2019.

[15] J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. E. Turner. Meta-learning probabilistic inference for prediction. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[16] Y. Hu, V. Gripon, and S. Pateux. Leveraging the feature distribution in transfer-based few-shot learning. In *International Conference on Artificial Neural Networks*, pages 487–499. Springer, 2021.

[17] S. Ioffe. Probabilistic linear discriminant analysis. In *European Conference on Computer Vision*, pages 531–542. Springer, 2006.

[18] T. S. Jaakkola and M. I. Jordan. Improving the mean field approximation via the use of mixture distributions. In *Learning in graphical models*, pages 163–173. Springer, 1998.

[19] D. Kang, H. Kwon, J. Min, and M. Cho. Relational embedding for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8822–8833, 2021.

[20] M. Kearns, Y. Mansour, and A. Y. Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Learning in graphical models*, pages 495–520. Springer, 1998.

[21] J. Kim, T. Kim, S. Kim, and C. D. Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2019.

[22] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[23] M. Lazarou, T. Stathaki, and Y. Avrithis. Iterative label cleaning for transductive and semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8751–8760, 2021.

[24] E. Lee, C.-H. Huang, and C.-Y. Lee. Few-shot and continual learning with attentive independent mechanisms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9455–9464, 2021.

[25] M. Lichtenstein, P. Sattigeri, R. Feris, R. Giryes, and L. Karlinsky. Tafssl: Task-adaptive feature sub-space learning for few-shot classification. In *European Conference on Computer Vision*, pages 522–539. Springer, 2020.

[26] J. Liu, L. Song, and Y. Qin. Prototype rectification for few-shot learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 741–756. Springer, 2020.

[27] X. Luo, L. Wei, L. Wen, J. Yang, L. Xie, Z. Xu, and Q. Tian. Rectifying the shortcut learning of background for few-shot learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[28] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, and V. N. Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2218–2227, 2020.

[29] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[30] B. Oreshkin, P. Rodríguez López, and A. Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018.

[31] O. V. Prezhdo. Mean field approximation for the stochastic schrödinger equation. *The Journal of chemical physics*, 111(18):8366–8377, 1999.

[32] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel. Meta-learning for semi-supervised few-shot classification. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[33] M. N. Rizve, S. Khan, F. S. Khan, and M. Shah. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10836–10846, 2021.

[34] P. Rodríguez, I. Laradji, A. Drouin, and A. Lacoste. Embedding propagation: Smoother manifold for few-shot classification. In *European Conference on Computer Vision*, pages 121–138. Springer, 2020.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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

[37] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[38] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[39] O. Veilleux, M. Boudiaf, P. Piantanida, and I. Ben Ayed. Realistic evaluation of transductive few-shot learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[40] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[41] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[42] Y. Wang, W. Chao, K. Q. Weinberger, and L. van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *CoRR*, abs/1911.04623, 2019.

[43] D. Wertheimer, L. Tang, and B. Hariharan. Few-shot classification with feature map reconstruction networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8012–8021, 2021.

[44] L. Yang, L. Li, Z. Zhang, X. Zhou, E. Zhou, and Y. Liu. Dpgn: Distribution propagation graph network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13390–13399, 2020.

[45] S. Yang, L. Liu, and M. Xu. Free lunch for few-shot learning: Distribution calibration. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[46] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020.

[47] C. Zhang, Y. Cai, G. Lin, and C. Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12203–12213, 2020.

[48] I. Ziko, J. Dolz, E. Granger, and I. B. Ayed. Laplacian regularized few-shot learning. In *International Conference on Machine Learning*, pages 11660–11670. PMLR, 2020.

5.3 Discussions

Our proposed method in this work is called "BAVARDAGE" , it reaches state-of-the-art performance in the unbalanced setting, especially in the case of 1 shot, BAVARDAGE is able to gain up to 6% accuracy compared with [Vei+21] under the same conditions. Furthermore, our proposed method also obtains competitive results in the balanced setting without a class-balanced prior, which is more practical for real world scenarios. In this section we will address more details of our proposed method.

5.3.1 Comparison with other dimension reduction techniques

In BAVARDAGE, we apply a PLDA to reduce feature dimension. Given the fact that PLDA projects data while reshaping them to have identity matrix as the covariance matrix, this corresponds to our assumption of a shared isotropic covariance matrix for the test data, and gives the best results. In comparison with other feature dimension techniques, here we provide the performance using 1) Principle Component Analysis (PCA) and 2) Linear Discriminant Analysis (LDA), with the same VB model as in the paper.

In detail, with PCA by applying it before the VB inference (since it is unsupervised), and we obtain 67.10%/76.95% accuracy for 1/5 shots in the unbalanced setting (dataset:*mini*-Imagenet, backbone: WRN from [Vei+21]). With LDA we apply it by computing the projection matrix from $\mathbf{\Phi}_w^{-1}\mathbf{\Phi}_b$ instead of Eq. (1) in the paper, and we obtain 70.87%/83.97% accuracy under the same setting, both inferior to the performance of reported BAVARDAGE (74.1%/85.5%), suggesting the effectiveness of PLDA.

5.3.2 Model complexity

Note that in our proposed method we do not apply a full VB model where the cluster covariances are regarded as hidden variables as well, instead we suppose a shared isotropic covariance matrix for all clusters, adjusted by a hyperparameter. This is due to the two following reasons: 1) a shared isotropic covariance corresponds to the assumption of PLDA that can been viewed as a whitening process; 2) injecting too many hidden variables may render the VB model more complex, unstable and sensitive to hyperparameters, especially in the case of few shot where there is already a relative high level of uncertainty in cluster estimations to begin with. Therefore, as we grant a VB model more flexibility on certain parameters, a balance should be maintained so that the model remains solid and does not collapse.

To better prove the point, we test the performance using 1) Kmeans and 2) a full VB model that are applied on the reduced dimensional data from PLDA (dataset:*mini*-Imagenet, backbone: WRN from [Vei+21]), and we obtain 70.36%/83.68% accuracy for 1/5 shots for 1), 48.56%/66.98% for 2), both inferior to the performance of BAVARDAGE reported in the paper (74.1%/85.5%). Therefore from our experiments, a partial VB model with a shared isotropic covariance matrix is shown to give the best results, although there is still room for the future work to find a workable solution for other forms of covariance matrix.

From the above results, we can observe a balance between model complexity and performance. A less complex model like Kmeans or a too complex one like full VB inference both can result in sub-optimal accuracy, especially in the case of a full VB model, we see a catastrophic decrease of accuracy. Therefore, we should be cautious about the model complexity in order to prevent it from overfitting or falsely estimating some of its parameters. In our considered VB model, we always had in mind a compromise between the expressivity of the general framework and the ability to correctly estimate the introduced parameters (Typically in our case we could face the issue of estimating a $D \times D$ covariance matrix with $D = 512$ or 640 on the basis of only few dozen observations). The obtained trade-off is likely overspecialized to our specific benchmarks, as is illustrated with the diminished gains in accuracy when the number of shots increases. Yet in the extreme case of 1-shot, where the uncertainty is maximum, the proposed combination of VB and PLDA achieves the state-of-the-art performance, suggesting the balance between complexity of the model and ability to estimate its parameters [BK10; Aka98; KW13] is close to optimal.

5.3.3 Performance on cross domain

As we can see, the cluster estimations in our proposed method are dependent on the base dataset, therefore resulting in different levels of accuracy increase on different benchmarks. Although BAVARDAGE has shown promising results on both coarse-grained (e.g. *tiered*-Imagenet and FC100) and fine-grained (e.g. CUB) benchmarks, there remains questions about the performance on cross domain where the base dataset has a complete different distribution with respect to the novel dataset. Therefore here in Table 2 we test the performance of our proposed method in the *mini*-to-CUB cross-domain setting where features of CUB are extracted from a backbone trained with *mini*-Imagenet. Here in our case we thus perform BAVARDAGE based on $\mathbf{\Phi}_w$ being the within-class scatter matrix of *mini*-Imagenet as well:

Table 2: Performance of the proposed BAVARDAGE on cross domain. Here we use the base dataset of *mini*-Imagenet to test out the performance on the novel dataset of CUB, accuracy is obtained with ResNet18 and WideResNet28_10 backbones from [Vei+21].

mini $\rightarrow$ **CUB**		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
NCM	RN18 [Vei+21]	46.27	66.18	46.28	66.09
BAVARDAGE (ours)		**53.02**	**70.01**	**54.63**	**71.45**
NCM	WRN [Vei+21]	48.54	66.26	48.47	68.16
BAVARDAGE (ours)		**56.60**	**74.02**	**58.13**	**75.41**

in which we still observe relative large increase of accuracy. In our opinion, the reason that the proposed method works in cross domain may be that a well pretrained model, regardless of the base dataset, could be a decent representative

for clusters consisting of novel scarce data. An interesting subject for the further research could be to analyse the impact of base dataset on the performance [SCA20; YLX21], and how to choose or design a base set that maximizes the boost in accuracy when evaluating with test data.

5.3.4 Further improvement with graph preprocessing

In [HGP21a] we proposed to integrate graph into feature preprocessing on the test set and obtain relatively large increase in accuracy compared with baseline inductive methods that do not use unlabeled samples. However, as more transductive approaches are put forth with sophisticated classifiers in use of the query set, the effect of graph becomes more incremental and delicate. For methods based on clustering under Gaussian assumption, the use of graph on the test set as preprocessing could be beneficial if features are more gaussian-like in terms of their distributions, which would facilitate the cluster estimations based on such assumption.

Therefore, in this experiment we test the effect of graph preprocessing on top of our proposed method BAVARDAGE, with the reason being: 1) BAVARDAGE is our latest contribution in tackling transductive FSC in a realistic unbalanced setting scenario, and 2) given that the proposed method projects the test set data to supposedly have a shared identity covariance matrix for clusters, the actual effect of graph only lies in the prototype (centroid) estimations for these clusters. In other words, the potential utility of graph preprocessing is to help align features for a better cluster prototype estimation. In terms of implementation, we use the same graph as presented in [HGP21a] and add it on top of the normalized test data as an extra process before projecting them onto the cluster centroids space. And we show the results of our proposed algorithm BAVARDAGE in combination of graph (denoted as "G+BAVARDAGE") in Table 3. Note that here we compare its performance with the original BAVARDAGE without graph preprocessing based on the same pretrained ResNet12 model [Ben+22b] and the accuracies are computed on multiple few-shot benchmarks in the unbalanced setting.

From Table 3 we can observe a slight increase of performance across the board for both 1-shot and 5-shot scenarios, proving the utility of graph in ameliorating the cluster estimations. In addition, the graph used in this experiment has 3 hyperparameters: 1) k that selects the k-est closest samples for each vertex according to cosine similarities, 2) α that adjusts the effect of the vertexes themselves for feature diffusion and 3) κ that determines the level of indirect influence among the vertexes (more details can be found in [HGP21a]). Therefore, here we also test the accuracy of our proposed G+BAVARDAGE as a function of these hyperparameters on various benchmarks to observe the algorithm's behaviors. Namely, in Fig. 29 and Fig. 30 we show the accuracy on two few-shot datasets (*mini*-Imagenet and FC100) as a function of k ranging from 2 to 50 and α ranging from 0 to 5, while keeping the other hyperparameters in BAVARDAGE fixed. For simplicity we set κ to be 1 for all benchmarks since it generally gives the best results. And the following observations could be drawn from the curves: 1) while the optimal hyperparameters may differ on datasets, the algorithm is generally robust with a non-zero *alpha* and

a k that is less than 30; 2) a too larger k would fuse information from unrelated vertexes that deteriorate the performance; and 3) Depending on the datasets, a zero α tends to change accuracy more drastically, suggesting the importance of self vertex in stabilizing the graph. In summary, applying graph as a preprocessing method on extract features slightly improves the performance on BAVARDAGE in the unbalanced setting. However, despite its simplicity, a graph would add in additional parameters to the algorithm and thus increase its sensitivity.

Table 3: Effect of graph preprocessing on the proposed BAVARDAGE.

***mini*-Imagenet**		**unbalanced**	
Method	Backbone	**1-shot**	**5-shot**
BAVARDAGE	RN12 [Ben+22b]	$77.85 \pm 0.28\%$	$88.02 \pm 0.14\%$
G+BAVARDAGE		$\mathbf{78.62 \pm 0.29}\%$	$\mathbf{88.02 \pm 0.13}\%$
***tiered*-Imagenet**		**unbalanced**	
Method	Backbone	**1-shot**	**5-shot**
BAVARDAGE	RN12 [Ben+22b]	$79.38 \pm 0.29\%$	$88.04 \pm 0.18\%$
G+BAVARDAGE		$\mathbf{80.37 \pm 0.29}\%$	$\mathbf{88.22 \pm 0.17}\%$
CUB		**unbalanced**	
Method	Backbone	**1-shot**	**5-shot**
BAVARDAGE	RN12 [Ben+22b]	$83.12 \pm 0.26\%$	$90.81 \pm 0.12\%$
G+BAVARDAGE		$\mathbf{84.14 \pm 0.24}\%$	$\mathbf{91.97 \pm 0.10}\%$
FC100		**unbalanced**	
Method	Backbone	**1-shot**	**5-shot**
BAVARDAGE	RN12 [Ben+22b]	$52.60 \pm 0.32\%$	$66.35 \pm 0.25\%$
G+BAVARDAGE		$\mathbf{53.49 \pm 0.30}\%$	$\mathbf{66.53 \pm 0.24}\%$
CIFAR-FS		**unbalanced**	
Method	Backbone	**1-shot**	**5-shot**
BAVARDAGE	RN12 [Ben+22b]	$82.68 \pm 0.27\%$	$88.97 \pm 0.18\%$
G+BAVARDAGE		$\mathbf{83.02 \pm 0.27}\%$	$\mathbf{89.06 \pm 0.18}\%$

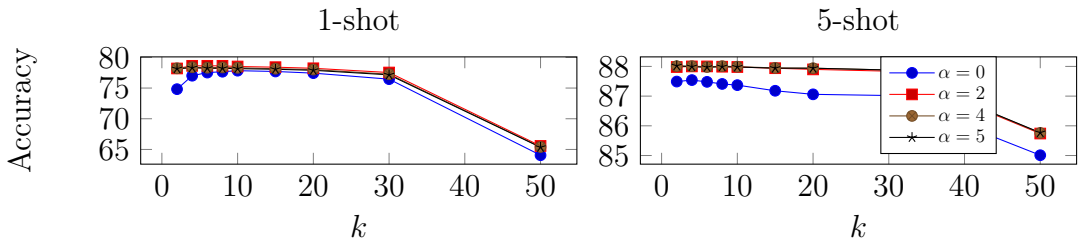


Figure 29: 1-shot and 5-shot accuracy on *mini*-Imagenet as a function of graph hyperparameters (setting: unbalanced).

5.3.5 Limitations and perspectives

Problematically, the proposed solutions comes with a certain number of hyperparameters, some of which are hard to tune without access to a set of validation proxy

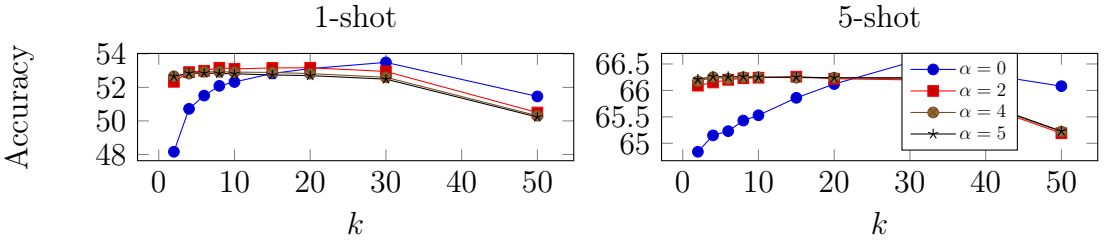


Figure 30: 1-shot and 5-shot accuracy on *mini*-Imagenet as a function of graph hyperparameters (setting: unbalanced).

few-shot tasks. This recurrent problem of gaining a few percentage in accuracy at the cost of adding hyperparameters could be at the heart of more discussions in the field, as it is more problematic than with standard classification where validation sets can overcome the tuning of these hyperparameters. The recent trend towards more diverse evaluation of few-shot classification, notably with the rise of Metadataset [Tri+20], is definitely a step towards the right direction.

In addition, a more well-thought-out covariance matrix could be studied. Since our method only uses base data to mirror cluster covariance for the test data, future work could take into consideration the task-specific information in the few-show task and for example perform a mixture via shrinkage [Bat+20; Bat+22] in which we suggest the estimated $\mathbf{\Phi}_w$ for projection to be the weighted sum of 1) the within-class covariance matrix $\boldsymbol{S}_w^{base}$ computed from base data, and 2) same matrix $\boldsymbol{S}_w^{test}$ computed from test data. The formula can be expressed as follows:

$$\mathbf{\Phi}_w = \alpha \boldsymbol{S}_w^{base} + (1-\alpha)\boldsymbol{S}_w^{test}, \tag{8}$$

where α is a hyperparameter adjusting the weights between two matrix. As the number of labeled samples increases, more weight should be allocated to $\boldsymbol{S}_w^{test}$.

Chapter 6

Conclusions and discussions

6.1 Conclusions

In this thesis we were interested in improving few-shot classification performance in vision problems, a topic that has seen numerous developments in the past few years. Especially, as presented in Chapter 2, we have been considering means to improve the 3 steps of the general pipeline: 1) backbone training, 2) feature preprocessing and 3) classifier design. In that context, we also focused on the use of simultaneous available queries (i.e. transductive setting) which we could benefit from.

In the following subsections, we quickly recap the main contributions.

6.1.1 Feature preprocessing

In terms of feature preprocessing, in [HGP21a] we proposed to construct a graph with similarity measures between the vertices, i.e. feature vectors that represent the data samples. The graph helps filter out the high frequencies in signals and thus aggregate data points that belong to the same class to be more gathered. Therefore, using a Simplified Graph Convolutional (SGC) neural network on the extracted features before applying a logistic regression classifier, we reached state-of-the-art performance in several benchmarks at the time of publication. It is worth noting that, at that time, the proposed method was among the first transductive methods to obtain significant gains compared to inductive ones, with the prediction accuracy boosted up to 12%. Moreover, the fact that we applied graphs on preprocessing as an independent step, as opposed to integrating them into the backbone training such as [Kim+19; SE18; Che+21a] shows more advantages on making use of features information from a specific few-shot task. In Chapter 5 we showed that graphs can improve the prediction accuracy when combined with clustering methods (Section 3). And in [Ham+21] we further built a more sophisticated graph on a class-wise basis, which gives a slight increase in accuracy for 5-shot compared with [HGP21b], suggesting the capacity of well designed graphs for further improvement.

Another feature preprocessing method that we proposed to apply in addressing transductive Few-Shot Classification is Power Transform (PT) [HGP21b; HPG22a]. Different from graph methods, PT is able to adjust the feature distributions for

the benefit of the upcoming classifier. Namely, for a clustering method based on a Gaussian Mixture Model, PT adjusts the feature distributions to be more Gaussian-like so that the assumed Gaussian classifier exerts its optimal utility. Using the same classifier, our proposed PT is shown to be utterly helpful and can bring up to 2% accuracy compared with other preprocessing methods without PT. Additionally, we also observed the benefit of using an ensemble of preprocessing methods, which are generally applied but not often discussed in the literature.

6.1.2 Classifier design

In terms of classifier design, in [HGP21b] we proposed a clustering method under the Expectation Maximization framework and integrated an algorithm based on Optimal Transport for the balanced few-shot setting. Under Gaussian assumptions, our applied Sinkhorn algorithm is able to allocate unlabeled samples into targeted classes with the minimum cost based on the distance metric, and therefore we obtained relatively reliable estimations of the class prototypes in the end. Our proposed method PT+MAP reached 82.92% accuracy on 1-shot and 88.88% accuracy on 5-shot using *mini*-ImageNet, largely surpassing the other alternatives in the same balanced setting at the time. The method has also been shown to bring significant gains on several few-shot benchmarks regardless of pretrained backbones. This proposed method was a long-time top performer method on the competition site Papers With Code [1] [2], it has further been taken by many other works that continue to make slight improvement [LSA21; CVK21]. However, the method has also led to some criticism, especially on the prior on the distribution of the query set that PT+MAP requires [Vei+21].

Moreover, in order to alleviate the dependency on the class-balanced prior in PT+MAP, in [HPG22a] we suggested a modified Sinkhorn algorithm that initializes class distributions to be the minimum number of class affiliations. And we also applied logistic regression on the entire test set, with soft prediction labels coming out of Sinkhorn for the query set for a better prototype estimation. Note that we are among the first to integrate a logistic regression algorithm that makes use of the pseudo labels on the unlabeled samples, and our modified method has brought further increase in accuracy compared with PT+MAP in the balanced setting.

Finally, in [HPG22b] we developed a method tackling the newly proposed unbalanced transductive setting. Namely, we proposed a partial Variational Bayesian model that deems class mixtures and centroids as hidden variables, while adding constrains on the cluster covariance matrix. In the meantime we deployed a Probabilistic Linear Discriminant Analysis to iteratively project the test data into a lower dimension space according to their estimated centroids. Our proposed combination of PLDA and VB largely boosted the performance and reached top accuracy compared to existing state-of-the-art alternatives in the unbalanced setting, especially in the case of 1 shot. The use of base data in estimating the projection matrix of PLDA helps increased accuracy up to 6%.

[1] https://paperswithcode.com/sota/few-shot-image-classification-on-mini-2
[2] https://paperswithcode.com/sota/few-shot-image-classification-on-mini-3

6.1.3 Other contributions

Besides our contributions on feature processing and classifier design, in [Ben+22b] we attempted to improve the prediction accuracy by learning a backbone that generalizes well on the test set. Namely, in terms of backbone training we proposed to use the following ingredients: 1) a rotation classifier that predicts the degree of rotation for input images, and 2) manifold mixup technique that linearly combines features in the hidden layers of a neural network. With these two elements integrated into the training process, our pretrained ResNet12 were able to reach competitive results with a simple NCM classifier. Moreover, we further used data augmentation on each test set image by randomly cropping it into patches and compute the averaged extracted features at the output. Also in this work we applied ensemble methods so that the final feature vector for each sample is the concatenation of 3 backbones pretrained in the same manner. Using a basic Soft-kmeans as the classifier, our proposed method obtained state-of-the-art performance on many few-shot benchmarks (e.g. 71.75% on 1-shot and 87.15% on 5-shot using *mini*-ImageNet).

6.2 Discussions

In this thesis, we have been focusing on improving the FSC performance under a specific set of conditions presented in Chapter 2), i.e. input size $84 \times 84 \times 3$ for the backbone training, no extra data, base/novel split from the same benchmark, etc. These conditions have been considered mainstream and are used by a large amount of works in the field, mainly for comparison purposes. However, there have been new conditions proposed over the course of my thesis [Bat+20; Che+21b; Bat+22].

With those associated evolutions in the domain of few-shot classification, more and more questions have emerged that require detailed investigation. In the coming subsections, we discuss some of them.

6.2.1 Evolutions of the FSC conditions

6.2.1.1 Resolution of the input images

In the standard setting for few-shot classification, the input data resolution is originally fixed at 84×84 pixels for both training and testing. With more and more methods seeking further improvement on the performance, some works emerged that used input data with a much larger resolution to train the model or to extract features from. For instance, in [Che+21b] the authors use 128×128 as input resolution for self-supervised training. And in [Luo+21] the authors apply multi-crop on the input of 224×224 pixels to obtain multiple patches of resolution 84×84 pixels for feature extraction. Although these methods have indeed boosted the prediction accuracy by a relatively large margin, it raises an important question about how much impact the input resolution can have for the performance improvement. Due to the fact that some of these methods also use additional data in combination with larger input images, it is difficult to disentangle and study the effect of each of them on the overall accuracy.

It is not surprising that higher resolution can help in some cases, especially when objects to recognize can be discriminated thanks to small or detailed features. With the advances of GPU capabilities, it is thus possible that current few-shot learning pipelines would increase significantly in performance by considering higher resolution inputs.

6.2.1.2 About the base dataset

In few-shot classification, it is crucial to learn a model that can generalize well to the unseen tasks. This would require not only a good model but also a well designed base dataset that can optimize the gain. However, as the majority of works in the field has been focusing on models, there is a paucity of literature targeting the impact of base dataset. [SCA20] is one of the first papers that shines a light on this aspect, the authors run extended experiments on the class prediction accuracy in relation with multiple factors on the base dataset such as the number of base classes, the number of samples per class and the coarseness of classes. In [Laf+22] the authors further illustrate the impact of base classes on the performance, showing that some classes could be significantly harmful to the performance on the given set of tasks (e.g. 1-shot problems with fixed ways). Similarly, another recent work [Ben+22c] studies the granularity of base classes and its impact on the few-shot performance, the authors suggest that the more granular the base classes are, the less accurate the predictions become.

As analysis started to emerge, further research could focus more on the design of base dataset so that a model is able to fully explore its characteristics.

6.2.1.3 Training with additional data

Besides the characteristics of the base dataset (e.g. class selection, number of samples per class, etc) that would require further studies, some works also use additional information to learn a model. For example in [Xin+19; Sch+19; Zha+21] the authors incorporate the semantic information of the base data into the training step, and the semantic features are learned using large text corpora. Although using additional resources during training can be beneficial, there needs further research about how to best integrate these information with base data.

Moreover, some works train their models with extended large data in addition to the base data, resulting in a large increase of accuracy. Namely, in [Che+21b; Bat+22] the authors firstly learn a model with ImageNet dataset that is much larger than a standard few-shot benchmark such as *mini*-Imagenet, then the model is finetuned with the base dataset of the benchmark. This raises the question about the impact of external data on the performance in the context of few shot, given that in other domains such as NLP, giant models (e.g. GPT3 [Bro+20]) are observed to be well adapted for tasks with limited data.

6.2.1.4 Cross-domain few-shot classification

In the typical settings, experiments on few-shot classification are conducted with the base, novel and validation classes drawn from the same initial dataset. However,

this may not be ideal as these sets typically display similar distributions of data. And yet in many real world applications, it is hard or even impossible to gather large numbers of data for backbone training. For instance in domains such as satellite imagery and cancerology where there are examples of rare categories, the requisition of training data would be too costly or unrealistic.

This leads to a new few-shot setting called "cross domain" where there is a large difference between base set and novel set. Early works such as [Che+19a; Wan+19b; Man+20; HGP21a; Zik+20] propose to train models using the base class set of a few-shot benchmark, and evaluate the performance using the novel class set of another benchmark. Although these works report their results on the defined cross-domain setting, we find only few works that study cross domain in a relatively thorough manner [Guo+20; Oh+22]. Given the vast domain differences between base classes and classes in the novel set, models often suffer from a large drop of accuracy. In addition, it would be difficult to tune hyperparameters with a validation set in cross-domain settings due to the same reason, making it more challenging for a model to reach its optimal condition.

Although methods have been proposed for the cross-domain setting in the case of few shot, this remains a challenging task due to the tremendously decreased accuracy caused by domain shift (e.g. 63.90%/79.15% 1/5-shot accuracy [HSS18] in *mini*-to-CUB cross-domain setting, compared with 91.91%/94.62% accuracy with CUB all alone). While some early works only perform their proposed methods in one particular *mini*-to-CUB situation, there needs to be experiments conducted on a broarder range of scenarios with various test datasets in different domains (e.g. [Guo+20]) to further evaluate the ability of a pretrained backbone for its feature generalization.

More recently, a newly proposed Meta-dataset[3] is used to conduct experiments in works such as [Bai+20; DGZ20; Bat+20; Bat+22]. Similar to standard few-shot benchmarks, Meta-dataset is also split into base, validation and novel class set. But each split in Meta-dataset is a combination of several datasets with different image sizes and class categories, which can further stress the proposed method capacity to leverage different training sources for improving their generalization [Tri+20]. Meta-dataset contains a collection of 10 datasets from different domains, representing a diverse data distribution:

- ILSVRC-2012 [Rus+15] (the ImageNet dataset, consisting of natural images with 1000 categories),
- Omniglot [LST15] (hand-written characters, 1623 classes),
- Aircraft [Maj+13] (dataset of aircraft images, 100 classes),
- CUB [Wah+11] (dataset of Birds, 200 classes),
- Describable Textures [Cim+14] (different kinds of texture images with 43 categories),
- Quick Draw [Jon+16] (black and white sketches of 345 different categories),

[3] `https://github.com/google-research/meta-dataset`

- Fungi [SC18] (a large dataset of mushrooms with 1500 categories),
- VGG Flower [NZ08] (dataset of flower images with 102 categories),
- Traffic Signs [Hou+13] (German traffic sign images with 43 classes),
- MSCOCO [Lin+14] (images collected from Flickr, 80 classes).

All datasets except Traffic Signs and MSCOCO have a base-validation-novel set split (proportioned roughly into 70%, 15%, 15%). The Traffic Signs and MSCOCO datasets are reserved for evaluation only [Tri+20].

Given that the base and novel classes in Meta-dataset consist of classes in several datasets of various domains, the setting of this benchmark can be considered as cross-domain. Additionally, not only the Meta-dataset opts in the addition of extra data for backbone training, the evaluation used in this dataset includes various shots and ways along with balanced and unbalanced settings. And it is gaining momentum in few-shot classification [Tri+20; Req+19; Bai+20; DGZ20; LLB22; Dum+21]. Considering that it operates under conditions that are close to the real world scenarios, this is the right direction for future works to explore and address the challenges related to this setting.

6.2.2 Solutions for further improvement

6.2.2.1 Self-Supervised Learning

Besides feature preprocessing and classifier design, another important step in the few-shot classification pipeline is the backbone training. With the development of research in related areas, there are techniques such as Self-Supervised Learning (SSL) [GSK18b; Ale+15; DGE15; ZIE16; Car+18; XGF16] that are proposed to improve the feature generalization. In few-shot literature, they are often used as an auxiliary training task to regularize the input data. Together with Supervised Learning (SL) on the training set, SSL techniques are observed to be effective in increasing the prediction accuracy [Man+20; Riz+21].

However, SSL as un unsupervised technique was hardly used alone in few-shot classification, and its impact on the cross domain is still yet to be explored. Few works study on the setting where the labels of the training data are not accessible, suggesting a complete unsupervised scenario for backbone training. In [CMLM21] and [Liu+21b] the authors apply the same contrastive learning methods as [He+20; Che+20] and obtain competitive accuracy compared to methods trained with labeled data.

More recently, SSL methods are shown to be applied as an adaptation technique to better estimate the domain of the test set. In [PH21; Isl+21] a novel setting has been suggested in which we additionally possess a certain number of unlabeled novel set data during training in order to address the challenge of domain differences. This newly proposed setting allows further possibilities on the research concerning cross domain and the effect of SSL. For instance, in [Oh+22] the authors suggest the effectiveness of SSL compared with SL in the case when the domain similarity is smaller or the few-shot difficulty is lower. In the same paper the authors also

propose a two-stage training strategy that firstly pretrains the backbone with labeled data in the training set, followed by finetuning the pretrained model with auxiliary data from the novel set. According to the paper, the two-stage training strategy stands out as the one that has the best results on the novel setting.

In summary, more studies need to be conducted on the use of SSL and its impact on various scenarios such as 1) no label for the training data; 2) training data plus additional unlabeled data from the target domain are available for training.

6.2.2.2 Domain adaptation

Since more and more studies in the field have been focusing on the cross domain that is presented above, one of the most common approaches is to apply domain adaptation.

There are a few works that use domain adaptation to address the domain difference between the training and the test set. Usually methods are proposed to have an adaptation layer attached to the feature extractor during training, so that the additional part of the architecture could help learn more task-specific information on the test set. For instance, in [Tse+20] the authors add a feature-wise transformation layer to simulate feature distributions in various domains. And in [Bat+20] the authors propose to add FiLM layers [Per+18] on top of a ResNet, the FiLM layers produce scale and shift parameters for the extracted features to adapt on a task by task basis, and these parameters are generated using the support set data.

As we observe, more research can be conducted for the purpose of 1) Analysing and quantifying the domain differences differences between the base and the test datasets; and 2) reducing these differences using domain adaptation methods.

6.2.3 Further future directions of research

6.2.3.1 Learning features of targeted class

In few-shot classification, the trained feature extractor has the ability of characterising distinct features of a class, which are further used for classification on the new tasks. However, there exist scenarios where a backbone might extract the wrong features. For example, for an input image that contains a photo of a cat, or a painting of a vase, it would be difficult to know whether a trained backbone extracts the features that discriminate a photo from a picture or those detecting an animal.

The above example can be categorized into the problematic of ambiguity in classification, an input image could be ambiguous due to the fact that it contains other non-targeted objects along with the targeted ones. In this scenario, we often need prior knowledge or contexts so that the model extracts features of the targeted class. However, it is difficult in few shot due to limited data. Especially in the case of 1-shot classification, undesirably extracted features could have more negative impact on the performance since there is only one labeled data and no other reference on the targeted class.

Therefore, methods are required so that the backbone produces features of the targeted class. One possible solution would be to use multimodel [Ngi+11; VPJ17; ASL16] and integrate semantic information as a prior into the training process [Xin+19]. Other solutions include 1) disentanglement [Ben13; Loc+19] that attempts to learn compact and independent factors of the data; 2) using attention-based models to help select the right features [Bat+20; Ye+20; Zha+22] and so on. However, more experiments need to be conducted for a full analysis about how to construct the prior knowledge and when to use it.

6.2.3.2 Few-shot classification with multiple objects in an input

In the context of few shot, especially 1-shot classification, working with one example is particularly challenging as there might be multiple objects in a scene, which can make it ambiguous even for the best trained model [Mor+21].

Authors in [Ben+22a] firstly identify the problem of having multiple objects in an image and propose a methodology to disambiguate them using data augmentation and an optimization process. Namely, the authors model the distribution formed by the features of different regions of a scene as a simplex. This modeling allows to extract different feature representations that can be identified with different objects in an image. Then, these representations are exploited to improve the performance on classification tasks in the one-shot classification setting.

The obtained results are encouraging, and it opens the door for further improvement when considering problems with multiple class-labeled data, as a similar approach should better identify the common object between multiple examples of the same class.

6.2.3.3 Classification on a single few-shot task

In the standard few-shot settings, a model is evaluated with hundreds and thousands of few-shot tasks and the final accuracy is the averaged performance of all. However, those methods often have various results depending on the tasks. Therefore, they may produce optimal accuracy on average, but suboptimal performance for a specific task. Moreover, in certain cases in the real world we only need the model to perform well on one single particular task, we thus need methods that are able to target specifically the given task, such as integrating it with base data during training so that the model takes the task-specific information into account; or autonomously optimizing the hyper-parameters for that task.

Note that with the necessity of task-specific information in this particular situation, a well-adapted training paradigm is required to better extract such information. This brings us back to the comparison between meta learning and transfer learning. We believe that although methods using meta learning paradigm tend to show inferior performance in standard few-shot settings compared to transfer learning, their potential is yet to be explored when tackling a specific task.

6.2.3.4 Active few-shot classification

In transductive few-shot classification, the error mainly comes from the following two sources: 1) false clustering of unlabeled samples; 2) incorrect class prototypes. The second type of error occurs when the selected labeled samples in the support set turn out to be the outliers that are closer to the prototype of another class [BI17]. Therefore, solutions [BI17; PZS20; Mül+22; Li+22; Abd+22] have been proposed using active learning which interacts with users in order to adjust the prototypes. The goal is for the user to select samples based on their predicted labels and consider them as labeled for the targeted class, so that there are additional labeled samples to re-evaluate its prototype. With few-shot learning being applied more and more in real world situations, active learning or methods that allow different selections of samples can be a very interesting research subject in the field that helps boost the performance in the context of little data.

Bibliography

[Abd+22] Aymane Abdali, Vincent Gripon, Lucas Drumetz, and Bartosz Boguslawski. *Active Few-Shot Classification: a New Paradigm for Data-Scarce Learning Settings*. 2022. DOI: 10.48550/ARXIV.2209.11481. URL: https://arxiv.org/abs/2209.11481.

[Aga18] Abien Fred Agarap. "Deep learning using rectified linear units (relu)". In: *arXiv preprint arXiv:1803.08375* (2018).

[AK00] Franz Aurenhammer and Rolf Klein. "Voronoi Diagrams." In: *Handbook of computational geometry* 5.10 (2000), pp. 201–290.

[Aka98] Hirotogu Akaike. "Information theory and an extension of the maximum likelihood principle". In: *Selected papers of hirotugu akaike*. Springer, 1998, pp. 199–213.

[Ale+15] Dosovitskiy Alexey, Philipp Fischer, Jost Tobias, Martin Riedmiller Springenberg, and Thomas Brox. "Discriminative unsupervised feature learning with exemplar convolutional neural networks". In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 99 (2015).

[ASL16] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Semantic segmentation of earth observation data using multimodal and multi-scale deep networks". In: *Asian conference on computer vision*. Springer. 2016, pp. 180–196.

[Bai+20] Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. "Meta-learning with adaptive hyperparameters". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 20755–20765.

[Bat+20] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. "Improved few-shot visual classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 14493–14502.

[Bat+22] Peyman Bateni, Jarred Barber, Jan-Willem van de Meent, and Frank Wood. "Enhancing few-shot image classification with unlabelled examples". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 2796–2805.

[Ben12] Yoshua Bengio. "Practical recommendations for gradient-based training of deep architectures". In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 437–478.

[Ben13] Yoshua Bengio. “Deep learning of representations: Looking forward”. In: *International conference on statistical language and speech processing*. Springer. 2013, pp. 1–37.

[Ben+22a] Yassir Bendou, Lucas Drumetz, Vincent Gripon, Giulia Lioi, and Bastien Pasdeloup. “Le manchot, la banane et la bibliothèque...(de la désambiguïsation d’une tâche de classification avec un exemple)”. In: *GRETSI 2022*. 2022.

[Ben+22b] Yassir Bendou, Yuqing Hu, Raphael Lafargue, Giulia Lioi, Bastien Pasdeloup, Stéphane Pateux, and Vincent Gripon. “Easy—Ensemble Augmented-Shot-Y-Shaped Learning: State-of-the-Art Few-Shot Classification with Simple Components”. In: *Journal of Imaging* 8.7 (2022). ISSN: 2313-433X. DOI: 10.3390/jimaging8070179. URL: https://www.mdpi.com/2313-433X/8/7/179.

[Ben+22c] Etienne Bennequin, Myriam Tami, Antoine Toubhans, and Céline Hudelot. “Few-Shot Image Classification Benchmarks are Too Far From Reality: Build Back Better with Semantic Task Sampling”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4767–4776.

[Ber+19] Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. “Meta-learning with differentiable closed-form solvers”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=HyxnZh0ct7.

[BI17] Rinu Boney and Alexander Ilin. “Semi-supervised and active few-shot learning with prototypical networks”. In: *arXiv preprint arXiv:1711.10856* (2017).

[BK10] Harish S Bhat and Nitesh Kumar. “On the derivation of the bayesian information criterion”. In: *School of Natural Sciences, University of California* 99 (2010).

[BLH21] Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. “Deep learning for AI”. In: *Communications of the ACM* 64.7 (2021), pp. 58–65.

[BMRG17] Federico Baldassarre, Diego González Morín, and Lucas Rodés-Guirao. “Deep koalarization: Image colorization using cnns and inception-resnet-v2”. In: *arXiv preprint arXiv:1712.03400* (2017).

[BN06] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[BNH19] Tal Ben-Nun and Torsten Hoefler. “Demystifying parallel and distributed deep learning: An in-depth concurrency analysis”. In: *ACM Computing Surveys (CSUR)* 52.4 (2019), pp. 1–43.

[Bon+21] Myriam Bontonou, Giulia Lioi, Nicolas Farrugia, and Vincent Gripon. “Few-Shot Decoding of Brain Activation Maps”. In: *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE. 2021, pp. 1326–1330.

[Bot10] Léon Bottou. "Large-scale machine learning with stochastic gradient descent". In: *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[Bou+20a] Malik Boudiaf, Imtiaz Ziko, Jérôme Rony, Jose Dolz, Pablo Piantanida, and Ismail Ben Ayed. "Information Maximization for Few-Shot Learning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 2445–2457. URL: https://proceedings.neurips.cc/paper/2020/file/196f5641aa9dc87067da4ff90fd81e7b-Paper.pdf.

[Bou+20b] Malik Boudiaf, Imtiaz Masud Ziko, Jérôme Rony, José Dolz, Pablo Piantanida, and Ismail Ben Ayed. "Transductive Information Maximization For Few-Shot Learning". In: *CoRR* abs/2008.11297 (2020). arXiv: 2008.11297. URL: https://arxiv.org/abs/2008.11297.

[Bri89] John Bridle. "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters". In: *Advances in neural information processing systems* 2 (1989).

[Bro+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[Bro+93] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. "Signature verification using a" siamese" time delay neural network". In: *Advances in neural information processing systems* 6 (1993).

[Car+18] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. "Deep clustering for unsupervised learning of visual features". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 132–149.

[CB01] Adrian Corduneanu and Christopher M Bishop. "Variational Bayesian model selection for mixture distributions". In: *Artificial intelligence and Statistics*. Vol. 2001. Morgan Kaufmann Waltham, MA. 2001, pp. 27–34.

[Che+19a] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. "A Closer Look at Few-shot Classification". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=HkxLXnAcFQ.

[Che+19b] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. "Image deformation meta-networks for one-shot learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8680–8689.

[Che+20] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. "Improved Baselines with Momentum Contrastive Learning". In: *CoRR* abs/2003.04297 (2020). arXiv: 2003.04297. URL: https://arxiv.org/abs/2003.04297.

[Che+21a] Cen Chen, Kenli Li, Wei Wei, Joey Tianyi Zhou, and Zeng Zeng. "Hierarchical graph neural networks for few-shot learning". In: *IEEE Transactions on Circuits and Systems for Video Technology* 32.1 (2021), pp. 240–252.

[Che+21b] Da Chen, Yuefeng Chen, Yuhong Li, Feng Mao, Yuan He, and Hui Xue. "Self-supervised learning for few-shot image classification". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 1745–1749.

[Che95] Yizong Cheng. "Mean shift, mode seeking, and clustering". In: *IEEE transactions on pattern analysis and machine intelligence* 17.8 (1995), pp. 790–799.

[Cim+14] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. "Describing textures in the wild". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 3606–3613.

[CMLM21] Zitian Chen, Subhransu Maji, and Erik Learned-Miller. "Shot in the dark: Few-shot learning with no base-class labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2668–2677.

[Cox58] David R Cox. "The regression analysis of binary sequences". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 20.2 (1958), pp. 215–232.

[Cra02] Jan Salomon Cramer. "The origins of logistic regression". In: (2002).

[CSZ09] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 542–542.

[CV95] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[CVK21] Tomáš Chobola, Daniel Vašata, and Pavel Kordík. "Transfer learning based few-shot classification using optimal transport mapping from preprocessed latent space of backbone neural network". In: *AAAI Workshop on Meta-Learning and MetaDL Challenge*. PMLR. 2021, pp. 29–37.

[CVS15] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. "Approximate fisher kernels of non-iid image models for image categorization". In: *IEEE transactions on pattern analysis and machine intelligence* 38.6 (2015), pp. 1084–1098.

[Dai+09] Wenyuan Dai, Ou Jin, Gui-Rong Xue, Qiang Yang, and Yong Yu. "Eigentransfer: a unified framework for transfer learning". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009, pp. 193–200.

[Der05] Konstantinos G Derpanis. "Mean shift clustering". In: *Lecture Notes* 32 (2005).

[Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[DGE15] Carl Doersch, Abhinav Gupta, and Alexei A Efros. "Unsupervised visual representation learning by context prediction". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1422–1430.

[DGZ20] Carl Doersch, Ankush Gupta, and Andrew Zisserman. "Crosstransformers: spatially-aware few-shot transfer". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21981–21993.

[Dhi+20] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. "A Baseline for Few-Shot Image Classification". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: `https://openreview.net/forum?id=rylXBkrYDS`.

[Dos+14] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. "Discriminative unsupervised feature learning with convolutional neural networks". In: *Advances in neural information processing systems* 27 (2014).

[DSM19] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. "Diversity with cooperation: Ensemble methods for few-shot classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 3723–3731.

[Dum+21] Vincent Dumoulin, Neil Houlsby, Utku Evci, Xiaohua Zhai, Ross Goroshin, Sylvain Gelly, and Hugo Larochelle. "A unified few-shot classification benchmark to compare transfer and meta learning approaches". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*. 2021.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1126–1135.

[FR12] Charles W Fox and Stephen J Roberts. "A tutorial on variational Bayesian inference". In: *Artificial intelligence review* 38.2 (2012), pp. 85–95.

[Gem+17] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. "Audio set: An ontology and human-labeled dataset for audio events". In: *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2017, pp. 776–780.

[GK19] Spyros Gidaris and Nikos Komodakis. "Generating classification weights with gnn denoising autoencoders for few-shot learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 21–30.

[GMS05] Marco Gori, Gabriele Monfardini, and Franco Scarselli. "A new model for learning in graph domains". In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 2. IEEE. 2005, pp. 729–734.

[GSK18a] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. "Unsupervised Representation Learning by Predicting Image Rotations". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: `https://openreview.net/forum?id=S1v4N2l0-`.

[GSK18b] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. "Unsupervised representation learning by predicting image rotations". In: *arXiv preprint arXiv:1803.07728* (2018).

[Guo+20] Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. "A broader study of cross-domain few-shot learning". In: *European conference on computer vision*. Springer. 2020, pp. 124–141.

[Ham+21] Mounia Hamidouche, Carlos Lassance, Yuqing Hu, Lucas Drumetz, Bastien Pasdeloup, and Vincent Gripon. "Improving Classification Accuracy With Graph Filtering". In: *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2021, pp. 334–338.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[He+20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[HG08] Shaobo Hou and Aphrodite Galata. "Robust estimation of Gaussian mixtures from noisy input data". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.

[HGP21a] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. "Graph-based interpolation of feature vectors for accurate few-shot classification". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 8164–8171.

[HGP21b] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. "Leveraging the feature distribution in transfer-based few-shot learning". In: *International Conference on Artificial Neural Networks*. Springer. 2021, pp. 487–499.

[HLLJ19] Gabriel Huang, Hugo Larochelle, and Simon Lacoste-Julien. "Are Few-shot Learning Benchmarks Too Simple?" In: (2019).

[HM95] Jun Han and Claudio Moraga. "The influence of the sigmoid function parameters on the speed of backpropagation learning". In: *International workshop on artificial neural networks*. Springer. 1995, pp. 195–201.

[Hou+13] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark". In: *The 2013 international joint conference on neural networks (IJCNN)*. Ieee. 2013, pp. 1–8.

[HPG22a] Yuqing Hu, Stéphane Pateux, and Vincent Gripon. "Squeezing backbone feature distributions to the max for efficient few-shot learning". In: *Algorithms* 15.5 (2022), p. 147.

[HPG22b] Yuqing Hu, Stéphane Pateux, and Vincent Gripon. *Adaptive Dimension Reduction and Variational Inference for Transductive Few-Shot Classification*. 2022. DOI: 10.48550/ARXIV.2209.08527. URL: https://arxiv.org/abs/2209.08527.

[HSS08] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. "Kernel methods in machine learning". In: *The annals of statistics* 36.3 (2008), pp. 1171–1220.

[HSS18] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

[Hu+22] Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M Hospedales. "Pushing the Limits of Simple Pipelines for Few-Shot Learning: External Data and Fine-Tuning Make a Difference". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9068–9077.

[HW79] John A Hartigan and Manchek A Wong. "Algorithm AS 136: A k-means clustering algorithm". In: *Journal of the royal statistical society. series c (applied statistics)* 28.1 (1979), pp. 100–108.

[Ian+16] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size". In: *arXiv preprint arXiv:1602.07360* (2016).

[IS15] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[Isl+21] Ashraful Islam, Chun-Fu Richard Chen, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, and Richard J Radke. "Dynamic distillation network for cross-domain few-shot recognition with unlabeled data". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 3584–3595.

[Jon+16] Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. "The quick, draw!-ai experiment". In: *Mount View, CA, accessed Feb* 17.2018 (2016), p. 4.

[Jum+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6980.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. *Learning multiple layers of features from tiny images*. Tech. rep. Citeseer, 2009.

[Kho+20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. "Supervised Contrastive Learning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 18661–18673. URL: https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

[Kim+19] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. "Edge-Labeling Graph Neural Network for Few-shot Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11–20.

[KSH17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[KW13] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[KZS15] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition". In: *ICML deep learning workshop*. Vol. 2. 2015.

[Laf+22] Raphael Lafargue, Jean-Philippe Diguet, Vincent Gripon, and Bastien Pasdeloup. "Classes adversaires dans l'apprentissage avec peu d'exemples". In: *GRETSI*. 2022.

[LAP21] Yann Lifchitz, Yannis Avrithis, and Sylvaine Picard. "Local propagation for few-shot learning". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 10457–10464.

[LB+95] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[Lee+19] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. "Meta-learning with differentiable convex optimization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10657–10665.

[LH16] Ilya Loshchilov and Frank Hutter. "Sgdr: Stochastic gradient descent with warm restarts". In: *arXiv preprint arXiv:1608.03983* (2016).

[Li+19] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. "Finding Task-Relevant Features for Few-Shot Learning by Category Traversal". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1–10.

[Li+22] Xiaorun Li, Zeyu Cao, Liaoying Zhao, and Jianfeng Jiang. "ALPN: Active-Learning-Based Prototypical Network for Few-Shot Hyperspectral Imagery Classification". In: *IEEE Geoscience and Remote Sensing Letters* 19 (2022), pp. 1–5. DOI: 10.1109/LGRS.2021.3101495.

[Lic+20] Moshe Lichtenstein, Prasanna Sattigeri, Rogerio Feris, Raja Giryes, and Leonid Karlinsky. "Tafssl: Task-adaptive feature sub-space learning for few-shot classification". In: *European Conference on Computer Vision*. Springer. 2020, pp. 522–539.

[Lin+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[Liu+18] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, and Yi Yang. "Transductive propagation network for few-shot learning". In: (2018).

[Liu+19] 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: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=SyVuRiC5K7.

[Liu+21a] Chen Liu, Yanwei Fu, Chengming Xu, Siqian Yang, Jilin Li, Chengjie Wang, and Li Zhang. "Learning a Few-shot Embedding Model with Contrastive Learning". In: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 8635–8643. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17047.

[Liu+21b] Chen Liu, Yanwei Fu, Chengming Xu, Siqian Yang, Jilin Li, Chengjie Wang, and Li Zhang. "Learning a few-shot embedding model with contrastive learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 10. 2021, pp. 8635–8643.

[LLB22] Wei-Hong Li, Xialei Liu, and Hakan Bilen. "Cross-domain Few-shot Learning with Task-specific Adapters". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7161–7170.

[Loc+19] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. "Challenging common assumptions in the unsupervised learning of disentangled representations". In: *international conference on machine learning*. PMLR. 2019, pp. 4114–4124.

[LSA21] Michalis Lazarou, Tania Stathaki, and Yannis Avrithis. "Iterative label cleaning for transductive and semi-supervised few-shot learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8751–8760.

[LSQ20] Jinlu Liu, Liang Song, and Yongqiang Qin. "Prototype rectification for few-shot learning". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer. 2020, pp. 741–756.

[LST15] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. "Human-level concept learning through probabilistic program induction". In: *Science* 350.6266 (2015), pp. 1332–1338.

[Luo+21] Xu Luo, Longhui Wei, Liangjian Wen, Jinrong Yang, Lingxi Xie, Zenglin Xu, and Qi Tian. "Rectifying the Shortcut Learning of Background for Few-Shot Learning". In: *Advances in Neural Information Processing Systems* 34 (2021).

[LW21] Sihan Liu and Yue Wang. "Few-shot Learning with Online Self-Distillation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1067–1070.

[Ma+21] Jiawei Ma, Hanchen Xie, Guangxing Han, Shih-Fu Chang, Aram Galstyan, and Wael Abd-Almageed. "Partner-assisted learning for few-shot image classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10573–10582.

[Maj+13] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. "Fine-Grained Visual Classification of Aircraft". In: *CoRR* abs/1306.5151 (2013). arXiv: 1306.5151. URL: http://arxiv.org/abs/1306.5151.

[Mal16] Stéphane Mallat. "Understanding deep convolutional networks". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150203.

[Man+20] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. "Charting the right manifold: Manifold mixup for few-shot learning". In: *The IEEE Winter Conference on Applications of Computer Vision*. 2020, pp. 2218–2227.

[Mar+08] André C Marreiros, Jean Daunizeau, Stefan J Kiebel, and Karl J Friston. "Population dynamics: variance and the sigmoid activation function". In: *Neuroimage* 42.1 (2008), pp. 147–157.

[MHN+13] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. icml*. Vol. 30. 1. Citeseer. 2013, p. 3.

[MHV16] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. "TUT Database for Acoustic Scene Classification and Sound Event Detection". In: *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*. Budapest, Hungary, 2016.

[Mor+21] V Morfi, D Stowell, V Lostanlen, A Strandburg-Peshkin, L Gill, H Pamula, D Benvent, I Nolasco, S Singh, S Sridhar, et al. *DCASE 2021 Task 5: Few-shot Bioacoustic Event Detection Development Set*. 2021.

[MPV21] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[MSN21] Pratik Mazumder, Pravendra Singh, and Vinay P Namboodiri. "Improving few-shot learning using composite rotation based auxiliary task". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 2654–2663.

[Mül+22] Thomas Müller, Guillermo Pérez-Torró, Angelo Basile, and Marc Franco-Salvador. "Active Few-Shot Learning with FASL". In: *Natural Language Processing and Information Systems - 27th International Conference on Applications of Natural Language to Information Systems, NLDB 2022, Valencia, Spain, June 15-17, 2022, Proceedings*. Ed. by Paolo Rosso, Valerio Basile, Raquel Martínez, Elisabeth Métais, and Farid Meziane. Vol. 13286. Lecture Notes in Computer Science. Springer, 2022, pp. 98–110. DOI: 10.1007/978-3-031-08473-7_9. URL: https://doi.org/10.1007/978-3-031-08473-7_9.

[MVDGB08] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. "The group lasso for logistic regression". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.1 (2008), pp. 53–71.

[Nar97] Sridhar Narayan. "The generalized sigmoid activation function: Competitive supervised learning". In: *Information sciences* 99.1-2 (1997), pp. 69–82.

[Ngi+11] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. "Multimodal deep learning". In: *ICML*. 2011.

[NZ08] Maria-Elena Nilsback and Andrew Zisserman. "Automated flower classification over a large number of classes". In: *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE. 2008, pp. 722–729.

[Oh+22] Jaehoon Oh, Sungnyun Kim, Namgyu Ho, Jin-Hwa Kim, Hwanjun Song, and Se-Young Yun. "Understanding Cross-Domain Few-Shot Learning: An Experimental Study". In: *CoRR* abs/2202.01339 (2022). arXiv: 2202.01339. URL: https://arxiv.org/abs/2202.01339.

[OHT21] Yassine Ouali, Céline Hudelot, and Myriam Tami. "Spatial Contrastive Learning for Few-Shot Classification". In: *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part I*. Ed. by Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and José Antonio Lozano. Vol. 12975. Lecture Notes in Computer Science. Springer, 2021, pp. 671–686. DOI: 10.1007/978-3-030-86486-6_41. URL: https://doi.org/10.1007/978-3-030-86486-6_41.

[ORLL18] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. "Tadam: Task dependent adaptive metric for improved few-shot learning". In: *Advances in neural information processing systems* 31 (2018).

[Ort+21] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. "Multi-objective interpolation training for robustness to label noise". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 6606–6615.

[Per+18] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. "Film: Visual reasoning with a general conditioning layer". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[PH21] Cheng Perng Phoo and Bharath Hariharan. "Self-training For Few-shot Transfer Across Extreme Task Differences". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=O3Y56aqpChA.

[Pic+22] Lukáš Picek, Milan Šulc, Jiří Matas, Thomas S Jeppesen, Jacob Heilmann-Clausen, Thomas Læssøe, and Tobias Frøslev. "Danish fungi 2020-not just another image recognition dataset". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 1525–1535.

[PY09] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[PZS20] Pouya Pezeshkpour, Zhengli Zhao, and Sameer Singh. "On the utility of active instance selection for few-shot learning". In: *NeurIPS HAMLETS* (2020).

[Rad+18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. "Improving language understanding by generative pre-training". In: (2018).

[Rad+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[Ren+18] 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". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=HJcSzz-CZ.

[Req+19] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. "Fast and flexible multi-task classification using conditional neural adaptive processes". In: *Advances in Neural Information Processing Systems* 32 (2019).

[Riz+21] Mamshad Nayeem Rizve, Salman H. Khan, Fahad Shahbaz Khan, and Mubarak Shah. "Exploring Complementary Strengths of Invariant and Equivariant Representations for Few-Shot Learning". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 10836–10846. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Rizve_Exploring_Complementary_Strengths_of_Invariant_and_Equivariant_Representations_for_Few-Shot_CVPR_2021_paper.html.

[RL17] Sachin Ravi and Hugo Larochelle. "Optimization as a Model for Few-Shot Learning". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=rJY0-Kcll.

[Rod+20] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. "Embedding propagation: Smoother manifold for few-shot classification". In: *European Conference on Computer Vision*. Springer. 2020, pp. 121–138.

[Ros20] Corby Rosset. "Turing-NLG: A 17-billion-parameter language model by Microsoft". In: *Microsoft Blog* 1.2 (2020).

[Rus+15] 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". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[Rus+19] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. "Meta-Learning with Latent Embedding Optimization". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=BJgklhAcK7.

[SC18] Brigit Schroeder and Yin Cui. "Fgvcx fungi classification challenge 2018". In: *Available online: github. com/visipedia/fgvcx_fungi_ comp (accessed on 14 July 2021)* (2018).

[SCA20] Othman Sbai, Camille Couprie, and Mathieu Aubry. "Impact of base dataset design on few-shot image classification". In: *European Conference on Computer Vision*. Springer. 2020, pp. 597–613.

[Sch+19] Eli Schwartz, Leonid Karlinsky, Rogerio Feris, Raja Giryes, and Alex M Bronstein. "Baby steps towards few-shot learning with multiple semantics". In: *arXiv preprint arXiv:1906.01905* (2019).

[SE18] Victor Garcia Satorras and Joan Bruna Estrach. "Few-Shot Learning with Graph Neural Networks". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=BJj6qGbRW.

[Sen+13] Andrew Senior, Georg Heigold, Marc'Aurelio Ranzato, and Ke Yang. "An empirical study of learning rates in deep neural networks for speech recognition". In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE. 2013, pp. 6724–6728.

[Sil+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), pp. 484–489.

[Sil+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. "Mastering the game of go without human knowledge". In: *nature* 550.7676 (2017), pp. 354–359.

[SL12] George AF Seber and Alan J Lee. *Linear regression analysis*. John Wiley & Sons, 2012.

[SS97] Jorge Sola and Joaquin Sevilla. "Importance of input data normalization for the application of neural networks to complex industrial problems". In: *IEEE Transactions on nuclear science* 44.3 (1997), pp. 1464–1468.

[SSZ17] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in Neural Information Processing Systems*. 2017, pp. 4077–4087.

[Sto79] George C. Stockman. "A minimax algorithm better than alpha-beta?" In: *Artificial Intelligence* 12.2 (1979), pp. 179–196.

[STT12] Ruslan Salakhutdinov, Joshua Tenenbaum, and Antonio Torralba. "One-shot learning with a hierarchical nonparametric bayesian model". In: *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. JMLR Workshop and Conference Proceedings. 2012, pp. 195–206.

[Su+19] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. "Vl-bert: Pre-training of generic visual-linguistic representations". In: *arXiv preprint arXiv:1908.08530* (2019).

[Sun+18] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. "Learning to compare: Relation network for few-shot learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1199–1208.

[Sun+19] Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. "FUNCTIONAL VARIATIONAL BAYESIAN NEURAL NETWORKS". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=rkxacs0qY7.

[Sze+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[Tia+20] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. "Rethinking Few-Shot Image Classification: A Good Embedding is All You Need?" In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIV*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Vol. 12359. Lecture Notes in Computer Science. Springer, 2020, pp. 266–282. DOI: 10.1007/978-

3-030-58568-6_16. URL: https://doi.org/10.1007/978-3-030-58568-6_16.

[Tri+20] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. "Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=rkgAGAVKPr.

[TS10] Lisa Torrey and Jude Shavlik. "Transfer learning". In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI Global, 2010, pp. 242–264.

[Tse+20] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. "Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=SJl5Np4tPr.

[Tuk77] John W. Tukey. *Exploratory data analysis*. Addison-Wesley series in behavioral science : quantitative methods. Addison-Wesley, 1977. ISBN: 0201076160. URL: https://www.worldcat.org/oclc/03058187.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[Vei+21] Olivier Veilleux, Malik Boudiaf, Pablo Piantanida, and Ismail Ben Ayed. "Realistic evaluation of transductive few-shot learning". In: *Advances in Neural Information Processing Systems* 34 (2021).

[Vil08] Cédric Villani. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.

[Vil09] Cédric Villani. *Optimal transport: old and new*. Vol. 338. Springer, 2009.

[Vin+16] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. "Matching networks for one shot learning". In: *Advances in neural information processing systems*. 2016, pp. 3630–3638.

[Vla+19] Mariia Vladimirova, Jakob Verbeek, Pablo Mesejo, and Julyan Arbel. "Understanding priors in bayesian neural networks at the unit level". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6458–6467.

[VNM07] John Von Neumann and Oskar Morgenstern. "Theory of games and economic behavior". In: *Theory of games and economic behavior*. Princeton university press, 2007.

[VPJ17] Valentin Vielzeuf, Stéphane Pateux, and Frédéric Jurie. "Temporal multimodal fusion for video emotion classification in the wild". In: *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. 2017, pp. 569–576.

[Wah+11] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. "The caltech-ucsd birds-200-2011 dataset". In: (2011).

[Wal+22] Reece Walsh, Mohamed H Abdelpakey, Mohamed S Shehata, and Mostafa M Mohamed. "Automated human cell classification in sparse datasets using few-shot learning". In: *Scientific Reports* 12.1 (2022), pp. 1–11.

[Wan+18] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. "Low-shot learning from imaginary data". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7278–7286.

[Wan+19a] Hong Wang, Xin Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. "Self-supervised learning for contextualized extractive summarization". In: *arXiv preprint arXiv:1906.04466* (2019).

[Wan+19b] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. "SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning". In: *CoRR* abs/1911.04623 (2019). arXiv: 1911.04623. URL: http://arxiv.org/abs/1911.04623.

[Wan+20] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. "Generalizing from a few examples: A survey on few-shot learning". In: *ACM computing surveys (csur)* 53.3 (2020), pp. 1–34.

[WBJ05] John Winn, Christopher M Bishop, and Tommi Jaakkola. "Variational message passing." In: *Journal of Machine Learning Research* 6.4 (2005).

[Wei05] Sanford Weisberg. *Applied linear regression*. Vol. 528. John Wiley & Sons, 2005.

[WKW16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big data* 3.1 (2016), pp. 1–40.

[WTH21] Davis Wertheimer, Luming Tang, and Bharath Hariharan. "Few-Shot Classification With Feature Map Reconstruction Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8012–8021.

[Wu+19a] Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E. Turner, Jose Miguel Hernandez-Lobato, and Alexander L. Gaunt. "Deterministic Variational Inference for Robust Bayesian Neural Networks". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=B1l08oAct7.

[Wu+19b] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. "Simplifying graph convolutional networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6861–6871.

[XGF16] Junyuan Xie, Ross Girshick, and Ali Farhadi. "Unsupervised deep embedding for clustering analysis". In: *International conference on machine learning*. PMLR. 2016, pp. 478–487.

[Xin+19] Chen Xing, Negar Rostamzadeh, Boris Oreshkin, and Pedro O O Pinheiro. "Adaptive cross-modal few-shot learning". In: *Advances in Neural Information Processing Systems* 32 (2019).

[Xu+21] Jingyi Xu, Hieu Le, Mingzhen Huang, ShahRukh Athar, and Dimitris Samaras. "Variational Feature Disentangling for Fine-Grained Few-Shot Classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8812–8821.

[Ye+20] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. "Few-shot learning via embedding adaptation with set-to-set functions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8808–8817.

[YLX21] Shuo Yang, Lu Liu, and Min Xu. "Free Lunch for Few-shot Learning: Distribution Calibration". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=JWOiYxMG92s.

[Yua+20] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. "Revisiting knowledge distillation via label smoothing regularization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3903–3911.

[Yue+20] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. "Interventional few-shot learning". In: *Advances in neural information processing systems* 33 (2020), pp. 2734–2746.

[Zha+17] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412* (2017).

[Zha+19] Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. "Variational few-shot learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1685–1694.

[Zha+20] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. "DeepEMD: Few-Shot Image Classification With Differentiable Earth Mover's Distance and Structured Classifiers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 12203–12213.

[Zha+21] Haipeng Zhang, Zhong Cao, Ziang Yan, and Changshui Zhang. "Sill-Net: Feature Augmentation with Separated Illumination Representation". In: *CoRR* abs/2102.03539 (2021). arXiv: 2102.03539. URL: https://arxiv.org/abs/2102.03539.

[Zha+22] Olivier Zhang, Nicolas Gengembre, Olivier Le Blouch, and Damien Lolive. "Dispeech: A Synthetic Toy Dataset for Speech Disentangling". In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 8557–8561.

[ZIE16] Richard Zhang, Phillip Isola, and Alexei A Efros. "Colorful image colorization". In: *European conference on computer vision*. Springer. 2016, pp. 649–666.

[Zik+20] Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. "Laplacian regularized few-shot learning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11660–11670.

[ZK16] Sergey Zagoruyko and Nikos Komodakis. "Wide Residual Networks". In: *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. Ed. by Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith. BMVA Press, 2016. URL: http://www.bmva.org/bmvc/2016/papers/paper087/index.html.

[ZK22] Hao Zhu and Piotr Koniusz. "EASE: Unsupervised Discriminant Subspace Learning for Transductive Few-Shot Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9078–9088.

[ZS20] Zhilu Zhang and Mert R. Sabuncu. "Self-Distillation as Instance-Specific Label Smoothing". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/1731592aca5fb4d789c4119c65c1Abstract.html.

[ZZK19] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. "Few-shot Learning via Saliency-guided Hallucination of Samples". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2770–2779.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Algorithmes et Prétraitement des Caractéristiques pour la Classification Transductive d'Images à Partir de peu de Données

Mots clés : Apprentissage Profond ; Apprentissage Automatique ; Apprentissage par Transfert ; Apprentissage Semi-Supervisé ; Apprentissage Few-Shot ; Clustering

Résumé : L'objectif de cette thèse est d'étudier l'un des défis les plus importants liés au développement de méthodes d'apprentissage automatique et profond. Notre recherche est menée dans le cadre où les modèles font des prédictions basées sur quelques exemples labélisés. En particulier, dans le contexte de la classification d'images, l'objectif de cette étude est d'apprendre un modèle capable de prédire correctement les labels de classe sur la base d'échantillons de données limités.

Nous discutons d'abord de l'amélioration des performances avec l'évolution des méthodes d'apprentissage profond, et présentons la problématique de peu de données. Dans un deuxième temps, nous introduisons les paramètres standards de cette problématique et présentons les méthodes de classification associées. Nous résumons un pipeline général pour s'y adresser.

Ensuite, nous mettons en évidence nos contributions qui adressent chaque étape du pipeline, en proposant des méthodes adaptives sur les données d'images ciblées dont le nombre est limité par le coût de l'annotation.

Enfin, nous tirons des conclusions de notre travail, ainsi que des discussions sur les nouveaux défis et les solutions potentielles liées au domaine.
.

Title : Algorithms and Feature Preprocessing for Transductive Few-Shot Image Classification

Keywords : Deep Learning ; Machine Learning ; Transfer Learning ; Semi-Supervised Learning ; Few-Shot Learning ; Clustering

Abstract : The purpose of this thesis is to investigate one of the most important challenges related to the development of machine and deep learning methods. Namely, our research is conducted in the setting where models make predictions based on a few labeled examples. Particularly in the context of image classification, the goal of this study is to learn a model that can correctly predict class labels based on limited data samples.

We firstly discuss the improved performance with the evolution of deep learning methods, and present the problematic of data thriftiness. Secondly, we introduce the standard settings of this problematic and present the related classification methods. We summarize a general pipeline to tackle it.

Then we highlight our contributions that address each step in the pipeline, by proposing adaptive methods on the targeted image data whose number is limited by the cost of annotation.

Finally, we draw conclusions of our work, along with discussions about the novel challenges as well as potential solutions related to the field.