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

Yuqing Hu
IMT Atlantique
Orange

Stéphane Pateux
Orange

Vincent Gripon
IMT Atlantique

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.

1 INTRODUCTION

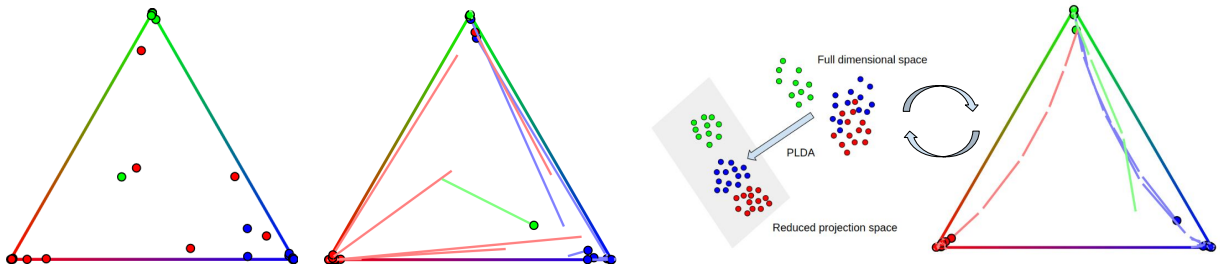
Few-shot learning, and in particular Few-Shot Classification, has become a subject of paramount importance in the last years with a large number of methodologies and discussions. Where large datasets continuously benefit from improved machine learning architectures, the ability to transfer this performance 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 (Antoniou et al., 2019; Snell et al., 2017; Ye et al., 2020; Rizve et al., 2021), 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 (Lee et al., 2021; Lazarou et al., 2021; Baik et al., 2021).

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, 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 (Veilleux et al., 2021). 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

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.



(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 very limited number of available samples, distribution-agnostic clustering algorithms are often preferred, such as K-MEANS or its variants (Moon, 1996; Lichtenstein et al., 2020; Ren et al., 2018) or mean-shift (Comaniciu and Meer, 2002).

In this paper, we are interested in showing it is possible to combine data reduction with statistical inference through a Variational Bayesian (VB) (Corduneanu and Bishop, 2001; Bishop and Nasrabadi, 2006) approach. Here, data reduction helps considerably reduce the number of parameters to infer, while VB inference 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 (Veilleux et al., 2021), 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 show the ability of the proposed algorithm to reach state-of-the-art transductive FSC performance on multiple vision benchmarks (balanced and unbalanced).
- We show the advantage of our proposed VB model and PLDA combination to have superior performance than alternative models and data reduction methods.

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 (Finn et al., 2017; Liu et al., 2020b; Baik et al., 2021; Vinyals et al., 2016; Oreshkin et al., 2018; Sung et al., 2018) that train a model in an episodic manner, and transfer learning methods (Chen et al., 2019; Mangla et al., 2020; Ziko et al., 2020; Boudiaf et al., 2020; Bendou et al., 2022; Rizve et al., 2021) that train a model with a set of mini-batches. Recent state-of-the-art works on inductive FSC (Ye et al., 2020; Zhang et al., 2020; Wertheimer et al., 2021; Kang et al., 2021) 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 (Gidaris and Komodakis, 2019; Chen et al., 2021; Yang et al., 2020; Hu et al., 2021a; Kim et al., 2019; Hamidouche et al., 2021) make use of the affinity among features and propose to group those that belong to the same class. More recent works such as (Hu et al., 2021b, 2022) 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 (Veilleux et al., 2021) 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 (Lichtenstein et al., 2020; Ren et al., 2018) have utilised algorithms such as Expectation Maximization (Dempster et al., 1977) (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 (Fox and Roberts, 2012; Rusu et al., 2019; Winn et al., 2005; Corduneanu and Bishop, 2001; Bishop and Nasrabadi, 2006), in which we regard some unknown parameters as hidden variables to inject more flexibility into the model, and we try to approximate the posterior of these 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 (Lichtenstein et al., 2020) applies a PCA method that reduces dimension in a non-supervised manner, and (Cao et al., 2020) 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 (Ioffe, 2006) (PLDA) that 1) is applied on extracted features, 2) fits data more desirably into distribution assumptions, and 3) is semi-supervised in combination with 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 (Veilleux et al., 2021) 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 (Veilleux et al., 2021) 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

In our work, PLDA (Ioffe, 2006) 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 (Yang et al., 2021). 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 $d = K - 1$. A more thorough description of the implementation can be found in Appendix.

3.3 Variational Bayesian 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 $\boldsymbol{\Lambda}_k = \boldsymbol{\Lambda} = T_{vb}\mathbf{I}$ for $k = 1, ..., K$, where T_{vb} is a hyper-parameter aiming at compensating 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:

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

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 (Veilleux et al., 2021), 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 (Prezhdo, 1999; Jaakkola and Jordan, 1998) 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}:

$$\begin{aligned}
q^*(\boldsymbol{\pi}) &= Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}), \\
q^*(\boldsymbol{\mu}) &= \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\beta_k\boldsymbol{\Lambda})^{-1})
\end{aligned} \tag{3}$$

where $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_k, ..., \alpha_K]$ are the estimated component weights for classes, $\alpha_k = \alpha_o + N_k$, 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 $\beta_k = \beta_o + N_k$ and the centroid $\boldsymbol{m}_k = \frac{1}{\beta_k}(\beta_o\boldsymbol{m}_o + \sum_{n=1}^N o_{nk}\boldsymbol{u}_n)$ 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 $\boldsymbol{m}_k$.

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

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

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

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

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 $\hat{\ell}(\boldsymbol{x}_n) = \arg\max_k (o_{nk})$

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 (Kearns et al., 1998; Lichtenstein et al., 2020; Ren et al., 2018) 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.

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 (Russakovsky et al., 2015), *tiered*-Imagenet Ren et al. (2018) and caltech-ucsd birds-200-2011 (CUB) (Wah et al., 2011). *mini*-Imagenet is a subset of ILSVRC-12 (Russakovsky et al., 2015) 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 benchmarks such as FC100 (Oreshkin et al., 2018) and CIFAR-FS (Bertinetto et al., 2019).

Settings Following previous works (Lichtenstein et al., 2020; Rodríguez et al., 2020; Veilleux et al., 2021), 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 (Veilleux et al., 2021) and set $\alpha_o^* = 2.0$, further experiments with different values are conducted next. The performance of our proposed method is evaluated by the averaged accuracy over $10,000$ Few-Shot tasks with a 95% confidence interval.

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 (Veilleux et al., 2021). 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 (Wang et al., 2019). 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 (Mangla et al., 2020; Bendou et al., 2022), 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 (Veilleux et al., 2021), and add the performance of our proposed method in comparison, using the same pre-

trained RN18 and WRN feature extractors, and we also report the result of a RN12 backbone pretrained following (Bendou et al., 2022). 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 (Hu et al., 2021b) 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. As for the time complexity of BAVARDAGE, we set n_{step} to be 5 for all experiments in the paper. The average execution time per few-shot task is around $1.7e-4$ seconds.

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 VB and PLDA. 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 combining both VB and PLDA.

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 (Van der Maaten and Hinton, 2008) 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.

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, applying PCA before the VB inference (since it is unsupervised), we obtain 67.10%/76.95% accuracy for 1/5 shots in the unbalanced setting (dataset:*mini*-Imagenet, backbone: WRN from (Veilleux et al., 2021)). Applying LDA by computing the projection matrix from $\mathbf{\Phi}_w^{-1}\mathbf{\Phi}_b$ instead of Eq. 1, 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.

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, a trade-off between expressivity and risk of overfitting should be looked after.

To better illustrate 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 (Veilleux et al., 2021)), 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-

Table 1: Comparison of the state-of-the-art methods on *mini*-Imagenet, *tiered*-Imagenet and CUB datasets using the same pretrained backbones as (Veilleux et al., 2021), along with the accuracy of our proposed method on a ResNet-12 backbone pretrained following the methodology described in (Bendou et al., 2022), reported to achieve state-of-the-art performance. Note that the full results of BAVARDAGE with 95% confidence intervals are provided in Appendix.

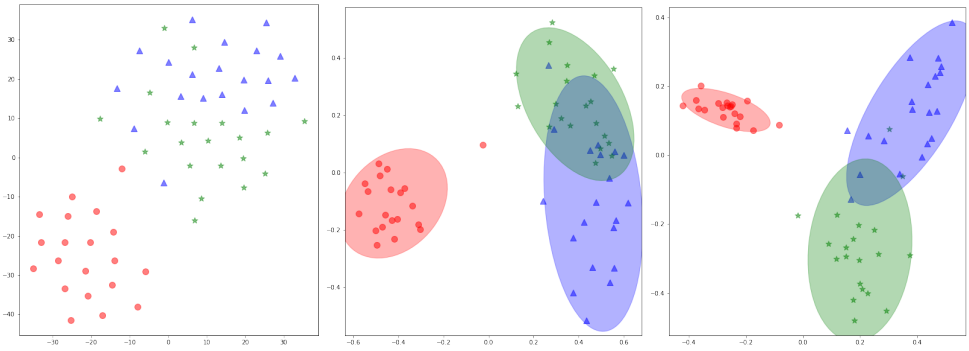
***mini*-Imagenet**		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
MAML (Finn et al., 2017)		47.6/−	64.5/−	51.4/−	69.5/−
Versa (Gordon et al., 2019)		47.8/−	61.9/−	50.0/−	65.6/−
Entropy-min (Dhillon et al., 2020)		58.5/60.4	74.8/76.2	63.6/66.1	82.1/84.2
PT-MAP (Hu et al., 2021b)		60.1/60.6	67.1/66.8	**76.9**/**78.9**	**85.3**/86.6
LaplacianShot (Ziko et al., 2020)	RN18/WRN (Veilleux et al., 2021)	65.4/70.0	81.6/83.2	70.1/72.9	82.1/83.8
BD-CSPN (Liu et al., 2020b)		67.0/70.4	80.2/82.3	69.4/72.5	82.0/83.7
TIM (Boudiaf et al., 2020)		67.3/69.8	79.8/81.6	71.8/74.6	83.9/85.9
α-TIM (Veilleux et al., 2021)		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 (Bendou et al., 2022)	**77.8**	**88.0**	**82.7**	**89.5**
***tiered*-Imagenet**		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Entropy-min (Dhillon et al., 2020)		61.2/62.9	75.5/77.3	67.0/68.9	83.1/84.8
PT-MAP (Hu et al., 2021b)		64.1/65.1	70.0/71.0	**82.9**/**84.6**	**88.8**/**90.0**
LaplacianShot (Ziko et al., 2020)		72.3/73.5	85.7/86.8	77.1/78.8	86.2/87.3
BD-CSPN (Liu et al., 2020b)	RN18/WRN (Veilleux et al., 2021)	74.1/75.4	84.8/85.9	76.3/77.7	86.2/87.4
TIM (Boudiaf et al., 2020)		74.1/75.8	84.1/85.4	78.6/80.3	87.7/88.9
α-TIM (Veilleux et al., 2021)		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 (Bendou et al., 2022)	**79.4**	**88.0**	**83.5**	**89.0**
CUB		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
PT-MAP (Hu et al., 2021b)		65.1	71.3	85.5	91.3
Entropy-min (Dhillon et al., 2020)		67.5	82.9	72.8	88.9
LaplacianShot (Ziko et al., 2020)		73.7	87.7	78.9	88.8
BD-CSPN (Liu et al., 2020b)	RN18 (Veilleux et al., 2021)	74.5	87.1	77.9	88.9
TIM (Boudiaf et al., 2020)		74.8	86.9	80.3	90.5
α-TIM (Veilleux et al., 2021)		75.7	89.8	−	−
BAVARDAGE (ours)		**82.0**	**90.7**	**85.6**	**91.4**
BAVARDAGE (ours)	RN12 (Bendou et al., 2022)	**83.1**	**90.8**	**87.4**	**92.0**

Table 2: Ablation study on using VB and PLDA in our method, with results 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**

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 is close to optimal.

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 $\alpha_o^* = 1$.



(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.

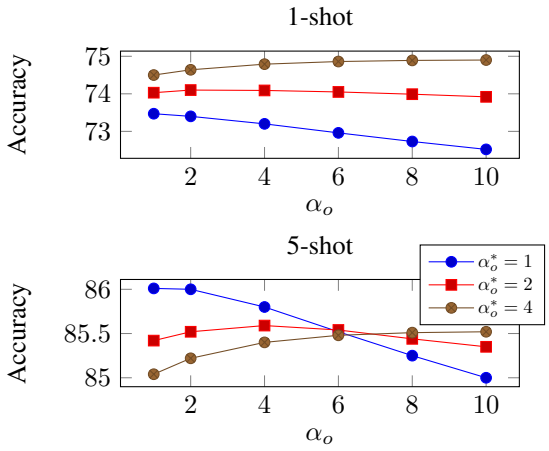


Figure 3: 1-shot and 5-shot accuracy on different imbalance levels (varying α_o^*) as a function of VB priors α_o (dataset: *mini*-Imagenet, backbone: WRN).

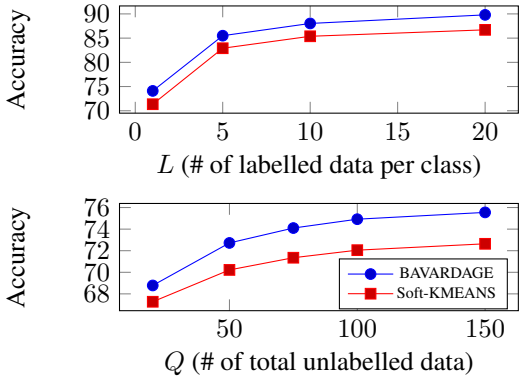


Figure 4: Accuracy as a funtion of L and Q in comparison with Soft-KMEANS (dataset: *mini*-Imagenet, backbone: WRN).

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.

Comparison with Similar Work Aligned with our proposed method, there is another work proposed in (Yang et al., 2021) that also uses the base dataset to estimate the distribution of the novel classes. However, the differences are that 1) (Yang et al., 2021) is an inductive method, while BAVARDAGE is a transductive method; 2) In (Yang et al., 2021) the covariance matrix of each novel class is calibrated using the closest base classes, while BAVARDAGE uses a fixed covariance matrix for all novel classes, and it is computed using all samples in the base dataset (See Eq. 6 in Appendix). To compare (Yang et al., 2021) with BAVARDAGE, here in Table 3 we conduct two experiments: 1) use the inductive method in (Yang et al., 2021) and report the performance in both balanced and unbalanced settings; and 2) integrate the calibrated covariance matrix computed using (Yang et al., 2021) into PLDA to perform BAVARDAGE.

We can see that in experiment 1), we compare BAVARDAGE directly with (Yang et al., 2021). Note that for a fair comparison, here we use the same WRN backbone pretrained in (Veilleux et al., 2021), while in (Yang et al., 2021) the authors train a WRN backbone following (Mangla et al., 2020). Hence the accuracy difference ($68.6\%/82.9\%$ reported in (Yang et al., 2021) for 1/5-shot balanced setting). In experiment 2) we observe that in BAVARDAGE, a fixed covariance matrix performs better than a covariance matrix calibrated per task. We think that it is thanks to a better bias/variance trade-off.

Table 3: Comparison with (Yang et al., 2021) (dataset: mini-ImageNet, backbone: WRN). Experiment 1) is a direct comparison between BAVARDAGE and (Yang et al., 2021), while experiment 2) computes $\mathbf{\Phi}_w$ using Distribution Calibration (DC) proposed in (Yang et al., 2021).

***mini*-Imagenet**			**unbalanced**		**balanced**	
Experiment	Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
1)	Logistic Regression with DC (Yang et al., 2021)	WRN (Veilleux et al., 2021)	65.6	80.3	65.7	80.4
2)	BAVARDAGE (ours) with DC		71.2	82.4	75.6	84.5
	BAVARDAGE (ours)	WRN (Veilleux et al., 2021)	**74.1**	**85.5**	**78.5**	**87.4**

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 in Table 4 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 we perform BAVARDAGE based on $\mathbf{\Phi}_w$ being the within-class scatter matrix of *mini*-Imagenet as well.

Table 4: 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 ResNet-18 and WideResNet28_10 backbones from (Veilleux et al., 2021).

***mini* → CUB**		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
NCM	RN18 (Veilleux et al., 2021)	46.3	66.2	46.3	66.1
BAVARDAGE (ours)		**53.0**	**70.0**	**54.6**	**71.5**
NCM	WRN (Veilleux et al., 2021)	48.5	66.3	48.5	68.2
BAVARDAGE (ours)		**56.6**	**74.0**	**58.1**	**75.4**

Table 5: Comparison of our proposed method with state-of-the-art methods on cross domain, accuracy is obtained with ResNet-18 backbone pretrained from (Veilleux et al., 2021).

***mini* → CUB**		
Method	Backbone	**5-shot**
MAML (Finn et al., 2017)	RN18 (Veilleux et al., 2021)	51.3
MatchNet (Vinyals et al., 2016)		53.1
RelatNet (Sung et al., 2018)		57.7
ProtoNet (Snell et al., 2017)		62.0
SimpleShot (Wang et al., 2019)		64.0
Baseline (Chen et al., 2019)		65.6
LaplacianShot (Ziko et al., 2020)		66.3
Neg-Cosine (Liu et al., 2020a)		67.0
TIM-ADM (Boudiaf et al., 2020)		70.3
TIM-GD (Boudiaf et al., 2020)		71.0
BAVARDAGE (ours)		**71.5**

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 (Sbai et al., 2020), and how to choose or design a base set that maximizes the boost in accuracy when evaluating with test data. For further comparison, in Table 5 we report our 5-shot performance under the balanced setting along with other state-of-the-art methods following (Boudiaf et al., 2020). With the same pretrained ResNet-18 backbone, our proposed method obtains the best accuracy.

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. The performance in the balanced setting looks less appealing because in BAVARDAGE we do not make use of the class-balanced prior, while current state-of-the-art methods do or they tend to use additional data. And we consider it unrealistic to have such a prior in real world scenarios (Veilleux et al., 2021).

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 (Veilleux et al., 2021), we encourage the community to rethink the works in transductive settings such as imbalance generation to provide fairer grounds of comparison between the various proposed approaches.

Acknowledgements

This work was financially supported by Orange. The author is grateful to those who contributed to the ideas, and also to reviewers who provided useful comments.

References

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

Baik, S., Choi, J., Kim, H., Cho, D., Min, J., and Lee, K. M. (2021). 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.

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

Bertinetto, L., Henriques, J. F., Torr, P. H. S., and Vedaldi, A. (2019). 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.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

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

Cao, T., Law, M. T., and Fidler, S. (2020). 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.

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

Chen, W., Liu, Y., Kira, Z., Wang, Y. F., and Huang, J. (2019). 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.

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

Corduneanu, A. and Bishop, C. M. (2001). Variational bayesian model selection for mixture distributions. In *Artificial intelligence and Statistics*, volume 2001, pages 27–34. Morgan Kaufmann Waltham, MA.

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

Dhillon, G. S., Chaudhari, P., Ravichandran, A., and Soatto, S. (2020). 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.

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

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

Gidaris, S. and Komodakis, N. (2019). 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.

Gordon, J., Bronskill, J., Bauer, M., Nowozin, S., and Turner, R. E. (2019). 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.

Hamidouche, M., Lassance, C., Hu, Y., Drumetz, L., Pasdeloup, B., and Gripon, V. (2021). Improving classification accuracy with graph filtering. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 334–338. IEEE.

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

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

Hu, Y., Pateux, S., and Gripon, V. (2022). Squeezing backbone feature distributions to the max for efficient few-shot learning. *Algorithms*, 15(5):147.

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

Jaakkola, T. S. and Jordan, M. I. (1998). Improving the mean field approximation via the use of mixture distribu-

tions. In *Learning in graphical models*, pages 163–173. Springer.

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

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

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

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

Lazarou, M., Stathaki, T., and Avrithis, Y. (2021). 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.

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

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

Liu, B., Cao, Y., Lin, Y., Li, Q., Zhang, Z., Long, M., and Hu, H. (2020a). Negative margin matters: Understanding margin in few-shot classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 438–455. Springer.

Liu, J., Song, L., and Qin, Y. (2020b). 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.

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

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

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

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

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

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. (2018). 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.

Rizve, M. N., Khan, S., Khan, F. S., and Shah, M. (2021). 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.

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

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

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. (2019). 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.

Sbai, O., Couprie, C., and Aubry, M. (2020). Impact of base dataset design on few-shot image classification. In *European Conference on Computer Vision*, pages 597–613. Springer.

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

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

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

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

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. In

Advances in neural information processing systems, pages 3630–3638.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

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

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

Winn, J., Bishop, C. M., and Jaakkola, T. (2005). Variational message passing. *Journal of Machine Learning Research*, 6(4).

Yang, L., Li, L., Zhang, Z., Zhou, X., Zhou, E., and Liu, Y. (2020). 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.

Yang, S., Liu, L., and Xu, M. (2021). 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.

Ye, H.-J., Hu, H., Zhan, D.-C., and Sha, F. (2020). 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.

Zhang, C., Cai, Y., Lin, G., and Shen, C. (2020). 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.

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

A IMPLEMENTATION DETAILS

A.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 $\mathbf{R} = [\boldsymbol{r}_1, ..., \boldsymbol{r}_i, ..., \boldsymbol{r}_D] \in \mathbb{R}^{D \times D}$ 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 $\mathbf{\Psi}$ of sphered samples can be calculated as:

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

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 $\mathbf{\Psi}$, 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 $\mathbf{\Psi}$ 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:

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

The entire process is described in Algorithm 2.

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 $\mathbf{U}$

A.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:

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

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:

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

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, 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.

A.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, 5, 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.

B ADDITIONAL EXPERIMENTS

B.1 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 (Krizhevsky et al., 2009) 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-novel splits as for *mini*-Imagenet. Each class contains 600 images of size 32x32 pixels. In Table 6 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 6 we also show the results using WRN and RN18 pretrained from (Veilleux et al., 2021) and RN12 pretrained from (Bendou et al., 2022), same as Table 1 in the paper, with a confidence interval of 95% added next to the accuracy. In addition, given that some works (Luo et al., 2021; Zhang et al., 2020) in the field utilize data augmentation techniques

[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`

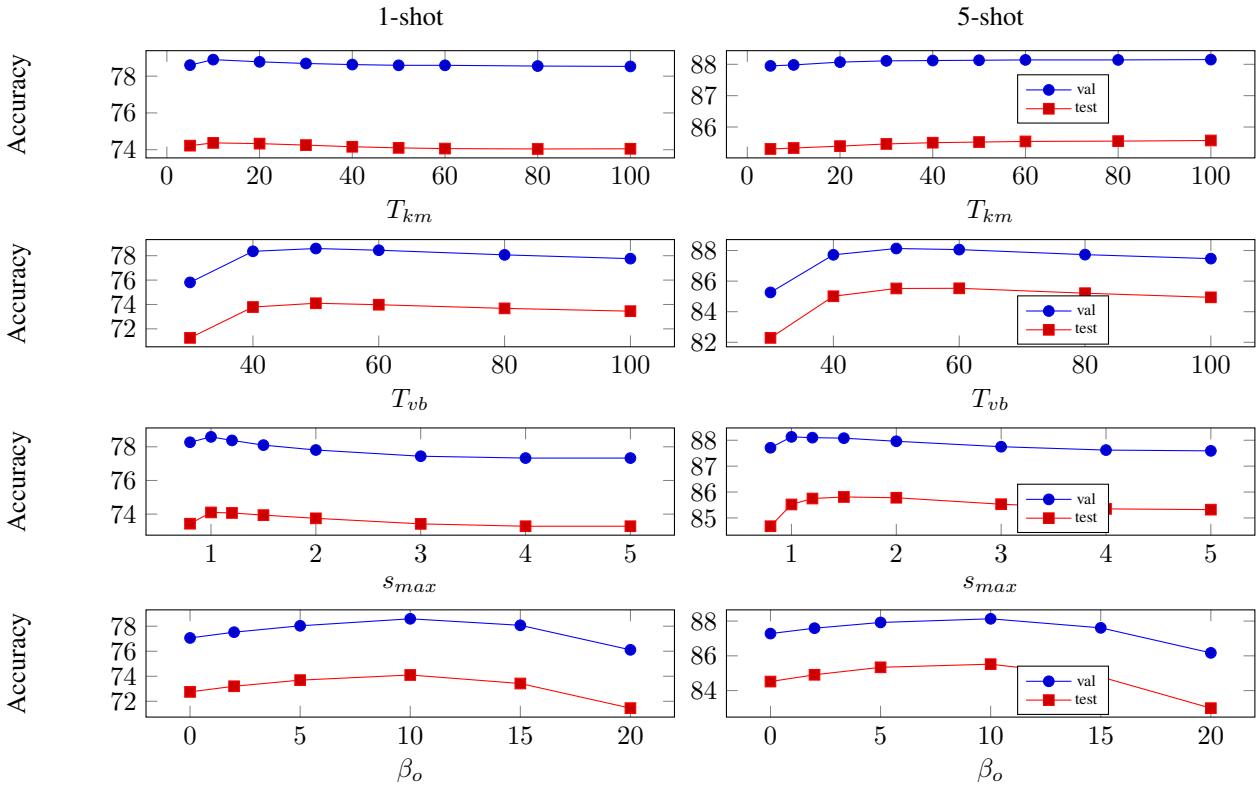


Figure 5: Hyperparameter tuning of our proposed method. Here we tune 4 hyperparameters of BAVARDAGE on *mini*-Imagenet (backbone: WRN) in the unbalanced setting.

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 (Bendou et al., 2022) 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.

Table 6: 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 (Veilleux et al., 2021), RN12 and RN12* pretrained from (Bendou et al., 2022).

***mini*-Imagenet**		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Soft-KMEANS	RN18 (Veilleux et al., 2021)	$68.82 \pm 0.27\%$	$81.27 \pm 0.17\%$	$73.47 \pm 0.26\%$	$83.04 \pm 0.15\%$
	WRN (Veilleux et al., 2021)	$71.35 \pm 0.27\%$	$82.41 \pm 0.16\%$	$75.70 \pm 0.25\%$	$84.42 \pm 0.14\%$
	RN12 (Bendou et al., 2022)	$75.65 \pm 0.25\%$	$86.35 \pm 0.14\%$	$80.81 \pm 0.24\%$	$87.92 \pm 0.12\%$
	RN12* (Bendou et al., 2022)	$77.51 \pm 0.26\%$	$87.78 \pm 0.14\%$	$82.14 \pm 0.24\%$	$89.08 \pm 0.12\%$
BAVARDAGE	RN18 (Veilleux et al., 2021)	$71.01 \pm 0.31\%$	$83.60 \pm 0.17\%$	$75.07 \pm 0.28\%$	$84.49 \pm 0.14\%$
	WRN (Veilleux et al., 2021)	$74.10 \pm 0.30\%$	$85.52 \pm 0.16\%$	$78.51 \pm 0.27\%$	$87.41 \pm 0.13\%$
	RN12 (Bendou et al., 2022)	$77.85 \pm 0.28\%$	$88.02 \pm 0.14\%$	$82.67 \pm 0.25\%$	$89.50 \pm 0.11\%$
	RN12* (Bendou et al., 2022)	$79.76 \pm 0.29\%$	$89.85 \pm 0.13\%$	$84.80 \pm 0.25\%$	$91.65 \pm 0.10\%$
***tiered*-Imagenet**		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Soft-KMEANS	WRN (Veilleux et al., 2021)	$73.92 \pm 0.28\%$	$85.02 \pm 0.18\%$	$78.59 \pm 0.27\%$	$85.76 \pm 0.16\%$
	RN18 (Veilleux et al., 2021)	$73.79 \pm 0.28\%$	$84.65 \pm 0.18\%$	$78.34 \pm 0.27\%$	$85.52 \pm 0.17\%$
	RN12 (Bendou et al., 2022)	$78.15 \pm 0.27\%$	$87.65 \pm 0.17\%$	$83.11 \pm 0.25\%$	$88.80 \pm 0.15\%$
	RN12* (Bendou et al., 2022)	$79.62 \pm 0.27\%$	$88.61 \pm 0.16\%$	$84.08 \pm 0.24\%$	$89.56 \pm 0.14\%$
BAVARDAGE	WRN (Veilleux et al., 2021)	$77.45 \pm 0.31\%$	$87.48 \pm 0.18\%$	$81.47 \pm 0.28\%$	$88.27 \pm 0.16\%$
	RN18 (Veilleux et al., 2021)	$76.55 \pm 0.31\%$	$86.46 \pm 0.19\%$	$80.32 \pm 0.28\%$	$87.14 \pm 0.16\%$
	RN12 (Bendou et al., 2022)	$79.38 \pm 0.29\%$	$88.04 \pm 0.18\%$	$83.52 \pm 0.26\%$	$89.03 \pm 0.15\%$
	RN12* (Bendou et al., 2022)	$81.17 \pm 0.29\%$	$89.63 \pm 0.17\%$	$85.20 \pm 0.25\%$	$90.41 \pm 0.14\%$
CUB		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Soft-KMEANS	RN18 (Veilleux et al., 2021)	$77.54 \pm 0.26\%$	$86.70 \pm 0.14\%$	$82.67 \pm 0.24\%$	$89.04 \pm 0.11\%$
	RN12 (Bendou et al., 2022)	$81.24 \pm 0.25\%$	$87.27 \pm 0.14\%$	$84.87 \pm 0.22\%$	$89.64 \pm 0.11\%$
	RN12* (Bendou et al., 2022)	$82.40 \pm 0.24\%$	$89.40 \pm 0.13\%$	$87.38 \pm 0.20\%$	$91.29 \pm 0.10\%$
BAVARDAGE	RN18 (Veilleux et al., 2021)	$82.00 \pm 0.28\%$	$90.67 \pm 0.12\%$	$85.64 \pm 0.25\%$	$91.42 \pm 0.10\%$
	RN12 (Bendou et al., 2022)	$83.12 \pm 0.26\%$	$90.81 \pm 0.12\%$	$87.41 \pm 0.22\%$	$92.03 \pm 0.09\%$
	RN12* (Bendou et al., 2022)	$86.96 \pm 0.24\%$	$92.84 \pm 0.10\%$	$90.42 \pm 0.20\%$	$93.50 \pm 0.08\%$
FC100		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Soft-KMEANS	RN12 (Bendou et al., 2022)	$51.24 \pm 0.27\%$	$64.70 \pm 0.22\%$	$54.59 \pm 0.26\%$	$66.37 \pm 0.20\%$
	RN12* (Bendou et al., 2022)	$51.64 \pm 0.27\%$	$65.26 \pm 0.22\%$	$54.87 \pm 0.26\%$	$66.89 \pm 0.20\%$
BAVARDAGE	RN12 (Bendou et al., 2022)	$52.60 \pm 0.32\%$	$65.35 \pm 0.25\%$	$56.66 \pm 0.28\%$	$69.69 \pm 0.21\%$
	RN12* (Bendou et al., 2022)	$53.78 \pm 0.30\%$	$68.75 \pm 0.24\%$	$57.27 \pm 0.29\%$	$70.60 \pm 0.21\%$
CIFAR-FS		**unbalanced**		**balanced**	
Method	Backbone	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Soft-KMEANS	RN12 (Bendou et al., 2022)	$80.72 \pm 0.25\%$	$88.31 \pm 0.17\%$	$85.47 \pm 0.22\%$	$89.36 \pm 0.15\%$
	RN12* (Bendou et al., 2022)	$81.75 \pm 0.25\%$	$88.92 \pm 0.17\%$	$86.07 \pm 0.22\%$	$89.85 \pm 0.15\%$
BAVARDAGE	RN12 (Bendou et al., 2022)	$82.68 \pm 0.27\%$	$88.97 \pm 0.18\%$	$86.20 \pm 0.23\%$	$89.58 \pm 0.15\%$
	RN12* (Bendou et al., 2022)	$83.82 \pm 0.27\%$	$89.84 \pm 0.18\%$	$87.35 \pm 0.23\%$	$90.63 \pm 0.16\%$