UNEM: UNrolled Generalized EM for Transductive Few-Shot Learning

Anonymous CVPR submission

Paper ID 16744

Abstract

Transductive few-shot learning has recently triggered wide attention in computer vision. Yet, current methods introduce key hyper-parameters, which control the prediction statistics of the test batches, such as the level of class balance, affecting performances significantly. Such hyper-parameters are empirically grid-searched over validation data, and their configurations may vary substantially with the target dataset and pre-training model, making such empirical searches both sub-optimal and computationally intractable. In this work, we advocate and introduce the unrolling *paradigm, also referred to as "learning to optimize", in the context of few-shot learning, thereby learning efficiently and effectively a set of optimized hyper-parameters. Specifically, we unroll a generalization of the ubiquitous Expectation-Maximization (EM) optimizer into a neural network architecture, mapping each of its iterates to a layer and learning a set of key hyper-parameters over validation data. Our unrolling approach covers various statistical feature distributions and pre-training paradigms, including recent foundational vision-language models and standard vision-only classifiers. We report comprehensive experiments, which cover a breadth of fine-grained downstream image classification tasks, showing significant gains brought by the proposed unrolled EM algorithm over iterative variants. The achieved improvements reach up to 10% and 7.5% on vision-only and vision-language benchmarks, respectively. The source code and learned parameters are available at* https://anonymous.4open.science/r/UNEM.

1. Introduction

Deep learning has transfigured computer vision, driving substantial progress in tasks such as image classification, captioning, object detection and segmentation. However, these successes often come with a high cost: the requirement for large amounts of labeled data. Additionally, the generalization of these models is seriously challenged when evaluated on new classes (concepts), unseen during pre-training, or when operating under distribution shifts [1, 51].

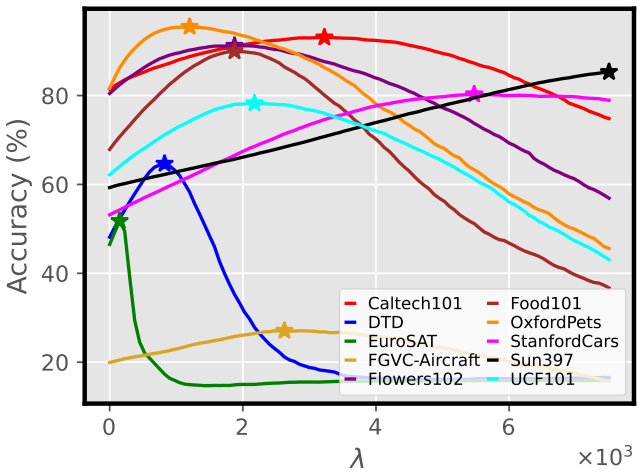

Figure 1. Impact of the class-balance hyperparameter λ on the accuracy of transductive few-shot classification. The accuracy results are obtained using the EM-Dirichlet algorithm [33] applied to vision-language models (with 4-shots). The plot shows that the choice of λ has a strong impact on the performance, and that the optimal λ (indicated with the star symbol) might vary by orders of magnitudes, depending on the target downstream dataset (e.g., the ten very different fine-grained classification datasets). Further comments on the optimal λ values and the values chosen in [33] are provided in Section 4. The values of the learned hyper-parameters based on the proposed unrolled algorithm are illustrated and analyzed in Appendix D.

To address these challenges, Few-Shot Learning (FSL) has recently attracted wide attention within the computer vision community. In the standard few-shot image classification setting, a feature extractor is first pre-trained on a set of classes, often called the base classes. Then, the model is adapted and evaluated on new classes and tasks. Each evaluation task is performed on a set of unlabeled samples (referred to as the query set), and supervised by a support set composed of few labeled samples per new class. Earlier FSL methods have relied on various concepts, including meta-learning [11, 19, 39], transfer learning [13, 37, 41], and metric learning [20, 26, 49]. However, most of these

works operate within the *inductive* setting, where, at inference time, the prediction for each sample is made independently from the other samples in the query set.

Recently, significant attention has shifted to the *transductive* scheme, in which inference is performed jointly on a batch of query samples. Transduction leverages the statistics of the unlabeled query samples, yielding notable performance gains, and have triggered a large body of recent works in few-shot learning, with various methods and mechanisms [4, 23–25, 29, 32, 57], as detailed in Section 2. Generally, transductive methods are built upon optimizing objective functions that integrate clustering terms, either discriminative as in information maximization [4, 48] or generative as in probabilistic K-means [32, 33]. However, many among these assume a perfectly balanced class distribution within the query set, and incorporate terms in the objective function or constraints that enforce a class-balance prior. The latter could limit the applicability of these methods, whose performances were shown to drop significantly when dealing with imbalanced query sets [27, 48].

Several recent attempts tackled this limitation, to handle more realistic scenarios [24, 32, 33, 46, 48]. Yet, these methods introduce key hyper-parameters, which control the prediction statistics of the unlabeled query set, such as the level of class balance. Such hyper-parameters are empirically grid-searched over pre-defined sets of values using validation data, and their optimal configurations may vary substantially with the target dataset and pre-training model [17]. To illustrate this, we depict in Fig. 1 the accuracy as a function of a class-balance hyper-parameter for the recent method in [33]. One may observe that this hyper-parameter has a crucial effect on the performance, and its optimal value might vary by orders of magnitude across the target datasets. The issue is further compounded by additional hyper-parameters, which makes intensive empirical grid searches for the hyper-parameters over validation data and pre-defined intervals of values both sub-optimal and computationally intractable. Therefore, we advocate and introduce the unrolling paradigm (also called "learning to optimize") in the context of few-shot learning, which enables to learn efficiently and effectively a set of optimized hyper-parameters. Specifically, our main contributions could be summarized as follows:

1. We study a generalization of the ubiquitous Expectation-Maximization (EM) algorithm, in which we make two hyper-parameters controlling prediction statistics – class balance and prediction entropy – explicit[1]. Our generalization encompasses several existing transductive few-shot methods as particular cases and, more importantly, enables to learn these crucial hyper-parameters through the proposed unrolling strategy.
2. We unroll the generalized EM optimizer into a neural network architecture, mapping each of its iterates to a layer and learning the introduced hyper-parameters over validation data. Our unrolling approach offers greater flexibility in optimizing the hyper-parameters, allowing them to vary across the network's layers. To the best of our knowledge, this study is the first to investigate unrolling in transductive few-shot learning.
3. We design our unrolling architecture in way that covers various statistical assumptions and pre-training paradigms: (i) Gaussian for vision-only classifiers and (ii) Dirichlet for vision-language models.
4. We report comprehensive experiments, which cover a breadth of fine-grained downstream image classification tasks and different pre-training models, showing consistent and substantial gains over recent state-of-the-art methods.

[1]Those hyper-parameters are implicit (hidden) in the standard EM formulation.

2. Related works

Few-shot classification with vision-only models: Few-shot classification using vision models has been widely explored in the literature, leading to the development of various methods. The first category, inductive methods, predicts the class of each test sample (in the query set) independently from others [16, 30]. In contrast, the second category, transductive methods, which has gained increasing attention in recent years, involves jointly predicting classes for a batch of test samples in each few-shot task. Numerous research efforts in transductive methods have leveraged concepts like clustering [24, 32], label propagation [29, 57], information maximization [4, 48], optimal transport [23, 47], prototype estimation [28, 57], and variational networks [25, 42], among others. Studies have shown that transductive methods significantly outperform inductive approaches, achieving accuracy gains of up to 15% as reported in several evaluations [4, 32].

Few-shot classification with vision-language models: Contrastive Vision-Language Pre-training (CLIP) has recently emerged as an effective model for enhancing various vision tasks through visual-text pairs. By learning transferable visual models with natural language supervision, CLIP demonstrates strong performance in zero-shot classification by matching the image features to text embeddings of novel classes [38]. To further enhance its classification capabilities, several studies have extended CLIP to the few-shot setting [6, 18, 53, 55, 56]. While the linear probe model [38] trains a logistic regression classifier using CLIP image features, the authors of proposed in [18] proposed a generalization of this baseline, modeling the classifier weights as learnable functions of the visual prototypes and text embeddings. Context Optimization (CoOp) [55] and its extended versions [6, 53, 56] have been developed

based on the concept of prompt learning. While CoOp [55] aims to model context in prompts using continuous representations, a knowledge-guided CoOp is proposed in [53] to enhance the generalization ability by minimizing the discrepancy between the learnable prompts and the original ones. In [6], the authors learn multiple prompts, describing the characteristics of each class, through the minimization of an optimal-transport distance. Unlike prompt learning methods, which fine-tune the input text, another family of methods, referred to as *adapters*, aims to transform the visual or text encoders, such as [12, 54]. For instance, TIP-Adapter [54] adds a non-parametric adapter to the weight-frozen CLIP model, and updates the prior knowledge encoded in CLIP by feature retrieval. CLIP-Adapter [12] introduces a multi-layered perceptron to learn new features, and combine them with the original CLIP-encoded features via residual connections.

It is worth noting that all the aforementioned methods belong to the inductive family. However, unlike vision-only models, the transductive few-shot setting is still not well investigated in the context of CLIP. To the best of our knowledge, the only transductive few-shot CLIP method is the one very recently proposed in [33], which, inspired by the Expectation Maximization algorithm, relies on the Dirichlet distribution to model the data. Moreover, as reported in [33], and unlike the behaviors observed with vision-only models, recent transductive few-shot methods do not always outperform their inductive counterparts with CLIP. This suggests that there is a need to further investigate transductive methods in the context of vision-language models, due to their aforementioned advantages with respect to inductive methods.

Class-balance and hyperparameters setting: Most of the existing transductive few-shot methods have been designed for perfectly balanced query sets (i.e., uniform class distribution), and have shown drops in performances under class-imbalanced settings. To address this flaw, various methods have been developed in the last years [24, 32, 33, 46, 48]. For instance, In [46], the categorical probability of each query sample is regularized to quantify the difference between the class marginal distribution and the uniform one. The works in [24, 32, 33, 48] explored various weighted terms, which are added to the objective functions, to mitigate the effect of the class-balance bias. To this end, some hyper-parameters are introduced, to weigh to contributions of such terms in the objective function. Unlike [32], where the introduced hyper-parameter has been theoretically chosen in order to compensate for the hidden class-balance bias in the used clustering objective, the hyper-parameters are often set in an empirical manner based on the validation classes of each dataset. Thus, a set of hyperparameter values (in a given range) are evaluated, and the ones yielding the highest accuracy are selected.

3. Proposed methodology

3.1. Generalized EM algorithm

Preliminaries: Let us first introduce the notations to formulate our transductive few-shot inference. Let $\{\boldsymbol{z}_n\}_{1\leq n\leq N}$ denote the set of feature vectors extracted from a pre-training network, and which are to be classified, with N the total number of samples within a given task. The whole dataset contains K distinct classes, whereas the number of randomly sampled classes present in each mini-batch task might be much smaller than K. This subset of sampled classes may also vary across mini-batches, and the method does not assume any prior knowledge on the specific classes that may appear in each mini-batch.

Moreover, for a given few-shot task, let us denote by $\mathbb{S} \subset \{1,\dots,N\}$ and $\mathbb{Q} = \{1,\dots,N\} \setminus \mathbb{S}$ the indices of samples within the support (labeled) and query (unlabeled) mini-batch sets, respectively. For every $n \in \mathbb{S}$, $\boldsymbol{y}_n = (y_{n,k})_{1\leq k\leq K} \in \{0,1\}^K$ are the one-hot-encoded labels, such that, for every $k \in \{1,\dots,K\}$, $y_{n,k} = 1$ if the n-th sample belongs to class k, and $y_{n,k} = 0$ otherwise.

Finally, given the extracted feature vectors $\boldsymbol{z}_n$, we assume that the data probability distribution knowing its class k is modeled by a given law, whose probability density function (pdf) is denoted by $\mathrm{p}(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k)$ and characterized by a vector of parameters $\boldsymbol{\theta}_k$. This means that the global distribution of $\boldsymbol{z}_n$ is a mixture of these pdfs.

Problem formulation: Our goal is to identify the classes of the unlabeled samples in the query set by optimizing a general clustering objective function, while embedding supervision constraints from the few labeled samples within the support set. We do so by unrolling iterative block-coordinate optimizers of the objective functions over two sets of variables:

- Soft assignment vectors $\boldsymbol{u} = (\boldsymbol{u}_n)_{1\leq n\leq N} \in (\Delta_K)^N$, where Δ_K is the probability simplex of $\mathbb{R}^K$. For every $n \in \{1,\dots,N\}$, $\boldsymbol{u}_n = (u_{n,k})_{1\leq k\leq K}$ where, for every $k \in \{1,\dots,K\}$, $u_{n,k}$ can be interpreted as the probability that the n-th sample belongs to class k. These probabilities have to be determined for the query set samples.
- Feature distribution parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_k)_{1\leq k\leq K}$.

Consider the following general probabilistic clustering problem:

$$\begin{aligned} &\underset{\boldsymbol{u},\boldsymbol{\theta}}{\text{minimize}}\, \mathcal{L}(\boldsymbol{u},\boldsymbol{\theta}) + \lambda\Psi(\boldsymbol{u}) + T\Phi(\boldsymbol{u}), \qquad (1)\\ &\text{subject to } \boldsymbol{u}_n \in \Delta_K \quad \forall n \in \mathbb{Q},\\ &\qquad\qquad u_{n,k} = y_{n,k} \quad \forall n \in \mathbb{S},\ \forall k \in \{1,\dots,K\}. \end{aligned}$$

where weighing factor λ and temperature scaling T are learnable optimized hyper-parameters, which we will estimate through the proposed unrolling strategy (Section 3.2),

and terms $\mathcal{L}$, Ψ and Φ are detailed in the following.

- The first term in objective function (1) is the negative log-likelihood of the feature vectors:

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{\theta}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} u_{n,k} \ln(\mathrm{p}(\boldsymbol{z}_n|\boldsymbol{\theta}_k)) \tag{2}$$

This general log-likelihood model fitting term is well known in the context of clustering methods [5, 21]. In fact, it generalizes the standard K-means clustering objective to arbitrary distributions[2]. It is well-known that minimizing (2) has an inherent bias towards class-balanced clustering [5, 21, 32, 33].

- The second term in (1) controls the partition complexity of the model, penalizing the number of non-empty clusters in the solution. It corresponds to the Shannon entropy of class distribution $(\pi_k)_{1\leq k\leq K}$, defined as:

$$\Psi(\boldsymbol{u}) = -\sum_{k=1}^{K} \pi_k \ln \pi_k, \tag{3}$$

where $\pi_k = \frac{1}{|\mathbb{Q}|}\sum_{n\in\mathbb{Q}} u_{n,k}$ is the proportion of query samples within class k, i.e., the empirical estimate of the marginal probability of class k.

Class-balance hyper-parameter (λ): The marginal entropy in (3) mitigates the class-balance bias of the log-likelihood clustering term in (2). It reaches its minimum for the extremely imbalanced solution in which all data samples are assigned to a single cluster, and its maximum for a perfectly balanced clustering. Hence, clearly, hyper-parameter λ controls the level of class balance in the solution. As depicted in Fig. 1, this hyper-parameter has a crucial effect on the performance, and its optimal value might vary by orders of magnitude from one dataset to another. This makes exhaustive grid searches for optimal λ over validation sets intractable computationally, which motivates learning this hyper-parameter through our unrolling strategy, as described in Section 3.2.

- The third term in (1) is an entropic barrier, enabling to soften assignments $u_{n,k}$, while imposing a non-negativity constraint on each of them. It is given by:

$$\Phi(\boldsymbol{u}) = \sum_{n=1}^{N}\sum_{k=1}^{K} u_{n,k} \ln u_{n,k}. \tag{4}$$

Temperature scaling hyper-parameter (T): Weighting factor T in (1) controls the trade-off between the clustering term and the entropic barrier in (4), i.e., the level of softness of assignments $u_{n,k}$. Therefore, this hyper-parameter has an important effect on performances and on the marginal class probabilities appearing in the class-balance term in (3), which motivates learning it. As described in Section 3.2, our unrolling algorithm enables to cope with this extra parameter.

Link to EM and other transductive few-shot methods: We examine solving problem (1) with an iterative block-coordinate descent algorithm (see Algorithm 1), which alternates two steps, one updating distribution parameters $\boldsymbol{\theta}_k^{(\ell)}$ and the other optimizing over class assignments $\boldsymbol{u}_n^{(\ell)}$, at each iteration ℓ. For the $\boldsymbol{u}$-step, and due to the nonconvexity of Ψ, we proceed with a Majorization-Minimization (MM) strategy to minimize a surrogate convex function with respect to $\boldsymbol{u}$ at each iteration. In addition, if $\mathrm{p}(\cdot|\boldsymbol{\theta}_k)$ is assumed to belong to the exponential family and $\boldsymbol{\theta}_k$ are its canonical parameters, the estimation w.r.t. $\boldsymbol{\theta}$, with $\boldsymbol{u}$ fixed, is a convex problem. We provide further details on this general iterative optimization scheme in Appendix A. It is worth noting that, when $T = 1$ and the data is modeled by the Dirichlet distribution, we recover the recent transductive few-shot learning algorithm in [33]. Interestingly, when $T = 1$ and $\lambda = |\mathbb{Q}|$, we recover the well-known Expectation-Maximization (EM) algorithm for estimating the parameters of mixture of distributions; see Proposition 1 in [33]. Therefore, Algorithm 1 could be viewed as a generalized EM (GEM), in which hyper-parameters λ and T control the class balance as well as prediction softness, and could be made learnable.

Examples of data models: While a broad range of distribution models in the exponential family could be adopted, we focus in this paper on two popular ones.

- The first one is the Gaussian distribution, which is commonly used in standard clustering and transductive few-shot-methods applied to vision-only models [32, 44, 48]. By assuming a Gaussian distribution with mean $\boldsymbol{\theta}_k$ and identity covariance matrix, the pdf $\mathrm{p}(\boldsymbol{z}_n|\boldsymbol{\theta}_k)$ reads as follows:

$$\mathrm{p}(\boldsymbol{z}_n|\boldsymbol{\theta}_k) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{z}_n - \boldsymbol{\theta}_k\|^2\right). \tag{5}$$

- The second one is the Dirichlet distribution, which has recently shown good modelling performance in the context of transductive few-shot for vision-language models such as CLIP [33]. For $\boldsymbol{z}_n = (z_{n,i})_{1\leq i\leq K}$ and $\boldsymbol{\theta}_k = (\theta_{k,i})_{1\leq i\leq K}$, the associated pdf is given by:

$$\mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k\right) = \frac{1}{\mathcal{B}(\boldsymbol{\theta}_k)} \prod_{i=1}^{K} z_{n,i}^{\theta_{k,i}-1} \, \mathbb{1}_{\boldsymbol{z}_n\in\Delta_K}, \tag{6}$$

where the normalization factor $\mathcal{B}(\boldsymbol{\theta}_k)$ is:

$$\mathcal{B}(\boldsymbol{\theta}_k) = \frac{\prod_{i=1}^{K}\Gamma(\theta_{k,i})}{\Gamma\left(\sum_{i=1}^{K}\theta_{k,i}\right)}, \tag{7}$$

and Γ denotes the Gamma function.

[2] K-means corresponds to choosing the multivariate Gaussian distribution, with identity covariance matrix, for parametric density $\mathrm{p}(\boldsymbol{z}_n|\boldsymbol{\theta}_k)$.

Algorithm 1 GEM based few-shot classification algorithm

Input: Compute $\boldsymbol{z}_n$ for the dataset samples, initialize $\boldsymbol{u}_n^{(0)}$ and $\boldsymbol{\theta}_k^{(0)}$, and fix the number of iterations L,
for $\ell = 0, 1, \ldots, L-1$ **do**
// Update Distribution parameters for each class using a given an estimation algorithm (denoted here by "DP_est")
$\boldsymbol{\theta}_k^{(\ell+1)} = \text{DP\_est}(\boldsymbol{u}_{\cdot,k}^{(\ell)}, \boldsymbol{\theta}_k^{(\ell)}), \quad \forall k \in \{1, \ldots, K\},$
// Update class proportions
$\pi_k^{(\ell+1)} = \frac{1}{|\mathbb{Q}|} \sum_{n \in \mathbb{Q}} u_{n,k}^{(\ell)}, \quad \forall k \in \{1, \ldots, K\},$
// Update assignment vectors for all query samples
$\boldsymbol{u}_n^{(\ell+1)} = \text{softmax}\left(\frac{1}{T}\left(\ln \text{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k^{(\ell+1)}\right) + \frac{\lambda}{|\mathbb{Q}|}\ln(\pi_k^{(\ell+1)})\right)_k\right), \quad \forall n \in \mathbb{Q}.$
end for

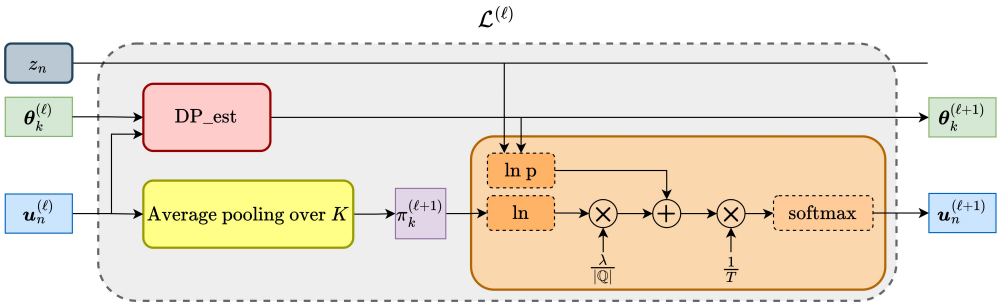

Figure 2. An overview of the unrolled GEM algorithm for a given iteration. Each iteration ℓ corresponds to a network layer $\mathcal{L}^{(\ell)}$. Each layer depends on the vector of hyperparameters $(\lambda^{(\ell)}, T^{(\ell)})$.

The necessary details related to feature representations in the cases of vision-only and vision-language models, distribution parameter estimation, and the resulting GEM algorithms for Gaussian and Dirichlet laws are provided in Appendices B and C, respectively.

3.2. UNrolled EM architecture (UNEM)

Overview: As discussed in Section 2, the choice of the optimal hyper-parameter, which controls the level of class balance, is a difficult task, often performed manually in the existing transductive few-shot classification methods. In addition to the class-balance parameter λ, Algorithm 1 involves an additional temperature parameter T, which makes hyper-parameter setting even more challenging. For this reason, we propose in this paper to resort to the unrolling (called also *learning to optimize*) paradigm [34], which enables to learn efficiently a set of optimized hyper-parameters. It is important to note that unrolling iterative optimization algorithms has found successful applications in diverse signal and image processing tasks [2, 7, 34]. However, to the best of our knowledge, this work is the first to leverage the unrolling paradigm for hyper-parameter optimization in a few-shot learning context. The main idea behind the learning-to-optimize paradigm is to map each iteration of a given optimization algorithm to a network layer, stack all layers together, and view the hyper-parameters to be optimized as the network's learnable parameters. More precisely, to unroll our generalized EM algorithm, the number of iterations L is used as the number of layers for the neural network architecture. Thus, each iteration $\ell \in \{0, 1, \ldots, L-1\}$ is associated with a tailored layer $\mathcal{L}^{(\ell)}$, which performs the update rules defined in Algorithm 1:

$$\left(\boldsymbol{\theta}_k^{(\ell+1)}, \pi_k^{(\ell+1)}, \boldsymbol{u}_n^{(\ell+1)}\right) = \mathcal{L}^{(\ell)}\left(\boldsymbol{\theta}_k^{(\ell)}, \pi_k^{(\ell)}, \boldsymbol{u}_n^{(\ell)}; \lambda^{(\ell)}, T^{(\ell)}\right) \tag{8}$$

where $(\lambda^{(\ell)}, T^{(\ell)})_{0 \le \ell \le L-1}$ is the vector of hyper-parameters to be learned. This leads to the UNrolled EM (UNEM) model shown in Fig. 2. The latter shows the inputs and outputs of a given layer $\mathcal{L}^{(l)}$ as well as its three main blocks for the update rules.

Learned hyper-parameters: Instead of a restricted number of handcrafted hyper-parameters (as often used in iterative algorithms), our unrolled algorithm offers more flexibility, enabling adaptation of the hyper-parameters along the processing workflow. This means that L vectors of hyper-parameters $(\lambda^{(\ell)}, T^{(\ell)})_{0 \le \ell \le L-1}$ could be learned and applied at each layer of the network. During the training of the unrolled model, we fulfill some design constraints, as described in more details in the following.
More specifically, to ensure the non-negativity of hyper-

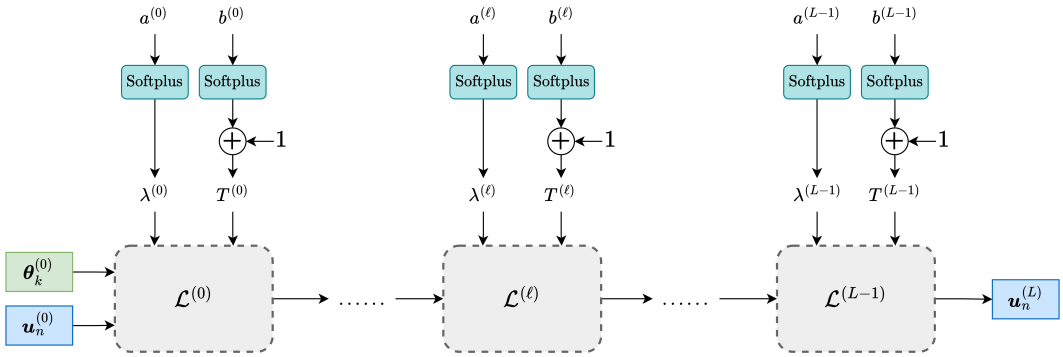

Figure 3. Overall architecture of the designed UNEM.

parameter $\lambda^{(\ell)}$, we propose to express it as a Softplus function, which could be seen as a smooth approximation of the RELU activation function, yielding:

$$\lambda^{(\ell)} = \text{Softplus}(a^{(\ell)}) = \log(1 + \exp(a^{(\ell)})) \tag{9}$$

where $a^{(\ell)}$ represents a learnable parameter for the unrolled architecture.

Regarding temperature scaling $T^{(\ell)}$, we observed in our experiments that imposing the non-negativity constraint alone might be a source of instability, resulting in values very close to zero and vanishing gradient issues during the training of the unrolled architecture. To circumvent this problem, we impose a lower bound equal to 1 on the temperature scaling:

$$T^{(\ell)} = 1 + \text{Softplus}(b^{(\ell)}) = 1 + \log(1 + \exp(b^{(\ell)})) \tag{10}$$

where $b^{(\ell)}$ denotes a parameter that needs to be learned during the training of the unrolled model.

Overall architecture and training approach: Based on the previous considerations, the overall architecture of UNEM can be summarized as the composition of the L layers $\mathcal{L}^{(L-1)} \circ \ldots \circ \mathcal{L}^{(0)}$. This architecture, depicted in Figure 3, illustrates: (i) the required inputs $\boldsymbol{\theta}_k^{(0)}$ and $\boldsymbol{u}_n^{(0)}$ as well as the feature vectors $\boldsymbol{z}_n$ (which are fed in all layers), (ii) the cascaded layers with their associated vector of parameters $\boldsymbol{w} = (\lambda^{(\ell)}, T^{(\ell)})_{0 \leq \ell \leq L-1}$ satisfying the aforementioned constraints, and (iii) the key output $\boldsymbol{u}_n^{(L)}$ representing the class assignment vectors.

The resulting neural network architecture is trained by minimizing a standard cross-entropy loss on a validation set:

$$\text{L}_\text{c}(\boldsymbol{w}) = \sum_{n \in \mathbb{Q}} \sum_{k=1}^{K} y_{n,k} \log(u_{n,k}^{(L)}). \tag{11}$$

4. Experiments

The effectiveness of the proposed approach is validated in both vision-only and vision-language transductive few-shot learning settings. We will designate our proposed unrolled methods by UNEM-Gaussian and UNEM-Dirichlet, respectively. After describing the experimental settings, this section is structured into two main sections discussing our experiments in both evaluation scenarios.

4.1. Experimental settings

Task generation: We adopt a realistic transductive few-shot evaluation protocol, which is in line with state-of-the-art evaluation protocols [29, 32, 33, 43, 48]. While K designates the total number of classes in the labeled support set $\mathbb{S}$, K_{eff} (with $K_{\text{eff}} \ll K$) denotes the number of effective classes present in the unlabeled query set $\mathbb{Q}$. The classes in the query set remain undisclosed during inference, while randomly selecting $|\mathbb{Q}|$ samples. On the other hand, the support set is built by uniformly selecting s images from each of the K classes. In this paper, the few-shot tasks are performed with (i) 5, 10, and 20 shots for vision-only models, and (ii) 4 shots for vision-language models. Moreover, in both of our evaluation scenarios, we used $K_{\text{eff}} = 5$ and $|\mathbb{Q}| = 75$.

Training specifications: For fair comparison and reproducibility purposes, the CLIP's pre-trained model has been directly used in the vision-language evaluation scenario, while for the vision-only scenario, the standard pre-trained ResNet-18 and WRN28-10 backbones have been fine-tuned. Their training is performed on the base classes set of each vision-only dataset, using cross-entropy loss with label smoothing set to 0.9 for 90 epochs. The learning rate is set to 0.1 and then decayed by a factor of 10.

On the other hand, the training of our unrolled architecture, composed of 10 layers (i.e., $L = 10$), is performed on several tasks sampled from the validation set of each dataset. This implies that the proposed solution is highly economical in terms of parameters, typically requiring only a few tenths, compared to standard neural network architectures. Let us recall that the validation set is often used to select the hyperparameters in recent state-of-the-art transductive methods as mentioned in Section 2. Specifically, for the

Method	Backbone	*mini*-ImageNet ($K = 20$)			*tiered*-ImageNet ($K = 160$)		
		5-shot	10-shot	20-shot	5-shot	10-shot	20-shot
Baseline [8]	ResNet-18	55.4	62.1	67.9	29.7	36.3	42.2
LR+ICI [50]		55.4	62.1	68.1	–	–	–
BD-CSPN [28]		49.8	54.6	56.5	11.4	11.0	11.7
PT-MAP [17]		25.7	27.2	28.4	5.2	6.0	6.6
LaplacianShot [58]		57.9	64.2	68.3	29.6	35.4	39.1
TIM [4]		**66.8**	69.9	70.8	29.3	28.7	27.8
α-TIM [48]		66.7	71.0	73.9	43.8	48.3	51.9
α-AM [24]		64.4	67.8	70.1	-	-	-
PADDLE [32]		62.9	73.5	79.8	45.4	61.4	70.6
UNEM-Gaussian		66.4	**75.6**	**80.4**	**52.3**	**65.7**	**73.2**
Baseline [8]	WRN28-10	59.0	65.7	72.1	31.9	39.0	45.6
LR+ICI [50]		58.8	65.7	72.0	–	–	–
BD-CSPN [28]		51.1	55.5	58.4	18.0	18.4	18.2
PT-MAP [17]		26.5	28.0	29.3	5.2	6.0	6.6
LaplacianShot [58]		61.0	66.8	71.0	31.4	37.3	41.5
TIM [4]		**72.1**	74.9	76.2	36.1	39.0	38.5
α-TIM [48]		71.5	75.2	78.3	45.8	51.4	55.2
α-AM [24]		68.2	71.3	73.3	-	-	-
PADDLE [32]		62.6	73.0	79.2	43.9	59.4	69.9
UNEM-Gaussian		71.6	**79.2**	**83.7**	**54.1**	**66.8**	**74.7**

Table 1. Comparison of the proposed UNEM-Gaussian with respect to state-of-the-art methods on *mini*-Imagenet and *tiered*-Imagenet. The metric is accuracy (in percentage). Results are averaged across 1,000 tasks. Results marked with '-' were intractable to obtain.

Method	CUB ($K = 50$)		
	5-shot	10-shot	20-shot
Baseline [8]	58.6	68.8	78.2
LR+ICI [50]	49.9	55.6	58.0
PT-MAP [17]	12.8	14.0	14.9
LaplacianShot [58]	58.8	66.5	71.0
TIM [4]	68.1	68.9	69.3
α-TIM [48]	74.3	79.3	83.6
α-AM [24]	66.2	68.9	69.8
PADDLE [32]	71.2	81.8	86.8
UNEM-Gaussian	**78.5**	**85.3**	**88.6**

Table 2. Comparison of the proposed UNEM-Gaussian with respect to state-of-the-art methods on CUB. The metric is accuracy (in percentage). Results are averaged across 1,000 tasks.

vision-only (resp. vision-language) models, the training of the unrolled architecture is carried out on 1,000 (resp. 100) tasks, while using an initial learning rate of 0.1 (resp. 0.5) with a decay factor of 0.5, a number of epochs equals to 80, and ADAM as optimizer. Our architecture is implemented in Pytorch (version 2.3.0) and run on NVIDIA QUADRO RTX8000 (with 48 GB of memory).

4.2. UNEM-Gaussian in vision-only few-shot setting

Let us recall that the Gaussian distribution is commonly used in standard clustering and transductive few-shot-methods applied to vision-only models [32, 44, 48]. For this reason, we will focus here on the proposed UNEM-Gaussian approach.

Datasets: The first UNEM-Gaussian architecture is evaluated on the following standard few-shot benchmark datasets: *mini*-ImageNet [49], *tiered*-ImageNet [40], CUB [14]. Mini-imagenet has 100 classes split into 64 base classes, 16 validation classes and 20 test classes. The tiered-imagenet has 608 classes instead, from which we follow a standard split with 351 for base training, 97 for validation and 160 for testing. For CUB, we followed the split proposed by [8] which consists of 100 base classes, 50 validation classes and 50 test classes. For each dataset, the feature vectors $\boldsymbol{z}_n$ are extracted using the fine-tuned backbones, while applying a scaling parameter T_z to the model's output as described in Appendix B. This parameter has also been learned through the unrolling approach.

Results: The UNEM-Gaussian architecture has been compared to its original version PADDLE [32] as well as several state-of-the-art methods. Tables 1 and 2 presents the results, averaged over 1,000 tasks with 5, 10, and 20 shots. Several observations can be made. First, the proposed UNEM-Gaussian outperforms the original PADDLE algorithm [32] and the other state-of-the-art methods. The achieved gains are much higher with fewer number of shots and reach 3.5% with *mini*-ImageNet, 6.9% with *tiered*-ImageNet, and 7.3% with CUB, while using ResNet-18 as backbone. Moreover, when WRN28-10 is used as backbone, it can be noticed that the performance of all methods has been improved, except

	Method	Food101	EuroSAT	DTD	OxfordPets	Flowers102	Caltech101	UCF101	FGVC Aircraft	Stanford Cars	SUN397	Average
Ind.	Tip-Adapter [54]	76.7	**72.5**	54.7	86.4	83.2	88.8	72.1	23.7	63.9	66.7	68.9
	CoOp [55]	76.3	63.2	52.2	86.2	81.0	87.7	67.0	22.2	61.3	63.4	66.1
Transd.	BDSCPN [28]	74.7	46.1	45.2	81.3	74.2	82.0	59.0	18.0	48.1	54.5	58.3
	Laplacian Shot [58]	76.6	53.0	52.6	88.4	85.5	86.8	67.0	22.2	60.4	63.8	65.6
	α-TIM [48]	66.1	46.1	45.3	87.1	79.1	83.3	59.4	20.4	53.4	53.4	59.4
	PADDLE [32]	71.8	45.9	50.0	84.7	82.3	81.9	63.7	21.3	56.1	60.6	61.6
	EM-Dirichlet [33]	88.7	50.8	62.6	92.5	91.3	90.1	76.1	24.9	73.5	80.9	73.1
	UNEM-Dirichlet	**91.4**	53.8	**65.3**	**96.0**	**95.6**	**93.4**	**78.5**	**30.4**	**80.0**	**88.5**	**77.3**

Table 3. Comparison of the proposed UNEM-Dirichlet with respect to the state-of-the-art methods on 10 different datasets. The metric is accuracy (in percentage). Results are averaged across 1,000 tasks.

for PADDLE, which shows results similar to those obtained with ResNet-18 pre-training. This can be explained by the fact that the class-balance parameter λ was set in [32] to $|\mathbb{Q}|$ (i.e. 75), which becomes sub-optimal with WRN28-10 features. Most importantly, by learning efficiently the hyperparameters, our UNEM-Gaussian achieves higher gain with respect to its original version, yielding an accuracy gain reaching up to 10% in 5-shot scenario, when WRN28-10 is used as backbone.

4.3. UNEM-Dirichlet for few-shot CLIP

Dirichlet distribution has recently demonstrated good modelling performance in the context of transductive few-shot for vision-language models such as CLIP [33]. Thus, we will focus in this second round of experiments on UNEM-Dirichlet approach.

Datasets: The second unrolled architecture, designated by UNEM-Dirichlet, is assessed on 10 benchmark datasets that are commonly used for CLIP scenario: Caltech101 [10], OxfordPets [36], StanfordCars [22], Flowers102 [35], Food101 [3], FGVCAircraft [31], SUN397 [52], DTD [9], EuroSAT [15], and UCF101 [45]. These datasets cover diverse classification challenges, from object recognition (Caltech101, Food101) to fine-grained tasks (StanfordCars, FGVCAircraft) and scene understanding (SUN397, EuroSAT). For each dataset, the vision-text feature vectors $\boldsymbol{z}_n$ are extracted using the CLIP's pre-trained model, while applying a temperature parameter T_z as described in Appendix C. This temperature parameter has also been learned through the unrolling approach.

Results: The proposed UNEM-Dirichlet has also been compared to its original EM-Dirichlet version [33] as well as different few-shot classification methods. Table 3 depicts the accuracy results evaluated over 1,000 tasks with 4-shots. Thus, it can be noticed that UNEM-Dirichlet outperforms state-of-the-art methods for most datasets. In particular, unrolling the recent EM-Dirichlet algorithm yields further improvement of about 4.2% in average. The achieved gain is more significant (reaching up to 7.5%) with more challenging datasets having a large number of classes like FGVC Aircraft, Stanford Cars, and SUN397. This is due to the inappropriate choice of the hyperparameter λ for these datasets. Indeed, the selected λ parameter was set to $\frac{K}{K_{\text{eff}}}|\mathbb{Q}|$ in EM-Dirichlet approach [33]. Thus, by computing the numerical values of the hyperparameter for each of the aforementioned datasets, it can be deduced from Figure 1 that the difference between the selected value (which is 3,000 for Stanford Cars) and the optimal one (which is around 5,500 for Stanford Cars) is more important and impacts significantly the accuracy performance. This problem is well addressed by resorting to the proposed unrolled model for hyperparameters optimization.

Appendix D includes additional results to illustrate, in both scenarios, the effects of the temperature parameter in this framework, and to show the benefits of learning variable hyper-parameters across the architecture layers.

5. Conclusion

In this paper, we focus on transductive few-shot methods grounded on a generalized form of the EM algorithm. These methods include a parameter to control class balance. The proposed GEM algorithm, applicable to any mixture of distributions, incorporates also a temperature scaling parameter. The optimization process is then unrolled into a neural network architecture, enabling efficient learning of the introduced hyper-parameters. The results demonstrate the effectiveness of the proposed approach on both vision-only and vision-language models. In future work, it would be valuable to explore the potential of unrolling techniques across a broader range of computer vision tasks, especially with the recent rise of foundational vision-langauge models.

References

[1] Christina Baek, Yiding Jiang, Aditi Raghunathan, and Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 19274–19289, 2022. 1

[2] C. Bertocchi, E. Chouzenoux, M.-C. Corbineau, J.-C. Pesquet, and M. Prato. Deep unfolding of a proximal interior point method for image restoration. *Inverse Problems, Special Issue on Variational Methods and Effective Algorithms for Imaging and Vision*, 36(3):1–27, 2020. 5

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461, Zurich, Switzerland, 2014. Springer. 8

[4] Malik Boudiaf, Ziko Imtiaz Masud, Jérôme Rony, José Dolz, Pablo Piantanida, and Ismail Ben Ayed. Transductive information maximization for few-shot learning. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 2445–2457, 2020. 2, 7

[5] Y. Boykov, H. N. Isack, C. Olsson, and I. Ben Ayed. Volumetric bias in segmentation and reconstruction: Secrets and solutions. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015. 4

[6] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Prompt learning with optimal transport for vision-language models. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 3

[7] Tianlong Chen, Xiaohan Chen, Wuyang Chen, and Zhangyang Wang. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research (JMLR)*, 23:1–59, 2022. 5

[8] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations (ICLR)*, pages 1–19, 2019. 7

[9] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 8

[10] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006. 8

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017. 1

[12] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. CLIP-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 32:581–595, 2023. 3

[13] Bharath Hariharan and Ross B. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3018–3027, 2017. 1

[14] Xiangteng He and Yuxin Peng. Fine-grained visual-textual representation learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2):520–531, 2020. 7

[15] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 8

[16] Shell Xu Hu, Da Li, Jan Stuhmer, 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9068–9077, 2022. 2

[17] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Leveraging the feature distribution in transfer-based few-shot learning. In *International Conference on Artificial Neural Networks*, pages 487–499, 2021. 2, 7

[18] Yunshi Huang, Fereshteh Shakeri, Jose Dolz, Malik Boudiaf, Houda Bahig, and Ismail Ben Ayed. LP++: A surprisingly strong linear probe for few-shot clip. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23773–23782, 2024. 2

[19] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11719–11727, 2019. 1

[20] Leonid Karlinsky, Joseph Shtok, Sivan Harary, Eli Schwartz, Amit Aides, and Rogerio Feris. Repmet: Representative-based metric learning for classification and few-shot object detection. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2019. 1

[21] M. Kearns, Y. Mansour, and A. Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 1997. 4

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 8

[23] Michalis Lazarou, Tania Stathaki, and Yannis Avrithis. Iterative label cleaning for transductive and semi-supervised few-shot learning. In *IEEE/CVF International Conference on Computer Vision*, pages 8751–8760, 2021. 2

[24] Michalis Lazarou, Yannis Avrithis, and Tania Stathaki. Adaptive manifold for imbalanced transductive few-shot learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, page 2297–2306, 2024. 2, 3, 7

[25] Gao Yu Lee, Tanmoy Dam, Daniel Puiu Poenar, Vu N. Duong, and Md Meftahul Ferdaus. HELA-VFA: A hellinger distance-attention-based feature aggregation network for few-shot classification. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2173–2183, 2024. 2

[26] Xiaoxu Li, Xiaochen Yang, Zhanyu Ma, and Jing-Hao Xue. Deep metric learning for few-shot image classification: A review of recent developments. *Pattern Recognition*, 138: 109381, 2023. 1

[27] 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 (ECCV)*, pages 22–539, 2020. 2

[28] Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. In *European Conference on Computer Vision (ECCV)*, pages 741–756, 2020. 2, 7, 8

[29] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations*, page 34677–34688, 2019. 2, 6

[30] Xu Luo, Hao Wu, Ji Zhang, Lianli Gao, Jing Xu, and Jingkuan Song. A closer look at few-shot classification again. In *International Conference on Machine Learning (ICML)*, pages 23103–23123, 2023. 2

[31] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. Technical report, Oxford University, 2013. 8

[32] Ségolène Martin, Malik Boudiaf, Emilie Chouzenoux, Jean-Christophe Pesquet, and Ismail Ben Ayed. Towards practical few-shot query sets: Transductive minimum description length inference. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, page 34677–34688, 2022. 2, 3, 4, 6, 7, 8

[33] Ségolène Martin, Yunshi Huang, Fereshteh Shakeri, Jean-Christophe Pesquet, and Ismail Ben Ayed. Transductive zero-shot and few-shot clip. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28816–28826, 2024. 1, 2, 3, 4, 6, 8

[34] Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021. 5

[35] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Fraphics & Image Processing*, pages 722–729. IEEE, 2008. 8

[36] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 8

[37] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5822–5830, 2018. 1

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and et al. Jack Clark. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 2

[39] Sachin Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, pages 1–11, 2017. 1

[40] 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 *International Conference on Learning Representations (ICLR)*, pages 1–15, 2018. 7

[41] Tyler Scott, Karl Ridgeway, and Michael C. Mozer. Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 76–85, 2018. 1

[42] Anuj Singh and Hadi Jamali-Rad. Transductive decoupled variational inference for few-shot classification. *Transactions on Machine Learning Research (TMLR)*, 2023. 2

[43] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 4080–4090, 2017. 6

[44] Yu Song and Changsheng Chen. MPPCANet: A feedforward learning strategy for few-shot image classification. *Pattern Recognition*, 113:107792, 2021. 4, 7

[45] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. Technical report, Center for Research in Computer Vision, University of Central Florida, 2012. arXiv:1212.0402. 8

[46] Ran Tao, Hao Chen, and Marios Savvides. Boosting transductive few-shot fine-tuning with margin-based uncertainty weighting and probability regularization. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15752–15761, 2023. 2, 3

[47] Long Tian, Jingyi Feng, Xiaoqiang Chai, Wenchao Chen, Liming Wang, Xiyang Liu, and Bo Chen. Prototypes-oriented transductive few-shot learning with conditional transport. In *International Conference on Computer Vision (ICCV)*, pages 16317–16326, 2023. 2

[48] Olivier Veilleux, Malik Boudiaf, Pablo Piantanida, and Ismail Ben Ayed. Realistic evaluation of transductive few-shot learning. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 9290–9302, 2021. 2, 3, 4, 6, 7, 8

[49] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 3637–3645, 2016. 1, 7

[50] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12833–12842, 2020. 7

[51] Yimu Wang, Yihan Wu, and Hongyang Zhang. Lost domain generalization is a natural consequence of lack of training domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 15689–15697, 2024. 1

[52] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. 8

[53] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6757–6767, 2023. 2, 3

[54] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. TIP-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision (ECCV)*, pages 493–510, 2022. 3, 8

[55] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16816–16825, 2022. 2, 3, 8

[56] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *International Conference on Computer Vision (ICCV)*, pages 15659–15669, 2023. 2

[57] Hao Zhu and Piotr Koniusz. Transductive few-shot learning with prototype-based label propagation by iterative graph refinement. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23996–24006, 2023. 2

[58] Imtiaz Masud Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian regularized few-shot learning. In *International Conference on Machine Learning (ICML)*, 2020. 7, 8

UNEM: UNrolled Generalized EM for Transductive Few-Shot Learning

Supplementary Material

A. Details on the minimization steps of the GEM optimization algorithm

The optimization algorithm alternates between a minimization step w.r.t. the distribution parameters and one w.r.t. the assignment variables. In the following, ℓ designates the current iteration.

• Minimization step w.r.t. the distribution parameter

For every $k \in \{1, \dots, K\}$, the first estimation step w.r.t. $\boldsymbol{\theta}_k$, with $\boldsymbol{u}_n = (u_{n,k}^{(\ell)})_{1 \leq k \leq K}$ given, is performed by considering the following optimization problem:

$$\underset{\boldsymbol{\theta}_k}{\text{minimize}} \quad -\sum_{n=1}^{N} u_{n,k}^{(\ell)} \ln \mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k\right), \tag{12}$$

For a pdf belonging to the exponential family, this optimization problem is a convex. For instance, in the case of a Gaussian distribution whose pdf is defined in (5), the negative log-likelihood term, designated by function F, reduces to

$$F(\boldsymbol{\theta}_k) = \frac{1}{2} \sum_{n=1}^{N} u_{n,k}^{(\ell)} \|\boldsymbol{z}_n - \boldsymbol{\theta}_k\|^2. \tag{13}$$

The minimization of the above function (13) w.r.t $\boldsymbol{\theta}_k$ results in an explicit form of the estimated distribution parameter $\boldsymbol{\theta}_k^{(\ell+1)}$ given by

$$\boldsymbol{\theta}_k^{(\ell+1)} = \frac{\sum_{n=1}^{N} u_{n,k}^{(\ell)} \boldsymbol{z}_n}{\sum_{n=1}^{N} u_{n,k}^{(\ell)}}. \tag{14}$$

In turn, in the case of Dirichlet distribution whose pdf is defined in (6), the negative log-likelihood term reads

$$F(\boldsymbol{\theta}_k) = \sum_{n=1}^{N} u_{n,k}^{(\ell)} \Bigg(-\sum_{i=1}^{K} (\theta_{k,i} - 1) \ln z_{n,i} + \sum_{i=1}^{K} \ln \Gamma(\theta_{k,i}) - \ln \Gamma\left(\sum_{i=1}^{K} \theta_{k,i}\right) \Bigg). \tag{15}$$

Unlike the Gaussian model, the minimization of Dirichlet negative log-likelihood (15) has no closed form solution. To circumvent this problem, we resort to the Majorization-Minorization (MM) strategy recently developed in [33]. Thus, the estimated distribution parameter $\boldsymbol{\theta}_k^{(\ell+1)}$ can be expressed as follows

$$\boldsymbol{\theta}_k^{(\ell+1)} = \mathrm{MM}(\boldsymbol{u}_{\cdot,k}^{(\ell)}, \boldsymbol{\theta}_k^{(\ell)}). \tag{16}$$

• Minimization step w.r.t. the assignment variable

For every $n \in \mathbb{Q}$, the second estimation step w.r.t. $\boldsymbol{u}_n$ is achieved by minimizing the objective function (1), while keeping the distribution parameter set to the estimated vector $\boldsymbol{\theta}_k^{(\ell+1)}$. However, since the partition complexity term Ψ is non convex, it is replaced by a linear tangent upper bound. More specifically, the following tangent inequality can be used:

$$\pi_k \ln \pi_k \geq \pi_k^{(\ell+1)} \ln \pi_k^{(\ell+1)} + (1 + \ln \pi_k^{(\ell+1)})(\pi_k - \pi_k^{(\ell+1)}) \tag{17}$$

Knowing that $\pi_k = \frac{1}{|\mathbb{Q}|} \sum_{n \in \mathbb{Q}} u_{n,k}$, the optimization problem (1) can be rewritten as follows

$$\underset{\boldsymbol{u}_n}{\text{minimize}} \quad G(\boldsymbol{u}_n) \tag{18}$$

with

$$\begin{aligned} G(\boldsymbol{u}_n) = &-\sum_{k=1}^{K} u_{n,k} \ln \mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k^{(\ell+1)}\right) \\ &- \lambda \sum_{k=1}^{K} \frac{(1 + \ln \pi_k^{(\ell+1)})}{|\mathbb{Q}|} (u_{n,k} - u_{n,k}^{(\ell)}) \\ &+ T \sum_{k=1}^{K} u_{n,k} \ln u_{n,k} + \gamma_n \left(\sum_{k=1}^{K} u_{n,k} - 1 \right) \end{aligned} \tag{19}$$

where γ_n is a Lagrange multiplier aiming to enforce the sum-to-one constraint. The nonnegativity constraint can be dropped since we will show next that it is satisfied by the minimizer of G subject to the sum-to-one constraint.

The above optimization problem is convex. By cancelling the derivative of the above objective function (19) w.r.t. $u_{n,k}$, it can be checked that

$$\ln u_{n,k} = -1 - \frac{\gamma_n}{T} + \frac{1}{T} \left(\ln \mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k^{(\ell+1)}\right) + \frac{\lambda}{|\mathbb{Q}|} (1 + \ln \pi_k^{(\ell+1)}) \right). \tag{20}$$

By applying the exponential function to (20) and determining the multiplier γ_n so that the sum-to-one constraint is satisfied, it can be deduced that the optimal class assignment vector $\boldsymbol{u}_n^{(\ell+1)}$ is obtained by applying the softmax function:

$$\boldsymbol{u}_n^{(\ell+1)} = \mathrm{softmax}\left(\frac{1}{T} \left(\ln \mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k^{(\ell+1)}\right) + \frac{\lambda}{|\mathbb{Q}|} \ln(\pi_k^{(\ell+1)}) \right)_k \right). \tag{21}$$

B. Generalized EM algorithm in the case of Gaussian distribution

B.1. Feature representation in vision-only few-shot-setting

Let us consider a few-shot scenario for vision-only models. Thus, for all dataset samples $\boldsymbol{x}_n$ with $n \in \{1, \ldots, N\}$, the feature vectors $\boldsymbol{z}_n$ are generated using a visual feature extractor $f^{(v)}$ as follows

$$\boldsymbol{z}_n = T_z f^{(v)}(\boldsymbol{x}_n) \tag{22}$$

where T_z is a positive scaling parameter.

B.2. Optimization algorithm

Using (13), (14), and (21), the proposed GEM algorithm reduces to Algorithm 2 in the case of a Gaussian distribution model.

Algorithm 2 GEM-Gaussian based few-shot classification algorithm

Input: Compute $\boldsymbol{z}_n$ for the dataset samples and, for all $k \in \{1, \ldots, K\}$, initialize $\boldsymbol{\theta}_k^{(0)}$ as the means computed on the support set, and $\pi_k^{(0)} = 1$

for $\ell = 0, 1, \ldots, L-1$ **do**

// Update assignment vectors for all query samples

$$\boldsymbol{u}_n^{(\ell+1)} = \text{softmax}\left(\frac{1}{T}\left(-\frac{1}{2}\|\boldsymbol{z}_n - \boldsymbol{\theta}_k^{(\ell)}\|^2 + \frac{\lambda}{|\mathbb{Q}|}\ln(\pi_k^{(\ell)})\right)_k\right)$$

// Update the mean parameter for each class

$$\boldsymbol{\theta}_k^{(\ell+1)} = \frac{\sum_{n=1}^N u_{n,k}^{(\ell+1)} \boldsymbol{z}_n}{\sum_{n=1}^N u_{n,k}^{(\ell+1)}}, \quad \forall k \in \{1, \ldots, K\},$$

// Update class proportions

$$\pi_k^{(\ell+1)} = \frac{1}{|\mathbb{Q}|}\sum_{n\in\mathbb{Q}} u_{n,k}^{(\ell+1)}, \quad \forall k \in \{1, \ldots, K\},$$

end for

C. Generalized EM algorithm in the case of Dirichlet distribution

C.1. Feature representation in few-shot CLIP

Our second few-shot scenario is devoted to vision-language models such as CLIP. Let us assume $f^{(v)}$ a vision-based feature extractor, and $f^{(l)}$ a language-based feature extractor. Thus, for a sample $\boldsymbol{x}_n$ with $n \in \{1, \ldots, N\}$ and a text prompt $\boldsymbol{t}_k$ of class $k \in \{1, \ldots, K\}$ (for example $\boldsymbol{t}_k =$ "a photo of a {class k}"), the visual and text features are given by $f^{(v)}(\boldsymbol{x}_n)$ and $f^{(l)}(\boldsymbol{t}_k)$, respectively. Then, the resulting feature embeddings of the data sample $\boldsymbol{x}_n$ is defined as its probability vector of belonging to class k:

$$\boldsymbol{z}_n = \text{softmax}\left\{T_z \cos\left(f^{(v)}(\boldsymbol{x}_n), f^{(l)}(\boldsymbol{t}_k)\right)_{1\le k\le K}\right\}, \tag{23}$$

where $T_z > 0$ is a temperature scaling parameter.

C.2. Optimization algorithm

Using (16) and (21), and in the case of a Dirichlet data distribution model, the proposed GEM algorithm yields Algorithm 3.

Algorithm 3 GEM-Dirichlet based few-shot classification algorithm

Input: Compute $\boldsymbol{z}_n$ for the dataset samples, initialize $\boldsymbol{u}_n^{(0)} = \boldsymbol{z}_n$, and $\boldsymbol{\theta}_k^{(0)} = \mathbf{1}_K$

for $\ell = 0, 1, \ldots, L-1$ **do**

// Update the Dirichlet parameter for each class

$$\boldsymbol{\theta}_k^{(\ell+1)} = \text{MM}(\boldsymbol{u}_{\cdot,k}^{(\ell)}, \boldsymbol{\theta}_k^{(\ell)}), \quad \forall k \in \{1, \ldots, K\},$$

// Update class proportions

$$\pi_k^{(\ell+1)} = \frac{1}{|\mathbb{Q}|}\sum_{n\in\mathbb{Q}} u_{n,k}^{(\ell)}, \quad \forall k \in \{1, \ldots, K\},$$

// Update assignment vectors for all query samples

$$\mathcal{L}_{n,k}^{(\ell)} = \sum_{i=1}^K (\theta_{k,i}^{(\ell+1)} - 1)\ln z_{n,i} - \sum_{i=1}^K \ln\Gamma(\theta_{k,i}^{(\ell+1)}) + \ln\Gamma\left(\sum_{i=1}^K \theta_{k,i}^{(\ell+1)}\right)$$

$$\boldsymbol{u}_n^{(\ell+1)} = \text{softmax}\left(\frac{1}{T}\left(\mathcal{L}_{n,k}^{(\ell)} + \frac{\lambda}{|\mathbb{Q}|}\ln(\pi_k^{(\ell+1)})\right)_k\right)$$

end for

D. Additional results

D.1. Ablation studies

In this part, we perform ablation studies to illustrate the effects of the introduced temperature scaling parameter and show the benefits of learning adaptive hyper-parameters across the unrolled architecture layers.

• Effects of temperature scaling

To perform this study, we compare our unrolled architectures (UNEM-Gaussian as well as UNEM-Dirichlet) in both cases: (i) without introducing the temperature scaling parameter (as considered in the original algorithms PADDLE [32] and EM-Dirichlet [33]); (ii) while incorporating the temperature scaling (as proposed in our GEM algorithm).

Tables 4 and 5 depict the accuracy results in vision-only few-shot setting. Thus, it can be noticed that including the temperature scaling yields an accuracy improvement, which may vary from 1% to 3%. Moreover, in the context of vision-language models whose accuracy results are shown

in Table 6, similar gains (reaching up to 3%), depending on the target donwstream dataset, are also achieved. This confirms again the advantage of incorporating the temperature scaling in our generalized algorithm.

• Fixed vs adaptive hyper-parameters across layers
One of the key advantages of unrolling algorithms is their flexibility in optimizing hyper-parameters, while allowing them to vary across the architecture layers. To show the potential of such hyper-parameter optimization approach, we propose to compare the proposed unrolled architectures (UNEM-Gaussian and UNEM-Dirichlet) in the following two cases: (i) the hyper-parameters are set fixed across the layers (as it is generally considered in original iterative algorithms), (ii) a set of hyper-parameters, adapted to the different layers, is learned.
Tables 7 and 8 provide the accuracy results for fixed and adaptive hyper-parameters optimization with vision-only models. It can be seen that learning adaptive hyper-parameters yields an accuracy gain of about 2-4% compared to the case when the hyper-parameters are kept fixed across layers. Similar comparisons are also performed with vision-language models as shown in Table 9. In this context, the improvement achieved by learning adaptive hyper-parameters often ranges from 1 to 2%.

D.2. Illustration and analysis of the learned hyperparameters

In this part, we propose to illustrate the variations of the learned hyper-parameters and analyze their orders-of-magnitude.

Illustration of the learned hyper-parameters
The evolutions of the learned hyper-parameters $\lambda^{(\ell)}$ and $T^{(\ell)}$ with respect to the layer index are illustrated in Figures 4 and 5 for some downstream image classification tasks. While Figure 4 shows that the learned hyper-parameters with CUB (ResNet18), *mini*-ImageNet (ResNet18), and *mini*-ImageNet (WRN28-10) have similar amplitudes, much different orders-of-magnitude are observed with vision-language models as shown in Figure 5 for some test datasets. Let us recall that the different learned hyper-parameters, for all datasets, are available at https://anonymous.4open.science/r/UNEM.

Analysis of the learned hyper-parameters
Different observations could be made from the previous illustrations. On the one hand, in the case of vision-only models, it can be seen that the learned hyper-parameters $\lambda^{(\ell)}$ appear quite similar. However, the evolution of $T^{(\ell)}$ values shows different behaviors. Moreover, it is important to note that the optimal hyper-parameters also depend on the pre-training model as observed with *mini*-ImageNet (ResNet18) and *mini*-ImageNet (WRN28-10). On the other hand, with vision-language models, it can be observed that both hyper-parameter values $\lambda^{(\ell)}$ and $T^{(\ell)}$ strongly depend on the target dataset. Indeed, unlike the vision-only models where the feature vectors have a fixed size (which is equal to the dimension of the pre-trained model's output), the feature vectors $\boldsymbol{z}_n$ in the context of few-shot CLIP have different sizes, depending on the number of classes of each target dataset. For instance, knowing that EuroSAT, Flowers102 and Stanford Cars have 10, 102, and 196 classes, respectively; it can be observed that the smallest (resp. largest) values of $\lambda^{(\ell)}$ are obtained with EuroSAT (resp. Stanford Cars). These results are expected since, by increasing the dimension of $\boldsymbol{z}_n$, the magnitude of the log-likelihood term may increase, and so, a higher value of $\lambda^{(\ell)}$ is needed to mitigate the class-balance bias.
This study shows the dependence of the introduced hyper-parameters on the target downstream dataset as well as the pre-training model, and confirms the importance of optimizing hyper-parameters in both evaluation scenarios.

Temperature scaling	Backbone	*mini*-ImageNet ($K = 20$)			*tiered*-ImageNet ($K = 160$)		
		5-shot	10-shot	20-shot	5-shot	10-shot	20-shot
×	ResNet-18	66.1	75.4	80.3	49.7	63.2	70.0
✓		**66.4**	**75.6**	**80.4**	**52.3**	**65.7**	**73.2**
×	WRN28-10	**71.9**	78.9	82.8	52.0	65.8	73.0
✓		71.6	**79.2**	**83.7**	**54.1**	**66.8**	**74.7**

Table 4. Effects of the temperature scaling on the accuracy performance of UNEM-Gaussian approach applied to *mini*-ImageNet and *tiered*-ImageNet datasets.

Temperature scaling	CUB ($K = 50$)		
	5-shot	10-shot	20-shot
×	78.1	85.2	88.6
✓	**78.5**	**85.3**	**88.6**

Table 5. Effects of the temperature scaling on the accuracy performance of UNEM-Gaussian approach applied to CUB dataset.

Temperature scaling	Food101	EuroSAT	DTD	OxfordPets	Flowers102	Caltech101	UCF101	FGVC Aircraft	Stanford Cars	SUN397
×	90.6	51.9	**65.4**	95.4	92.0	92.4	**79.1**	27.5	78.2	88.4
✓	**91.4**	**53.8**	65.3	**96.0**	**95.6**	**93.4**	78.5	**30.4**	**80.0**	**88.5**

Table 6. Effects of the temperature scaling on the accuracy performance of UNEM-Dirichlet approach applied to the vision-language models.

Params across the layers	Backbone	*mini*-ImageNet ($K = 20$)			*tiered*-ImageNet ($K = 160$)		
		5-shot	10-shot	20-shot	5-shot	10-shot	20-shot
Fixed	ResNet-18	62.5	72.5	78.0	49.8	63.6	70.4
Adaptive		**66.4**	**75.6**	**80.4**	**52.3**	**65.7**	**73.2**
Fixed	WRN28-10	68.7	77.0	82.0	51.6	64.6	72.1
Adaptive		**71.6**	**79.2**	**83.7**	**54.1**	**66.8**	**74.7**

Table 7. Fixed vs adaptive hyper-parameters setting in the UNEM-Gaussian approach, using *mini*-ImageNet and *tiered*-ImageNet datasets.

Params across layers	CUB ($K = 50$)		
	5-shot	10-shot	20-shot
Fixed	75.2	82.9	87.1
Adaptive	**78.5**	**85.3**	**88.6**

Table 8. Fixed vs adaptive hyper-parameters setting in the UNEM-Gaussian approach, using CUB dataset.

Params across layers	Food101	EuroSAT	DTD	OxfordPets	Flowers102	Caltech101	UCF101	FGVC Aircraft	Stanford Cars	SUN397
Fixed	89.6	52.2	64.8	95.3	95.3	92.3	**79.2**	**31.6**	78.0	87.6
Adaptive	**91.4**	**53.8**	**65.3**	**96.0**	**95.6**	**93.4**	78.5	30.4	**80.0**	**88.5**

Table 9. Fixed vs adaptive hyper-parameters setting in the UNEM-Dirichlet approach, using the vision-language models.

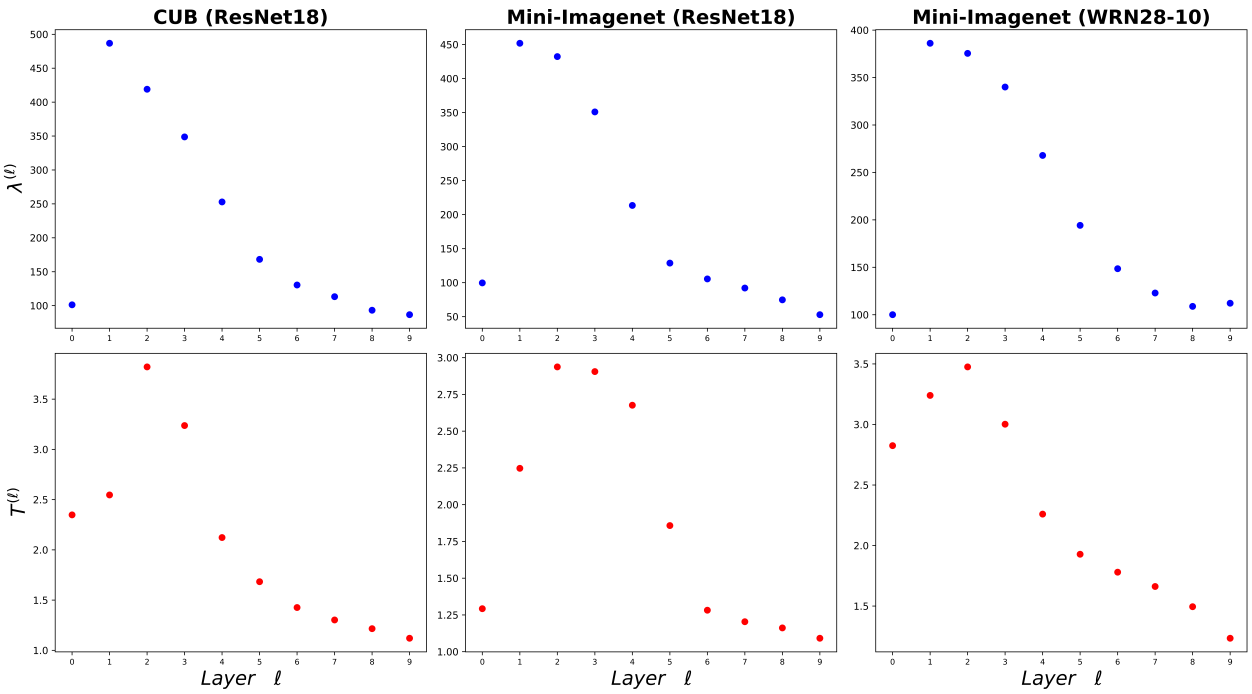

Figure 4. Illustration of the learned hyper-parameters $\lambda^{(\ell)}$ and $T^{(\ell)}$ across layers for CUB (with ResNet18 model), *mini*-ImageNet (with ResNet18 model) and *mini*-ImageNet (with WRN28-10 model).

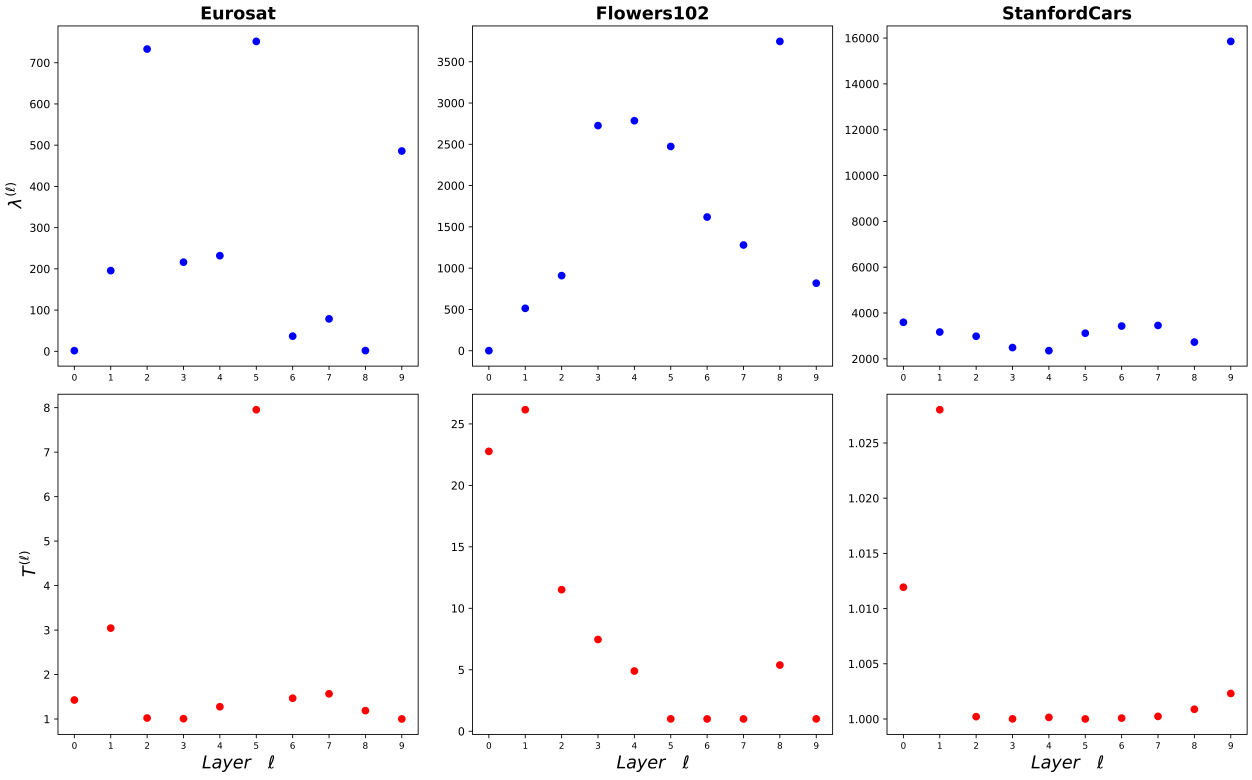

Figure 5. Illustration of the learned hyper-parameters $\lambda^{(\ell)}$ and $T^{(\ell)}$ across layers for some datasets with vision-language models.