Better Embedding and More Shots for Few-shot Learning

Ziqiu Chi[1,2] , **Zhe Wang**[1,2*] , **Mengping Yang**[1,2] , **Wei Guo**[1,2] and **Xinlei Xu**[1,2]
[1]Key Laboratory of Smart Manufacturing in Energy Chemical Process, East China University of Science and Technology, China
[2]School of Information Science and Engineering, East China University of Science and Technology, China
chiziqiu@mail.ecust.edu.cn, wangzhe@ecust.edu.cn,
{mengpingyang, wei_guo, Y20190072}@mail.ecust.edu.cn

Abstract

In few-shot learning, methods are enslaved to the scarce labeled data, resulting in suboptimal embedding. Recent studies learn the embedding network by other large-scale labeled data. However, the trained network may give rise to the distorted embedding of target data. We argue two respects are required for an unprecedented and promising solution. We call them Better Embedding and More Shots ($\mathcal{BEMS}$). Suppose we propose to extract embedding from the embedding network. $\mathcal{BE}$ maximizes the extraction of general representation and prevents over-fitting information. For this purpose, we introduce the topological relation for global reconstruction, avoiding excessive memorizing. $\mathcal{MS}$ maximizes the relevance between the reconstructed embedding and the target class space. In this respect, increasing the number of shots is a pivotal but intractable strategy. As a creative method, we derive the bound of information-theory-based loss function and implicitly achieve infinite shots with negligible cost. A substantial experimental analysis is carried out to demonstrate the state-of-the-art performance. Compared to the baseline, our method improves by up to 10%+. We also prove that $\mathcal{BEMS}$ is suitable for both standard pre-trained and meta-learning embedded networks.

1 Introduction

The outstanding successes of deep learning are partly accredited to the sheer amount of trainable samples. With the rapid development of deep learning, few-shot learning conceives a more challenging scenario, where only scarce labeled samples exist in each novel class. Under a standard few-shot learning setting, the embedding network is trained on substantial labeled data that sampled from base classes, *i.e.*, no intersection with novel classes. Further, the model can be fine-tuned on the few-shot labeled data of novel classes. Recent researchs [Tian *et al.*, 2020] [Hou and Sato, 2021] prove that embedding is the most vital aspect of few-shot image classification, which is also the focus of this paper. The embedding network training paradigm contains standard supervised pre-training [Chen *et al.*, 2019a] and meta-learning [Finn *et al.*, 2017]. Either way, we debate that two problems remain to be solved.

First, the embedding network is not well suited for novel classes as it is associated with base classes closely. While we can employ a few labeled samples to fine-tune the embedding in the novel-class space, there is a gamble of over-fitting. Additionally, the embedding network memorizes overmuch details specific to base classes, resulting in distorted novel-class embedding. Although the relevant art [Lee and Chung, 2021] is aimed at preventing over-fitting to the base classes via early-stage embedding reconstruction, the embedding is still not tailored for novel classes. With a more profound thought, we creatively reconstruct the topological relation instead of the instance similarity. It alleviates the excessive priority on the original feature space and pays more attention to global information. Furthermore, we put on the wings of information theory to make the reconstructed embedding more in line with the novel-class space. Nonetheless, we still face a tough nut, which leads us to the second problem.

Second, increasing the number of shots is pivotal but intractable [Cao *et al.*, 2019]. Explicit GANs-based data augmentation techniques [Zhang *et al.*, 2018] [Li *et al.*, 2020] occur as feasible solutions. Besides, embedding sampling approaches [Yang *et al.*, 2020] [Chi *et al.*, 2021] generate pseudo-labeled embedding in metric space. Unsatisfactorily, these remedies introduce much complexity. In this paper, we unprecedentedly introduce implicit data augmentation into few-shot learning, taking inspiration from ISDA [Wang *et al.*, 2021]. We abstractly achieve infinite augmentation with negligible extra computational cost, where the only alteration reflects in the derived upper bound of the loss function. In summary, our contributions are:

- This paper argues that two respects are required for novel-class-specific embedding learning. First, how to properly reconstruct pre-trained embedding. Second, how to make the reconstructed embedding fit the novel-class space better. Together, these two aspects constitute our Better Embedding and More Shots ($\mathcal{BEMS}$).
- This paper creatively considers topological reconstruction avoiding excessive memorizing, corresponding to the Better Embedding ($\mathcal{BE}$). In addition, we make the

[*] corresponding author

embedding more in accordance with the novel-class space through the derived upper bound of the loss function. This bound implicitly achieves infinite shots with negligible cost, corresponding to the More Shots ($\mathcal{MS}$). To the best of our knowledge, it is the first work to introduce implicit data augmentation into the few-shot learning.

- Substantial evaluations and ablation studies prove the promising performance. We also demonstrate that our method is appropriate for both standard pre-trained and meta-learning embedded networks.

2 Related Work

Both the embedding network and downstream module can obtain better embedding. The cross-entropy-based pre-trained methods are the most common for the embedding network. Other strategies, such as self-supervised learning [Chen *et al.*, 2021] and mixup [Mangla *et al.*, 2020], are available for the more general-purpose embedded feature. For the downstream module, one tends to improve the metric space of the embedding. BD-CSPN [Liu *et al.*, 2020] rectifies the prototype based on intra-class and inter-class biases. S2M2$_\mathrm{R}$ [Mangla *et al.*, 2020] adopts the manifold mixup for robust general-purpose representation. This paper focuses on the downstream module. We reveal that proper reconstruction is beneficial for better embedding.

Most methods explicitly accomplish more shots augmentation in sample or embedding space. MetaGAN [Zhang *et al.*, 2018] generates non-perfect samples to help the classifier identify much tighter decision boundaries based on generative adversarial networks. Similarly, AFHN [Li *et al.*, 2020] takes the few labeled samples as the conditional context to synthesize fake features. MVT [Park *et al.*, 2020] generates virtual embedding to boost the target space. However, it is still an explicit sampling process needing additional regularization. Instead of synthesizing image instances, TriNet [Chen *et al.*, 2019b] proposes to synthesize embedding directly. Distribution Calibration [Yang *et al.*, 2020] and Learning2Capture [Chi *et al.*, 2021] generate pseudo-labeled embedding based on the similarity relation. All of these methods introduce additional complex modules, while the modification in our method is only reflected in the loss function, which is more straightforward and more efficient.

3 Method

3.1 Notation

The few-shot problem is represented by the data in novel class $\mathcal{C}_n$. With $\boldsymbol{x}_i$ as the embedded D-dimension representation and y_i as its label, the consuetudinary naming, the N-way K-shot task sampled from $\mathcal{C}_n$ denotes the support set $\mathcal{S} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{\mathrm{N}\times\mathrm{K}}$, with N classes and each has K labeled samples. Corresponding with the $\mathcal{S}$, the unlabeled query set is defined as $\mathcal{Q} = \{\boldsymbol{x}_i, y_i\}_{i=\mathrm{N}\times\mathrm{K}+1}^{\mathrm{N}\times\mathrm{K}+\mathrm{T}}$, where T means the volume. In this paper, we utilize the vector and matrix forms flexibly. For example, we also use $(X \in \mathbb{R}^{\mathrm{T}\times D}, Y \in \mathbb{R}^{\mathrm{T}\times\mathrm{N}})$ to represent data pairs in $\mathcal{Q}$. To obtain embedding, an embedding network $f_\Phi(\cdot)$ is trained on a large-scale labeled dataset belonging to the base class $\mathcal{C}_b$, where $\mathcal{C}_b \cap \mathcal{C}_n = \emptyset$. Typically, the embedding network can be standard pre-trained [Wang *et al.*, 2019] or meta-learning model [Liu *et al.*, 2019].

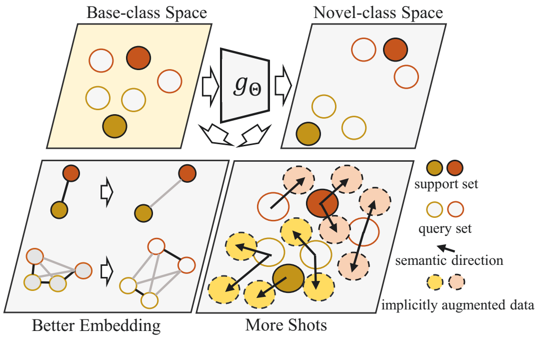

Figure 1: Overview of our method. Different colored circles indicate different categories of embedding. By and large, we reconstruct the embedding from $\mathcal{C}_b$ space to $\mathcal{C}_n$ space based on two aspects. The topological reconstruction leads to better embedding, and the implicit augmentation achieves more shots.

3.2 Overview

Figure 1 gives a general description of our method. Given the embedding $\boldsymbol{x}_i$ and our topological reconstruction network $g_\Theta(\cdot)$, we implement Better Embedding ($\mathcal{BE}$) based on topological reconstruction (Section 3.3), and More Shots ($\mathcal{MS}$) by implicit augmentation (Section 3.4). Finally, the trained network provides $\mathcal{C}_n$-specific embedding: $\boldsymbol{z}_i = g_\Theta(\boldsymbol{x}_i)$.

3.3 Better Embedding

We introduce that two steps lead to better embedding. Suppose we propose to extract the embedding by $f_\Phi(\cdot)$. The first step is to maximize extraction of general representation and minimize extraction of $\mathcal{C}_b$-specific information. As we have discussed in Section 2, one may utilize pre-training tricks [Chen *et al.*, 2021] to make $f_\Phi(\cdot)$ more general or stop the embedding reconstruction in time [Lee and Chung, 2021] to prevent $\mathcal{C}_b$-specific information. Instead of reconstructing the embedding directly, we rebuild the topological relation learned by $f_\Phi(\cdot)$. It makes the $g_\Theta(\cdot)$ pay more attention to the global information, avoiding over reconstruction. Formally, we have:

$$\mathcal{L}_{\mathcal{BE}}(X;\Theta) = \left\|X^T X - Z^T Z\right\|_F^2, \tag{1}$$

where $Z = g_\Theta(X)$ and F denotes the Frobenius norm. We gingerly strike their trade-off balance for $\mathcal{S}$ and $\mathcal{Q}$. Because $\mathcal{S}$ often acts as the precious supervised prototype [Liu *et al.*, 2020], and $\mathcal{Q}$ provides more samples. Specifically, we obtain better embedding by:

$$\mathcal{L}_{\mathcal{BE}} = \lambda_1 \mathcal{L}_{\mathcal{BE}}^{\mathcal{S}} + \lambda_2 \mathcal{L}_{\mathcal{BE}}^{\mathcal{Q}}, \tag{2}$$

where λ_1 and λ_2 balance the contribution between $\mathcal{S}$ and $\mathcal{Q}$. $\mathcal{L}_{\mathcal{BE}}$ achieves the purpose of the first step, *i.e.*, appropriate information extraction. Next, we move on to step two.

3.4 More Shots

In $\mathcal{BE}$, the reconstructed embedding is still stuck in the $\mathcal{C}_b$ space. Ideally, the second step is to maximize the relevance between the reconstructed embedding and the $\mathcal{C}_n$ space. This inspiration can be depicted as:

$$\max R(Z, \mathcal{C}_n), \tag{3}$$

where $R(\cdot, \cdot)$ represents the relevance. Nevertheless, scarce labeled data in $\mathcal{C}_n$ are not up to this challenge. Fortunately, we propose that distribution capture is a sound thought, where more or even infinite shots can be sampled. Once we have captured the class-conditional distribution, we can perform the following sampling:

$$\widetilde{\boldsymbol{z}}_i \sim \mathcal{N}(\boldsymbol{z}_i, \alpha \Sigma_{y_i}), \tag{4}$$

where Σ_{y_i} is the class-conditional covariance matrix, and α is a positive coefficient that controls the strength. Implementing such seemingly trivial sampling requires complex explicit methods, such as metric learning [Chi *et al.*, 2021] and GANs [Li *et al.*, 2020]. Nonetheless, this paper presents straightforward implicit data augmentation without introducing additional modules.

We primarily expand from $\mathcal{Q}$. It is critical to secure accurately estimated distribution for the unlabeled ones, as otherwise, it will cast a catastrophic impact. Consequently, we embrace the prevalent prototypical classifier to acquire pseudo distribution:

$$p_{ij} = \frac{d\left(\boldsymbol{z}_i, \boldsymbol{\mu}_j\right)}{\sum_k^{\mathrm{N}} d\left(\boldsymbol{z}_i, \boldsymbol{\mu}_k\right)}, \tag{5}$$

where $d(\cdot, \cdot)$ can be any distance function. The prototype $\boldsymbol{\mu}_j$ is calculated by the average of j-class $\mathcal{S}$, mathematically, $\boldsymbol{\mu}_j = \mathrm{E}_{\boldsymbol{z} \sim \mathcal{S}_j}[\boldsymbol{z}]$. According to these pseudo distributions, we can compute Σ_{y_i}, where $y_i = \arg\min_{j \in N} d(\boldsymbol{z}_i, \boldsymbol{\mu}_j)$. Correspondingly, we identify the meaningful semantic directions and proceed to sampling.

Assume we train the $g_\Theta(\cdot)$ with the weight matrix $\boldsymbol{W}$ and corresponding bias $\boldsymbol{b}$. With M times sampling, we maximize $R(Z, \mathcal{C}_n)$ via minimizing the $\mathcal{L}_R^{\mathcal{Q}}$:

$$\mathcal{L}_R^{\mathcal{Q}} = \frac{1}{\mathrm{T}} \sum_{i=1}^{\mathrm{T}} \sum_{j=1}^{\mathrm{N}} p_{ij} \frac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} -\log\left(\frac{e^{\boldsymbol{w}_j^T \widetilde{\boldsymbol{z}}_i^m + b_j}}{\sum_{k=1}^{\mathrm{N}} e^{\boldsymbol{w}_k^T \widetilde{\boldsymbol{z}}_i^m + b_k}}\right). \tag{6}$$

Based on it, the reconstructed embedding leans toward the pseudo $\mathcal{C}_n$ distributions. Covetously, when $\mathrm{M} \to \infty$, we rewrite Equation (6) in an expectation form:

$$\mathcal{L}_R^{\mathcal{Q}} = \frac{1}{\mathrm{T}} \sum_{i=1}^{\mathrm{T}} \sum_{j=1}^{\mathrm{N}} p_{ij} \mathrm{E}_{\widetilde{\boldsymbol{z}}_i}\left[\log\left(\sum_{k=1}^{\mathrm{N}} e^{\boldsymbol{v}_{kj}^T \widetilde{\boldsymbol{z}}_i + b_k - b_j}\right)\right], \tag{7}$$

where $\boldsymbol{v}_{kj} = \boldsymbol{w}_k - \boldsymbol{w}_j$. The above formula, however, is hard to apply. We show the possibility to derive an easy-to-compute upper bound, achieving this unattainable goal.

Theorem 1. *The upper bound of $\mathcal{L}_R^{\mathcal{Q}}$ is given by [Wang* et al.*, 2021]:*

$$\begin{aligned} \mathcal{L}_R^{\mathcal{Q}} &\leq \frac{1}{\mathrm{T}} \sum_{i=1}^{\mathrm{T}} \sum_{j=1}^{\mathrm{N}} p_{ij} \log\left(\sum_{k=1}^{\mathrm{N}} e^{\boldsymbol{v}_{kj}^T \boldsymbol{z}_i + b_k - b_j + \frac{\alpha}{2} \boldsymbol{v}_{kj}^T \Sigma_{y_i} \boldsymbol{v}_{kj}}\right) \\ &\triangleq \mathcal{L}_{\mathcal{MS}}^{\mathcal{Q}}. \end{aligned}$$

Proof. According to Jensen's inequality $\mathrm{E}[\log X] \leq \log \mathrm{E}[X]$, we derive:

$$\mathcal{L}_R^{\mathcal{Q}} \leq \frac{1}{\mathrm{T}} \sum_{i=1}^{\mathrm{T}} \sum_{j=1}^{\mathrm{N}} p_{ij} \log\left(\sum_{k=1}^{\mathrm{N}} \mathrm{E}_{\widetilde{\boldsymbol{z}}_i}\left[e^{\boldsymbol{v}_{kj}^T \widetilde{\boldsymbol{z}}_i + b_k - b_j}\right]\right). \tag{8}$$

According to the moment-generating function $\mathrm{E}\left[e^{tX}\right] = e^{t\mu + \frac{1}{2}\sigma^2 t^2}$, where $X \sim \mathcal{N}(\mu, \sigma^2)$, we have:

$$\mathrm{E}_{\widetilde{\boldsymbol{z}}_i}\left[e^{\boldsymbol{v}_{kj}^T \widetilde{\boldsymbol{z}}_i + b_k - b_j}\right] = e^{\boldsymbol{v}_{kj}^T \boldsymbol{z}_i + b_k - b_j + \boldsymbol{v}_{kj}^T \Sigma_{y_i} \boldsymbol{v}_{kj}}. \tag{9}$$

Thus, the upper bound of $\mathcal{L}_R^{\mathcal{Q}}$ is derived:

$$\mathcal{L}_R^{\mathcal{Q}} \leq \frac{1}{\mathrm{T}} \sum_{i=1}^{\mathrm{T}} \sum_{j=1}^{\mathrm{N}} p_{ij} \log\left(\sum_{k=1}^{\mathrm{N}} e^{\boldsymbol{v}_{kj}^T \boldsymbol{z}_i + b_k - b_j + \frac{\alpha}{2} \boldsymbol{v}_{kj}^T \Sigma_{y_i} \boldsymbol{v}_{kj}}\right), \tag{10}$$

where α is the positive hyper-parameter. Finally, we derive the final loss function $\mathcal{L}_{\mathcal{MS}}^{\mathcal{Q}}$. □

So far, we have implicitly carried out infinite sampling based on the unlabeled $\mathcal{Q}$. In a similar spirit, we apply the upper bound of cross-entropy loss function for the labeled $\mathcal{S}$.

Theorem 2. *Similar to Theorem 1, we derive the upper bound of the cross-entropy loss [Wang* et al.*, 2021]. For $\boldsymbol{z}_i \in \mathcal{S}$, we have:*

$$\begin{aligned} &\mathcal{L}_R^{\mathcal{S}} \leq \frac{1}{\mathrm{T}} \sum_{i=1}^{\mathrm{T}} \log\left(\sum_{k=1}^{\mathrm{N}} e^{\boldsymbol{v}_{ky_i}^T \boldsymbol{z}_i + b_k - b_j + \frac{\alpha}{2} \boldsymbol{v}_{ky_i}^T \Sigma_{y_i} \boldsymbol{v}_{ky_i}}\right) \\ &\triangleq \mathcal{L}_{\mathcal{MS}}^{\mathcal{S}}. \end{aligned} \tag{11}$$

Proof. Similar to the proof of Theorem 1, we have the following derivation based on Jensen's inequality and moment-generating function.

$$\begin{aligned} \mathcal{L}_R^{\mathcal{S}} &= \frac{1}{\mathrm{T}} \sum_{i=1}^{\mathrm{T}} \mathrm{E}_{\widetilde{\boldsymbol{z}}_i}\left[\log\left(\sum_{k=1}^{\mathrm{N}} e^{\boldsymbol{v}_{ky_i}^T \widetilde{\boldsymbol{z}}_i + b_k - b_j}\right)\right] \\ &\leq \frac{1}{\mathrm{T}} \sum_{i=1}^{\mathrm{T}} \log\left(\sum_{k=1}^{\mathrm{N}} \mathrm{E}_{\widetilde{\boldsymbol{z}}_i}\left[e^{\boldsymbol{v}_{ky_i}^T \widetilde{\boldsymbol{z}}_i + b_k - b_j}\right]\right) \\ &= \frac{1}{\mathrm{T}} \sum_{i=1}^{\mathrm{T}} \log\left(\sum_{k=1}^{\mathrm{N}} e^{\boldsymbol{v}_{ky_i}^T \boldsymbol{z}_i + b_k - b_j + \frac{\alpha}{2} \boldsymbol{v}_{ky_i}^T \Sigma_{y_i} \boldsymbol{v}_{ky_i}}\right). \end{aligned}$$

□

3.5 Overall Objective Function

In the mass, we minimize the overall objective given by:

$$\mathcal{L}_{\mathcal{BEMS}} = \underbrace{\lambda_1 \mathcal{L}_{\mathcal{BE}}^{\mathcal{S}} + \lambda_2 \mathcal{L}_{\mathcal{BE}}^{\mathcal{Q}}}_{\mathcal{L}_{\mathcal{BE}}} + \underbrace{\mathcal{L}_{\mathcal{MS}}^{\mathcal{S}} + \mathcal{L}_{\mathcal{MS}}^{\mathcal{Q}}}_{\mathcal{L}_{\mathcal{MS}}}. \tag{12}$$

When $\mathrm{K} = 1$, $\mathcal{L}_{\mathcal{MS}}^{\mathcal{S}}$ is reduced to $\mathcal{L}_R^{\mathcal{S}}$. In brief, the first two terms work for $\mathcal{BE}$ and the last two terms work for $\mathcal{MS}$. They provide the $\mathcal{C}_n$-specific embedding together.

4 Experiments

4.1 Experimental Settings

Datasets. We perform evaluations on three popular datasets. The *mini*ImageNet [Vinyals *et al.*, 2016] and *tiered*ImageNet [Ren *et al.*, 2018] are the subsets of ImageNet. CUB-200-2011 [Wah *et al.*, 2011] is a fine-grained bird classification dataset. All images are resized to 84×84. More details are depicted in Table 1.

Evaluation Protocol. We report the average accuracy and 95% confidence interval on random sampled 600 tasks. Each task contains 15 query samples of each class, *i.e.*, $\mathrm{T} = 15 \times \mathrm{N}$. For reported results, the number in **bold** means the best performance and <u>underline</u> means the second. All our re-implemented algorithms, as indicated by the superscript†, adopt the unified trained embedding network for fairer comparisons.

4.2 Implementation Details

Embedding Networks. We train our embedding networks using the standard cross-entropy loss on $\mathcal{C}_b$ based on three backbones: ConvNet, ResNet-18, and WideResNet. ConvNet comprises four blocks. Each block contains a 64-filter 3×3 convolution, a batch normalization layer, a ReLU activation and a 2×2 max-pooling layer. ResNet-18 we used is a standard 18-layer residual network that removed the first two down-sampling layers. WideResNet is the wide residual network with 28 convolutional layers and 10 widening factors. For WideResNet training, we set the label-smoothing parameter as 0.1. We use SGD optimizer and 128 mini-batch sizes. Referring to [Ziko *et al.*, 2020], we use early stopping by prototypical classifier on the validation set. For ConvNet and ResNet-18 training, we adopt the source code provided by [Wang *et al.*, 2019]. In addition, we extend our applicability to meta-learning in Section 4.7. This section trains all re-implemented methods with Adam optimizer and an initial learning rate of 0.001. We cut the learning rate in half every 10,000 and 25,000 episodes for *mini*ImageNet and *tiered*ImageNet, respectively.

Reconstruction Training. We use 2-layer fully connected layers with ReLU function and a 0.5 dropout as our reconstruction module. The first layer reduces the dimension to half the input dimension, and the second layer restores the dimension. We conduct 200 training iterations. For optimizer, we use Adam with 0.001 learning rate and 0.01 weight decay.

Preprocessing. The output of the penultimate layer of the embedding network is extracted as the embedding feature. The centering and L2-normalization are used as preprocessing tools. For centering, we first calculate the average value of the embedded features in $\mathcal{C}_b$: $\boldsymbol{\mu} = \mathrm{E}_{\boldsymbol{x} \sim \mathcal{C}_b}[\boldsymbol{x}]$. Then $\boldsymbol{x} \leftarrow \boldsymbol{x} - \boldsymbol{\mu}$, where $\boldsymbol{x} \in \mathcal{C}_n$. For L2-normalization, we compute: $\boldsymbol{x} \leftarrow \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}$.

Hyper-parameters. The hyper-parameters are tuned by validation set. We set $\alpha = t \times \frac{\mathrm{iter}}{\mathrm{ITER}}$, where iter and ITER are the current and total iterations, respectively. We tune t in $\{0.01, 0.1, 0.25, 1, 10\}$. For λ_1 and λ_2, we tune them in $\{0.01, 0.02, 0.1, 0.5, 1\}$.

Dataset	Classes	Images	Train/Val/Test
*mini*ImageNet	100	60000	64/16/20
*tiered*ImageNet	608	779165	351/97/160
CUB	200	11788	100/50/50

Table 1: Details of datasets.

4.3 Improvement by our Method

We utilize the prototypical classifier as our baseline to investigate the improvement by our method. Concretely, it makes inference of $\mathcal{Q}$ by matching the nearest prototype, where Euclidean distance is adopted. Results of three backbones and two datasets are reported in Table 2. All backbones yield satisfactory results. The most remarkable performance is the 10%+ accuracy gain at most on 1-shot evaluation. Comparatively, 5-shot evaluation also gets a maximum 5.11% improvement. In addition, the performance gap between 1-shot and 5-shot results is significantly narrowed, which will be further proved in Section 4.8.

4.4 Comparison with Relevant Methods

We also report the performance of relevant methods in Table 2. (1) In comparison with the explicit data augmentation methods: Distribution Calibration, Learning2Capture, TriNet, MVT, and AFHN, our $\mathcal{BEMS}$ perform a significant improvement in a lighter augmentation manner. On all datasets and backbones, we averagely lead by 6.81% and 3.01% on 1-shot and 5-shot scenarios, respectively. (2) In comparison with the embedding adaptation methods: BD-CSPN and ESFR, we surpass them by 2.18% and 0.97% averagely on 1-shot and 5-shot scenarios, respectively, because our approach pays more attention to the topological structure and alleviates the over-fitting risk on $\mathcal{C}_b$. ESFR avoids the over-memorizing of $\mathcal{C}_b$, which is similar to our motivation. Differently, our $\mathcal{MS}$ further brings the reconstructed embedding closer to $\mathcal{C}_n$. (3) In comparison with the information theory based AWGIM, our $\mathcal{BEMS}$ leads by as much as 12.26% on *mini*ImageNet and 12.74% on *tiered*ImageNet. AWGIM maximizes the mutual information between generated weights and $\mathcal{S}$ as well as $\mathcal{Q}$ to retain information within the task, while our utilization is essential for better embedding.

4.5 Cross-domain Evaluation

We conduct further evaluations on CUB and cross-domain scenario, *i.e.*, *mini*ImageNet → CUB. Results are shown in Table 3. We also compare the most relevant algorithms, and our method still maintains the top performance. In cross-domain problems, our $\mathcal{BEMS}$ is second only to Learning2Capture, which uses explicit data augmentation.

4.6 Significance of Better Embedding and More Shots

First, we take our core components apart for ablation in Figure 2(*Left*). In this evaluation, we fix $t = 10$. We analyze the following alterations:

- *Baseline* denotes we directly make predictions through pre-trained embedding and prototypical classifier.

Methods	Backbone	*mini*ImageNet 1-shot	*mini*ImageNet 5-shot	*tiered*ImageNet 1-shot	*tiered*ImageNet 5-shot
MVT [Park *et al.*, 2020]	ConvNet	-	67.67 ± 0.70	-	-
LaplacianShot [Ziko *et al.*, 2020]		55.70 ± 0.85	68.04 ± 0.65	**57.04** ± **0.99**	71.37 ± 0.68
Learning2Capture† [Chi *et al.*, 2021]		51.18 ± 0.85	66.29 ± 0.69	51.60 ± 0.97	66.60 ± 0.79
BD-CSPN† [Liu *et al.*, 2020]		52.35 ± 0.87	68.51 ± 0.64	55.15 ± 0.90	71.87 ± 0.71
Prototypical Classifier†	ConvNet	50.70 ± 0.79	66.15 ± 0.70	50.73 ± 0.79	69.51 ± 0.70
+ESFR† [Lee and Chung, 2021]		52.92 ± 0.89	68.91 ± 0.71	55.86 ± 0.98	71.72 ± 0.77
+$\mathcal{BEMS}$ **(Ours)**		**56.11** ± **1.05**	**71.26** ± **0.74**	56.87 ± 1.16	**72.68** ± **0.74**
TriNet [Chen *et al.*, 2019b]	ResNet-18	58.12 ± 1.37	76.92 ± 0.69		
AFHN [Li *et al.*, 2020]		62.38 ± 0.72	78.16 ± 0.56		
Baseline [Chen *et al.*, 2019a]		51.75 ± 0.80	74.27 ± 0.63	-	-
Baseline++ [Chen *et al.*, 2019a]		51.87 ± 0.77	75.68 ± 0.63	-	-
LaplacianShot [Ziko *et al.*, 2020]		72.11 ± 0.19	82.31 ± 0.14	78.98 ± 0.21	86.39 ± 0.16
Learning2Capture† [Chi *et al.*, 2021]		67.38 ± 0.97	81.04 ± 0.61	76.55 ± 0.95	85.26 ± 0.63
BD-CSPN† [Liu *et al.*, 2020]		69.81 ± 0.95	82.32 ± 0.59	78.13 ± 0.94	86.88 ± 0.60
Prototypical Classifier†	ResNet-18	63.82 ± 0.82	79.71 ± 0.59	69.53 ± 0.89	85.32 ± 0.57
+ESFR† [Lee and Chung, 2021]		71.92 ± 0.92	82.27 ± 0.57	78.26 ± 0.98	85.83 ± 0.68
+$\mathcal{BEMS}$ **(Ours)**		**73.63** ± **1.08**	**82.76** ± **0.64**	**80.36** ± **0.99**	**87.04** ± **0.61**
Distribution Calibration [Yang *et al.*, 2020]	WideResNet	68.57 ± 0.55	82.30 ± 0.34	78.19 ± 0.25	**89.90** ± **0.41**
AWGIM [Guo and Cheung, 2020]		63.12 ± 0.08	78.40 ± 0.11	67.69 ± 0.11	82.82 ± 0.13
LaplacianShot [Ziko *et al.*, 2020]		74.86 ± 0.19	84.13 ± 0.14	80.18 ± 0.21	87.56 ± 0.15
Learning2Capture† [Chi *et al.*, 2021]		68.65 ± 0.92	81.92 ± 0.60	75.09 ± 0.96	86.14 ± 0.64
BD-CSPN† [Liu *et al.*, 2020]		72.55 ± 0.91	84.02 ± 0.55	79.56 ± 0.94	88.36 ± 0.59
Prototypical Classifier†	WideResNet	65.95 ± 0.90	81.76 ± 0.56	71.16 ± 0.89	86.32 ± 0.56
+ESFR† [Lee and Chung, 2021]		73.06 ± 0.91	82.80 ± 0.55	79.87 ± 0.94	87.14 ± 0.64
+$\mathcal{BEMS}$ **(Ours)**		**75.38** ± **1.03**	**84.25** ± **0.53**	**80.43** ± **1.04**	88.16 ± 0.58

Table 2: Comparison with our baseline and relevant approaches.

Methods	CUB 1-shot	CUB 5-shot	*mini* → CUB
TriNet [Chen *et al.*, 2019b]	69.61	84.10	-
baseline [Chen *et al.*, 2019a]	65.51	82.85	64.80
baseline++ [Chen *et al.*, 2019a]	67.02	83.58	62.04
LaplacianShot [Ziko *et al.*, 2020]	80.96	88.68	66.33
Learning2Capture† [Chi *et al.*, 2021]	76.53	87.50	**69.06**
BD-CSPN† [Liu *et al.*, 2020]	78.70	88.74	65.99
Prototypical Classifier†	70.31	86.44	65.85
+ESFR† [Lee and Chung, 2021]	79.94	88.24	65.00
+$\mathcal{BEMS}$	**82.74**	**89.12**	67.34

Table 3: The classification accuracy (%) on CUB and *mini* → CUB (5-shot). ResNet-18 is used as the backbone.

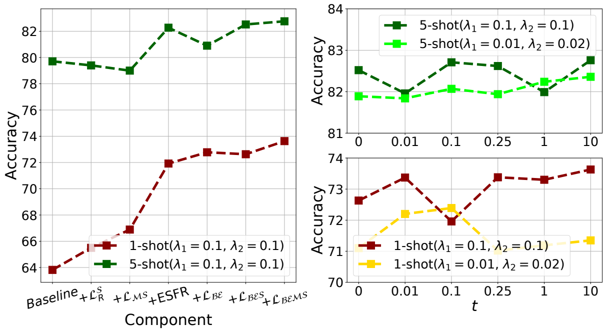

Figure 2: *mini*ImageNet ablation based on ResNet-18. (*Left*) The effectiveness of each component. (*Right*) The influence of α in $\mathcal{MS}$.

- $+\mathcal{L}_R^S$ denotes we only optimize $g_\Theta(\cdot)$ by $\mathcal{L}_R^S$;
- $+\mathcal{BE}$ and $+\mathcal{MS}$ denote we use the $\mathcal{L}_{\mathcal{BE}}$ and $\mathcal{L}_{\mathcal{MS}}$ terms in Equation (12), respectively;
- +ESFR denotes we take it as a contrast to $\mathcal{BE}$;
- $+\mathcal{BES}$ denotes the loss function without $\mathcal{MS}$: $\mathcal{L}_{\mathcal{BES}} = \lambda_1 \mathcal{L}_{\mathcal{BE}}^S + \lambda_2 \mathcal{L}_{\mathcal{BE}}^Q + \mathcal{L}_R^S + \mathcal{L}_R^Q$;
- $+\mathcal{L}_{\mathcal{BEMS}}$ denotes the complete version.

As we have mentioned in Section 4.3, our $\mathcal{BEMS}$ narrows the accuracy gap between different shot numbers, which is more intuitive in the line chart. The proposed $\mathcal{BE}$ substantially reconstructs better embedding. The proposed $\mathcal{MS}$ implicitly provides infinite data volume, blurring the shot disparity and boosting the performance. Taking ESFR as a reference for reconstruction problems, $\mathcal{BE}$ performs better on the 1-shot setting. In addition, ESFR further supports our $\mathcal{MS}$ perspective that embedding should get close to $\mathcal{C}_n$ space on the basis of $\mathcal{C}_b$ information reconstructing. Although $\mathcal{MS}$ shows a slightly negative effect on the 5-shot scenario, it improves significantly on the 1-shot setting. Finally, the $\mathcal{BEMS}$ consistently achieves the highest accuracy.

Second, we report the sensitivity of t in Figure 2(*Right*). In this evaluation, we fix the $\lambda_1 = \lambda_2 = 0.1$ and $\lambda_1 = 0.01, \lambda_2 = 0.02$. We observe that $\mathcal{MS}$ works positively in most t values. The overall trends show that the parameter sensitivity of $\mathcal{MS}$ is affected by $\mathcal{BE}$, which is reflected in the different effects of t on performance under different λ_1 and λ_2. This is because $\mathcal{BE}$ and $\mathcal{MS}$ have a latent antagonistic relationship. $\mathcal{MS}$ makes embedding tend to $\mathcal{C}_n$ space, which interferes with the reconstruction in $\mathcal{BE}$ to some extent.

Methods	*mini*ImageNet	*tiered*ImageNet
ProtoNet† [Snell *et al.*, 2017]	50.70	50.56
$+\mathcal{BEMS}$	**54.10**	**52.30**
TPN† [Liu *et al.*, 2019]	53.80	56.42
$+\mathcal{BEMS}$	**55.24**	**56.95**

Table 4: Meta-learning methods based on ConvNet. The 1-shot performances are reported.

Methods		*mini*ImageNet		*tiered*ImageNet	
		1-shot	5-shot	1-shot	5-shot
w/o	$\mathcal{L}^{\mathcal{Q}}_{\mathcal{MS}}$	73.33	81.98	**79.48**	86.89
	$\mathcal{L}^{\mathcal{Q}}_{\mathcal{BE}}$	60.15	80.72	62.89	86.56
	$\mathcal{Q}$	64.79	64.79	70.81	85.69
w/o	$\mathcal{L}^{\mathcal{S}}_{\mathcal{MS}}$	72.79	79.96	79.18	85.16
	$\mathcal{L}^{\mathcal{S}}_{\mathcal{BE}}$	54.94	28.64	60.22	29.06
	$\mathcal{S}$	72.56	79.38	78.72	85.16
Prototypical Classifier		63.82	79.71	69.53	85.32
FeatRec		64.54	78.21	69.35	83.96
$\mathcal{BEMS}$		**73.63**	**82.76**	79.44	**87.04**

Table 5: More detailed ablation studies based on ResNet-18

4.7 Meta-learning Embedding

As an extension study, we explore the enhancement effect of $\mathcal{BEMS}$ on meta-learning-based embedding. We select two representative meta-learning methods, ProtoNet [Snell *et al.*, 2017] and TPN [Liu *et al.*, 2019], for evaluations. Concretely, we use $\mathcal{BEMS}$ to boost the test phase of two algorithms, making the meta-knowledge transfer to the $\mathcal{C}_n$ better. Results in Table 4 strongly prove that our method also significantly improves meta-learning embedding.

4.8 Ablation Study

Based on the overall loss function, Equation (12), we conduct more detailed ablations in Table 5. We fix $\lambda_1 = \lambda_2 = 0.1$ and $t = 10$ in this evaluation. (1) Considering the relevant methods [Park *et al.*, 2020] [Lee and Chung, 2021], we also try the feature reconstruction. Accordingly, *FeatRec* denotes we change our topological reconstruction to $\|X - g_\Theta(X)\|_F^2$. The results show feature reconstruction plays a negative role in our approach, except for the 1-shot *mini*ImageNet. (2) *w/o* denotes we drop the corresponding component in the loss function. Six results report the $\mathcal{S}$ and $\mathcal{Q}$ ablations. We observe that the reconstructions of both $\mathcal{S}$ and $\mathcal{Q}$ are indispensable, especially for $\mathcal{S}$. The $\mathcal{MS}$ of $\mathcal{S}$ and $\mathcal{Q}$ have a steady performance improvement. When the $\mathcal{S}$ component is entirely absent, there is a certain negative impact on performance. When we drop all $\mathcal{Q}$ components, the performance degradation is more significant.

Parameter Sensitivity Analysis. We conduct the parameter sensitivity analysis in Figure 3. We observe higher λ_1 and λ_2 lead to better performances, where this phenomenon is more sensitive to λ_2. Sensitivity analysis shows that our method is stable and effective when the $\mathcal{L}_{\mathcal{BE}}$ occupies a high proportion in the overall loss function.

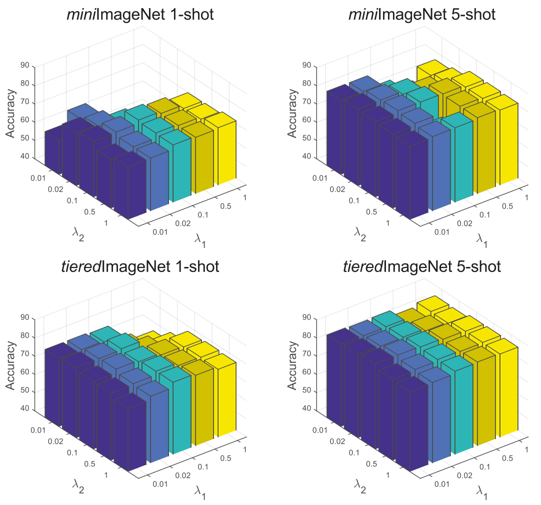

Figure 3: Parameter sensitivity analysis based on ResNet-18.

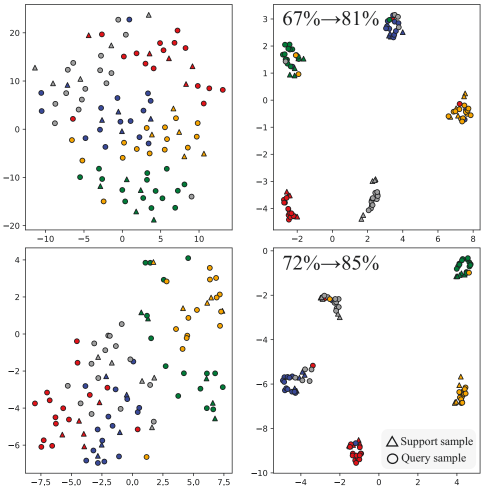

Figure 4: 5-way 5-shot T-SNE visualization on *mini*ImageNet. The ResNet-18 is utilized as the backbone.

Visualization. We use T-SNE to visualize the original and reconstructed embedding in Figure 4. The original embedding is distorted, which is reflected in poor inter-class and intra-class relations. Exhilaratingly, our $\mathcal{BEMS}$ solves the problem well, *i.e.*, sharper boundaries. Notably, the unlabeled $\mathcal{Q}$ is brilliantly reconstructed under the guidance of the implicit augmentation.

5 Conclusion

In this paper, we first reconstruct the topological relation of embedding, and then pioneer implicit more shots augmentation. We show significant improvement in different evaluations. In ablation studies, we discuss why $\mathcal{BEMS}$ works. Scarce labeled data is the fundamental problem of few-shot learning. Our approach implicitly alleviates this problem while introducing only negligible complexity, which is very promising.

Acknowledgments

This work is supported by Shanghai Science and Technology Program "Distributed and generative few-shot algorithm and theory research" under Grant No. 20511100600, Natural Science Foundation of China under Grant No. 62076094, and Shanghai Science and Technology Program "Federated based cross-domain and cross-task incremental learning" under Grant No. 21511100800.

References

[Cao *et al.*, 2019] Tianshi Cao, Marc T Law, and Sanja Fidler. A theoretical analysis of the number of shots in few-shot learning. In *ICLR*, 2019.

[Chen *et al.*, 2019a] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

[Chen *et al.*, 2019b] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Multi-level semantic feature augmentation for one-shot learning. *TIP*, 28(9):4594–4605, 2019.

[Chen *et al.*, 2021] Da Chen, Yuefeng Chen, Yuhong Li, Feng Mao, Yuan He, and Hui Xue. Self-supervised learning for few-shot image classification. In *ICASSP*, pages 1745–1749, 2021.

[Chi *et al.*, 2021] Ziqiu Chi, Zhe Wang, Mengping Yang, Dongdong Li, and Wenli Du. Learning to capture the query distribution for few-shot learning. *TCSVT*, 2021.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.

[Guo and Cheung, 2020] Yiluan Guo and Ngai-Man Cheung. Attentive weights generation for few shot learning via information maximization. In *CVPR*, pages 13499–13508, 2020.

[Hou and Sato, 2021] Mingcheng Hou and Issei Sato. A closer look at prototype classifier for few-shot image classification. *arXiv preprint arXiv:2110.05076*, 2021.

[Lee and Chung, 2021] Dong Hoon Lee and Sae-Young Chung. Unsupervised embedding adaptation via early-stage feature reconstruction for few-shot classification. In *ICML*, pages 6098–6108, 2021.

[Li *et al.*, 2020] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. Adversarial feature hallucination networks for few-shot learning. In *CVPR*, pages 13470–13479, 2020.

[Liu *et al.*, 2019] 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 *ICLR*, 2019.

[Liu *et al.*, 2020] Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. In *ECCV*, pages 741–756, 2020.

[Mangla *et al.*, 2020] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *WACV*, pages 2218–2227, 2020.

[Park *et al.*, 2020] Seong-Jin Park, Seungju Han, Ji-won Baek, Insoo Kim, Juhwan Song, Hae Beom Lee, Jae-Joon Han, and Sung Ju Hwang. Meta variance transfer: Learning to augment from the others. In *ICML*, pages 7510–7520, 2020.

[Ren *et al.*, 2018] 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. *arXiv preprint arXiv:1803.00676*, 2018.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 30, 2017.

[Tian *et al.*, 2020] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*, pages 266–282, 2020.

[Vinyals *et al.*, 2016] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, volume 29, 2016.

[Wah *et al.*, 2011] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[Wang *et al.*, 2019] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[Wang *et al.*, 2021] Yulin Wang, Gao Huang, Shiji Song, Xuran Pan, Yitong Xia, and Cheng Wu. Regularizing deep networks with semantic data augmentation. *TPAMI*, 2021.

[Yang *et al.*, 2020] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *ICLR*, 2020.

[Zhang *et al.*, 2018] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *NeurIPS*, volume 31, 2018.

[Ziko *et al.*, 2020] Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian regularized few-shot learning. In *ICML*, pages 11660–11670, 2020.