IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 26, 2021, accepted April 13, 2021, date of publication April 20, 2021, date of current version April 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3074525

Augmenting Few-Shot Learning With Supervised Contrastive Learning

TAEMIN LEE, (Graduate Student Member, IEEE), AND SUNGJOO YOO, (Senior Member, IEEE)
Department of Computer Science and Engineering, Seoul National University, Seoul 08826, South Korea
Corresponding author: Taemin Lee (taemin.lee@snu.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) under Grant 2016M3A7B4909604.

ABSTRACT Few-shot learning deals with a small amount of data which incurs insufficient performance with conventional cross-entropy loss. We propose a pretraining approach for few-shot learning scenarios. That is, considering that the feature extractor quality is a critical factor in few-shot learning, we augment the feature extractor using a contrastive learning technique. It is reported that supervised contrastive learning applied to base class training in transductive few-shot training pipeline leads to improved results, outperforming the state-of-the-art methods on Mini-ImageNet and CUB. Furthermore, our experiment shows that a much larger dataset is needed to retain few-shot classification accuracy when domain-shift degradation exists, and if our method is applied, the need for a large dataset is eliminated. The accuracy gain can be translated to a runtime reduction of 3.87× in a resource-constrained environment.

INDEX TERMS Few-shot learning, contrastive learning, information maximization.

I. INTRODUCTION

The impressive results of deep learning-based methods are mainly achieved using a large amount of labeled data [16], [39]. However, massive image labeling is labor-intensive, and a balanced dataset is challenging to obtain. By contrast, humans show excellent generalization performance from only one or a few examples, bringing motivation to the field of few-shot learning [17], [22], [23], [44]. Likewise, the aim of few-shot learning is to predict unlabeled data based on the observation of a few labeled data (e.g., one or five examples per class).

Compared with traditional inductive few-shot learning, two settings are introduced to address the low data count. A semi-supervised few-shot setting [21], [33] assumes that the model can utilize information from additional unlabeled data. Better accuracy can be obtained by increased amount of unlabeled data. A transductive few-shot setting [27], [31] accords that the model can access all the test data at once instead of one by one in the inference procedure. In the scope of this study is confined to the transductive few-shot setting as it is simple, yet effective [5] to achieve state-of-the-art result [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Rosalia Maglietta.

Conventional few-shot learning algorithms implement a two-stage training pipeline. *Base classes*, which are used only in the first stage of training, are large, separate classes for training the feature extractor, usually with conventional cross-entropy loss. In the consecutive stage, *novel classes*, which are a disjoint set of base classes, are learning targets with a few training examples per class. The first training stage attempts to learn general, transferable visual features from the base classes, whereas the main few-shot algorithms are implemented in the second stage to predict images from the novel classes.

As a feature extractor's performance is empirically related to the final classification accuracy, it is reasonable to use various augmentation techniques during the first training stage. These techniques [4], [43], [54], [56] are motivated by large-scale image classification tasks, such as ImageNet. Supervised contrastive learning [14] is proposed to replace cross-entropy loss by applying self-supervised representation learning with label information. It is examined that supervised contrastive loss instead of simple cross-entropy loss in the first training stage improves the final classification accuracy by a large margin, especially when the dataset is not large.

Assume that a few-shot learning task is running on an edge device, considering the scale of the problem. However, as the cost of the step is high—nearly a hundred epochs of training

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

the entire dataset—the base class training step is presumably processed by the server. The cost of supervised contrastive learning is an additional pretraining step at the base class training, which is amortized and processed efficiently by servers. With the accuracy gain obtained by the supervised contrastive learning, one can optimize the runtime latency of the algorithm with a simple method such as an early stopping.

Few-shot learning is associated with self-supervised representation learning, as noted in [8]. Both approaches have a similar goal: training the model with few or no data labels. Self-supervised representation learning is a method of unsupervised learning, which aims to learn from a dataset with no annotation. Instead, it learns using pretext information, such as the relative location of image patches or the rotation classification of images. Contrastive learning is a form of self-supervised representation learning that trains the model to classify similar (positive) samples and dissimilar (negative) samples in the embedding space. As supervised contrastive learning is an extension of contrastive learning, it implies the gain obtained in our experiment.

We observe that the feature extractor trained on a large, general dataset (i.e., Tiered-ImageNet) performs better than the feature extractor trained on a small, task-specific dataset (i.e., CUB) when evaluating a few-shot learning task. In our experiment, supervised contrastive learning improves the few-shot classification accuracy to the extent that even when trained on a small, task-specific dataset, it performs better than the feature extractor trained on a large, general dataset. Therefore, it is data-efficient and obtains superior performance without resort to a large dataset.

In summary, the contributions of our study are as follows:

- We propose using supervised contrastive learning in the first stage of few-shot learning to boost classification accuracy on the Mini-ImageNet and CUB datasets. Our method is referred as SPTA following the name of combined methods.
- We study the domain-shift setting, in which the feature extractor is trained on a different dataset, and the few-shot algorithm is evaluated on a fine-grained classification dataset, showing that a large dataset (i.e., Tiered-ImageNet) is needed to overcome domain-shift degradation. However, when supervised contrastive learning is applied to the CUB dataset, the case without a large dataset can score higher than the case with a large dataset.

II. RELATED WORK

A. FEW-SHOT LEARNING

There are many approaches to address few-shot learning tasks with less amount of data. Gradient descent-based approaches [7], [29], [32] learn how to re-adjust a model with a few gradient descent iterations to deal with a few-shot learning task. The model-agnostic meta-learning (MAML) [7] method trains the model with many tasks to generalize a new task efficiently. Reptile [29] is a first-order gradient-based meta-learning algorithm that trains the initialization of model parameters. [32] proposed a long short-term memory (LSTM)-based meta-learner whose states represent the update of the model parameter.

Metric-learning-based approaches [15], [37], [44], [47] learn distance metrics between a support set (training data of the target task) and a query set (test data of the target task) better by reforming feature embedding. [15] introduced Siamese convolutional neural networks that learn generic visual features on the character recognition task. The matching network [44] architecture is inspired by a memory-augmented neural network and generates a weighted nearest neighbor classifier using the distance between samples. Prototypical networks [37] utilize episodic training and assign each class to each prototype in the representation space to predict new data based on the distance metric to each prototype. [47] proposed using an additional data sample generator, which is trained with meta-learning methods, to augment the model training.

Transductive few-shot methods [1], [5], [12], [26], [31], [50], [59] assume that the model simultaneously accesses all the query set. A transductive episodic-wise adaptive metric (TEAM) [31] defined the optimization process as a standard semi-definite programming problem to train a generalizable classifier. A distribution propagation graph network (DPGN) [50] proposed utilizing both the distribution-level and instance-level relations by designing a dual complete graph network consisting of a point graph and a distribution graph. [5] proposed transductive fine-tuning, which pursues outputs with a peaked posterior or low Shannon entropy, and a hardness metric to deliver a standardized evaluation protocol. [26] proposed the prototype rectification, which lowers the class prototype's intra-class bias and cross-class bias and verifies the method theoretically. A synthetic information bottleneck (SIB) [12] introduced an empirical Bayes approach and a two-network architecture consisting of a synthetic gradient network and an initialization network to perform the synthetic gradient descent. LaplacianShot [59] implemented a constrained graph clustering method that attaches the query samples to the nearest prototype, and a pairwise Laplacian term advocates similar samples to output the same label. Transductive information maximization (TIM) [1] maximizes the mutual information between the query features and the predicted query label by minimizing the conditional entropy and maximizing the marginal entropy, and the alternating direction optimizer enables faster convergence than the typical gradient descent optimizer.

B. CONTRASTIVE LEARNING

Contrastive learning [2], [10], [13], [36], [41], [48] is a self-supervised learning method inspired by noise contrastive estimation [9], [28] or N-pair losses [38]. [48] proposed the use of a non-parametric softmax classifier to increase the instance-level distance on a 128-dimensional unit sphere after the CNN extracts a feature vector of the image. [13] improved contrastive predictive coding to implement a pretraining stage with a feature extractor and a context network to predict the

spatial location of the image patches. Deep InfoMax [10] proposed an approach for training an encoder that maximizes the mutual information between the input data and output features. [41] aimed to maximize the mutual information between different views of the same image by pulling views of the same scene together and pushing views of different scenes apart. Time-contrastive networks (TCN) [36] proposed learning from multi-view video by pulling the anchor and positive images together while pushing negative images apart. SimCLR [2] implemented two data augmentation paths and a learnable nonlinear transformation to train an encoder with a large batch by pulling the feature embedding from the same image. Supervised contrastive learning [14] is an extension of conventional contrastive learning that has been modified for supervised classification.

III. METHODOLOGY

This section introduces the formulation of the few-shot learning task and the proposed idea in detail.

A. PROBLEM DEFINITION

Given a labeled base dataset $\mathbf{D}_{\text{base}} := \{(\mathbf{x}_i, \mathbf{y}_i), \mathbf{y}_i \in \mathbf{C}_{\text{base}}\}$ and a novel dataset $\mathbf{D}_{\text{novel}} := \{(\mathbf{x}_i, \mathbf{y}_i), \mathbf{y}_i \in \mathbf{C}_{\text{novel}}\}$ where $\mathbf{C}_{\text{base}} \cap \mathbf{C}_{\text{novel}} = \emptyset$, the goal of a few-shot learning task is to train a visual model using the base dataset $\mathbf{D}_{\text{base}}$ and to generalize to the novel dataset $\mathbf{D}_{\text{novel}}$ which has a few training images per class. At inference, each few-shot learning task episode consists of a support set and a query set sampled from the novel dataset. The support set (S) is labeled and includes K samples per class with N classes (N-way K-shot setting), whereas the query set (Q) includes T samples per class with the same N classes without data labels. The goal is to map the samples in the query set to the desired label using the information gained from the support set. In the transductive setting, the model can access the entire dataset including the query set (i.e., $N \times K + N \times T$ samples) at once instead of one by one (i.e., $N \times K + 1$ samples each) in the traditional inductive setting.

B. EXAMINING A FEW-SHOT LEARNING METHOD

In this study, we examine the transductive information maximization (TIM) few-shot learning algorithm [1]. First, a feature extractor transforms an input image into embedded features. TIM maximizes the modified mutual information between the query image's feature and the query label by updating the soft-classifier's trainable weights. To maximize the information, TIM minimizes the conditional entropy and maximizes the marginal entropy. Minimizing conditional entropy aims to make confident predictions by modeling the cluster assumption, which implies that the classification criterion should not be present in the dense regions of the unlabeled features. Maximizing marginal entropy pushes the marginal distribution of labels to be uniform, which attempts to avoid the solution of outputting only one class. Together with the conventional cross-entropy loss, the TIM loss is defined as follows:

$$L^{tim} = -\frac{\lambda}{|S|}\sum_{i\in S}\sum_{n=1}^{N} y_{in}\log p_{in} - \mathcal{I}$$

$$\mathcal{I} := -\sum_{n=1}^{N}\hat{p_n}\log\hat{p_n} + \frac{\alpha}{|Q|}\sum_{i\in Q}\sum_{n=1}^{N} p_{in}\log p_{in}$$

where p_{in} is the posterior distribution over the labels given the features and $\hat{p_n}$ is the marginal distribution over the query labels.

Given the loss objective, two optimization methods are presented [1]. One is a conventional gradient descent (TIM-GD) method that minimizes the loss objective through mini-batch sampling. Although TIM-GD shows the best results, it is two orders of magnitude slower than inductive methods, which leads to the second method called the alternating direction method (TIM-ADM), which divides the problem into two more manageable subproblems and optimizes them iteratively. TIM-ADM shows competitive results compared to TIM-GD while being one order of magnitude faster. In both methods, sufficiently large number of iterations were required to converge to the best results. Typical values for the number of iterations for TIM-GD and TIM-ADM were 1,000 and 150, respectively.

C. AUGMENTING FEW-SHOT LEARNING WITH SUPERVISED CONTRASTIVE LEARNING

The quality of a feature extractor is one of the main challenges in improving a few-shot learning algorithm because it is directly related to the quality of the feature embeddings. Supervised contrastive learning [14] is an extension of self-supervised representation learning; it has a similar two-stage training procedure, as shown in Figure 1. The first stage prepares two copies of an input image and preprocesses them. An encoder network then transforms the images into normalized embedding, and an additional projection network transforms the embedding into a low-dimensional embedding. Supervised contrastive loss is computed on the low-dimensional embedding by attracting positive samples, which have the same class label or are from the same copied images, and by repelling the negative samples. The supervised contrastive loss is defined as follows:

$$L^{sup} = -\sum_{i\in I}\frac{1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp(z_i\cdot z_p/\tau)}{\sum_{a\in A(i)}\exp(z_i\cdot z_a/\tau)}$$

where z_l is the low-dimensional embedding, τ is a temperature parameter, $A(i) \equiv I \setminus \{i\}$, i is an anchor index, and $P(i) \equiv \{p \in A(i) : \bar{y}_p = \bar{y}_i\}$ is the set of indices of all positives except the anchor. The inner product operation on the embedding space measures the similarity between two feature embeddings. The loss is minimized when an anchor's feature embedding is similar to all the positive's feature embeddings and is different from all the negative's feature embeddings. The loss is generalized from the conventional

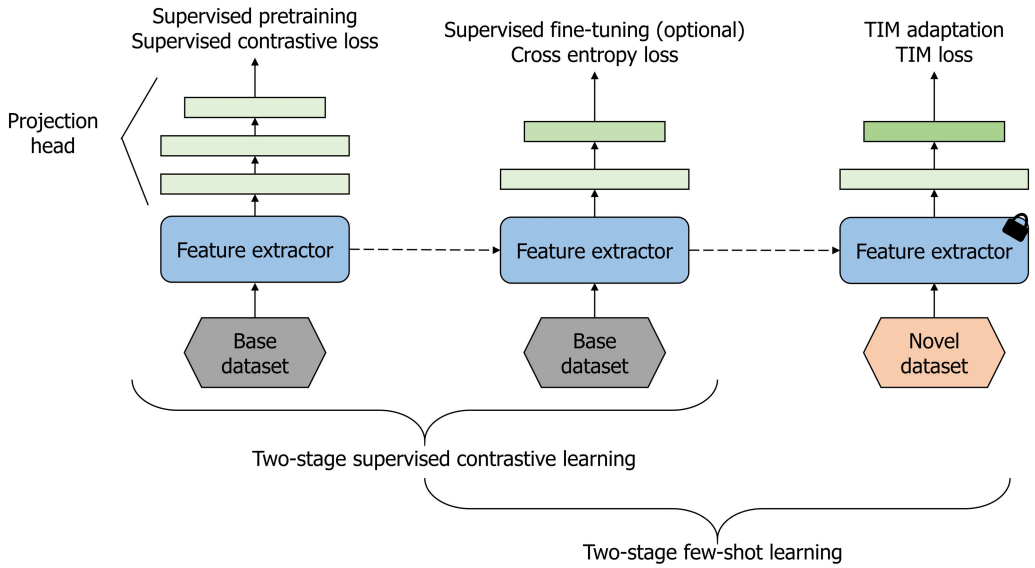

FIGURE 1. The proposed pretraining approach for few-shot learning consists of a multi-stage training process. The first stage of supervised contrastive learning uses supervised contrastive loss and projection head with the base dataset to learn visual representations. The second stage of supervised contrastive learning uses conventional cross-entropy loss with the base dataset to fine-tune the feature extractor. This two-stage supervised contrastive learning comprises the first stage of few-shot learning. The second stage of few-shot learning uses TIM [1] loss and the feature extractor fixed with the novel dataset to perform TIM adaptation. If the supervised fine-tuning becomes standard supervised training and the supervised contrastive pretraining is skipped, then the entire pipeline is the same as in the baseline method [1].

TABLE 1. Summary results for the fine-tuning setting. After the supervised contrastive learning, optional fine-tuning follows. One-shot and five-shot five-way classification accuracy on Mini-ImageNet is reported. In our experiment, fine-tuning improves accuracy. Our results are averaged over 10,000 episodes.

fine-tune	1-shot	5-shot
✓	78.83	87.76
✗	75.94	86.16

SimCLR [2] self-supervised contrastive loss to support multiple positives in the multiviewed batch.

Notably, performing supervised contrastive learning in the first stage of few-shot learning is proposed instead of performing the conventional training with base classes and cross entropy. The second step of the training procedure is to discard the projection network and fine-tune the encoder network with a new classifier. As representation learning implies, the encoder network becomes discriminative during the first step of the training procedure; therefore, the fine-tuning process is relatively short and is guided by a lower learning rate. Note that we fine-tuned the feature extractor with the base class and cross-entropy, which was pretrained in the first stage of supervised contrastive learning. The fine-tuning process in supervised contrastive learning is optional; we can skip the process that does not touch the feature extractor because we only use the feature extractor at the end. When we follow the linear evaluation protocol, we keep the feature extractor intact, which implies that we skip the fine-tuning process. We chose to use the fine-tuning approach because it produces better results than no fine-tuning as shown in Table 1.

In our experiment, we added a supervised contrastive learning approach as an additional pretraining step in the first few-shot training stage. Furthermore, we fine-tuned the feature extractor with cross-entropy loss using the base class dataset.

IV. EXPERIMENTS

An implementation of our SPTA is publicly available.[1]

A. DATASETS

We examined three few-shot learning datasets, namely Mini-ImageNet, Tiered-ImageNet, and CUB. The **Mini-ImageNet** dataset [44] is composed of 100 classes from the ImageNet [34] dataset. It has 64/16/20 base/validation/novel classes, respectively, with 600 84×84 sized images per class following the split proposed by [32]. The **Tiered-ImageNet** is composed of 608 classes from the ImageNet dataset. It has 351/97/160 base/validation/novel classes, respectively, with 779,165 84×84 sized images in total following the split proposed by [33]. Finally, the **Caltech-UCSD Birds 200-2011** [45] (CUB) dataset is composed of 200 classes and 11,788 images in total. It has 100/50/50 base/validation/novel classes, respectively, with 84×84 sized images following the split proposed by [3].

B. EVALUATIONS

We evaluate the algorithm's score by comparing the final predicted label at the second stage with the ground truth label.

[1] https://github.com/taemin-lee/SPTA

TABLE 2. Accuracy comparison to the state-of-the-art methods for five-way classification on Mini-ImageNet and CUB. The results are categorized according to the backbone network the algorithms use. The bold values are the best results within the algorithms that use the same backbone network. Our results score higher than existing methods by a large margin on Mini-ImageNet and CUB datasets. Our results are averaged over 10,000 episodes.

		Mini-ImageNet		CUB	
Method	Backbone	1-shot	5-shot	1-shot	5-shot
SimpleShot [46]	MobileNet	61.30	78.37	-	-
LaplacianShot [59]	MobileNet	70.27	80.10	-	-
Ours-SPTA	MobileNet	**76.57**	**85.82**	**83.76**	**89.01**
TEAM [31]	ResNet-18	60.07	75.90	80.16	87.17
MTL [40]	ResNet-12	61.2	75.5	-	-
vFSL [57]	ResNet-12	61.23	77.69	-	-
Neg-cosine [25]	ResNet-18	62.33	80.94	72.66	89.40
AFHN [20]	ResNet-18	62.38	78.16	70.53	83.95
MetaOpt [18]	ResNet-12	62.64	78.63	-	-
SimpleShot [46]	ResNet-18	62.85	80.02	-	-
Distill [42]	ResNet-12	64.82	82.14	-	-
ConstellationNet [49]	ResNet-12	64.89	79.95	-	-
DeepEMD [55]	ResNet-12	65.91	82.41	75.65	88.69
FEAT [52]	ResNet-12	66.78	82.05	-	-
IEPT [58]	ResNet-12	67.05	82.90	-	-
TRAML [19]	ResNet-12	67.10	79.54	-	-
CAN+T [11]	ResNet-12	67.19	80.64	-	-
MELR [6]	ResNet-12	67.40	83.40	-	-
DPGN [50]	ResNet-12	67.77	84.60	75.71	91.48
SIB+IFSL [53]	ResNet-10	68.85	80.32	-	-
LaplacianShot [59]	ResNet-18	72.11	82.31	80.96	88.68
TIM-GD [1]	ResNet-18	73.9	85.0	82.2	90.8
Ours-SPTA	ResNet-10	70.49	82.15	70.24	83.95
Ours-SPTA	ResNet-12	72.49	83.47	72.37	85.25
Ours-SPTA	ResNet-18	**78.83**	**87.76**	**88.81**	**93.11**
LEO [35]	WRN28-10	61.76	77.59	-	-
CC+rot [8]	WRN28-10	62.93	79.87	-	-
AWGIM [24]	WRN28-10	63.12	78.40	-	-
SimpleShot [46]	WRN28-10	63.50	80.33	-	-
FEAT [52]	WRN28-10	65.10	81.11	-	-
Transductive tuning [5]	WRN28-10	65.73	78.40	-	-
Logistic Regression with DC [51]	WRN28-10	68.57	82.88	79.56	90.67
SIB [12]	WRN28-10	70.0	79.2	-	-
BD-CSPN [26]	WRN28-10	70.31	81.89	-	-
SIB+IFSL [53]	WRN28-10	73.51	83.21	-	-
LaplacianShot [59]	WRN28-10	74.86	84.13	-	-
TIM-GD [1]	WRN28-10	77.8	87.4	-	-
Ours-SPTA	WRN28-10	**80.32**	**88.76**	-	-

For each few-shot learning episode, N-way K-shot tasks with T queries per class were randomly selected from the dataset of novel classes. We chose $N = 5$, $T = 15$, and $K = 1$ for 1-shot or $K = 5$ for 5-shot classification. We followed the evaluation protocol in [1].

C. IMPLEMENTATION DETAILS

We examined mainly three different backbone network models, namely ResNet-18, MobileNet, and WRN28-10, following the implementation of [1], [46]. We further examined two more ResNet variants, namely ResNet-10 and ResNet-12, in Table 2 for a fair comparison. Note that the number after ResNet indicates the depth of the network. Nevertheless, we report ResNet variants in one group following the convention of [1], [59]. We mainly investigated the alternating direction method (ADM) version of the TIM algorithm, which is faster than the gradient descent (GD) version.[2] We have added a prototype estimation technique [26], [59] to TIM. This further improved the 1-shot classification accuracy. We used a PyTorch [30] re-implementation of RandAugment[3] on the preprocessing stage of supervised

TABLE 3. Accuracy comparison to the state-of-the-art methods for five-way classification on Tiered-ImageNet. The results are categorized according to the backbone network the algorithms use. The bold values are the best results within the algorithms that use the same backbone network. Our results are averaged over 10,000 episodes.

		Tiered-ImageNet	
Method	Backbone	1-shot	5-shot
SimpleShot [46]	MobileNet	69.47	85.17
LaplacianShot [59]	MobileNet	79.13	86.75
Ours-SPTA	MobileNet	**79.17**	**87.16**
MetaOpt [18]	ResNet-12	65.99	81.56
SimpleShot [46]	ResNet-18	69.09	84.58
FEAT [52]	ResNet-12	70.80	84.79
DeepEMD [55]	ResNet-12	71.16	86.03
Distill [42]	ResNet-12	71.52	86.03
MELR [6]	ResNet-12	72.14	87.01
IEPT [58]	ResNet-12	72.24	86.73
DPGN [50]	ResNet-12	72.45	87.24
CAN+T [11]	ResNet-12	73.21	84.93
SIB+IFSL [53]	ResNet-10	78.03	85.43
LaplacianShot [59]	ResNet-18	78.98	86.39
TIM-GD [1]	ResNet-18	79.9	**88.5**
Ours-SPTA	ResNet-18	**81.16**	88.43

[2]https://github.com/mboudiaf/TIM

[3]https://github.com/ildoonet/pytorch-randaugment

TABLE 4. Ablation study on the influence of prototype estimation and supervised contrastive learning. Note that proto refers to prototype estimation and supcon refers to supervised contrastive learning. The bold values are the best results among the methods. Our results show that supervised contrastive learning improves the accuracy on Mini-ImageNet and CUB datasets, whereas prototype estimation improves 1-shot accuracy further. Our results are averaged over 10,000 episodes.

				Mini-ImageNet		Tiered-ImageNet		CUB	
Method	Backbone	proto	supcon	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
TIM-ADM [1]	ResNet-18			73.6	85.0	80.0	**88.5**	81.9	90.7
	ResNet-18	✓		74.86	84.95	**81.34**	88.41	83.66	90.72
	ResNet-18		✓	77.38	**87.82**	80.22	88.49	87.63	93.08
Ours-SPTA	ResNet-18	✓	✓	**78.83**	87.76	81.16	88.43	**88.81**	**93.11**

contrastive learning[4] with $N = 3$ and $M = 20$ to implement modified stacked RandAugment. When pretraining, we used the training epochs of 1,000 for the supervised contrastive learning, and this was followed by five epochs of fine-tuning. Our method is referred as SPTA following the name of combined methods (i.e., supervised contrastive learning, prototype estimation, and TIM-ADM).

D. COMPARISON TO THE STATE-OF-THE-ART

We evaluated the 5-way 1-shot and 5-shot classification accuracy of our method on the Mini-ImageNet and CUB datasets. Our results were averaged over 10,000 episodes following [1], [46] and are summarized in Table 2. In the table, we present methods with 1-shot accuracy over 60% on Mini-ImageNet, and the methods are arranged in ascending order. We excluded results from a semi-supervised setting because these methods require additional data at test time. We observed that consistent accuracy gains over the existing methods, regardless of the backbone network models. For example, 1-shot accuracy improved by more than 6% whereas 5-shot accuracy improved by more than 5% with the MobileNet network backbone on Mini-ImageNet surpassing all the existing methods with the ResNet network backbone model on Mini-ImageNet. With the ResNet-18 network backbone model on Mini-ImageNet, the 1-shot accuracy improved by almost 5% whereas the 5-shot accuracy improved by more than 2% surpassing all the existing methods with the WRN28-10 network backbone model on Mini-ImageNet. Therefore, the gain of our method is comparable to the selection of a better network architecture on Mini-ImageNet in improving performance.

The results for the Tiered-ImageNet are presented in Table 3. Again, our results are averaged over 10,000 episodes, and our method scores competitive accuracy results compared to the existing methods. By comparison, the score gain on the Tiered-ImageNet is not as high as that of Mini-ImageNet and CUBs (i.e., less than or approximately 1%). We assume that the Tiered-ImageNet is a very large dataset compared to Mini-ImageNet and CUB, and thus the visual representation learned from the Tiered-ImageNet is sufficiently discriminative with conventional cross-entropy loss. This implies that our method is data-efficient in terms of the dataset size. Thus, it works particularly well with small datasets, reducing the cost of data preparation.

E. ABLATION STUDY

We evaluated the influence of prototype estimation and supervised contrastive learning on the final accuracy of the method. Instead of the simple mean of support set examples, the prototype estimation technique calculates better initialization points by combining support set examples and query set examples. The results are reported in Table 4, and all of them used ResNet-18 as a backbone network model. From the TIM-ADM baseline method, prototype estimation and supervised contrastive learning were added one by one. We observe that most of the accuracy gain on the Mini-ImageNet and CUB datasets is from the supervised contrastive learning, and the prototype estimation improves 1-shot accuracy further, while it has a marginal impact on 5-shot accuracy. For example, 1-shot accuracy improved by almost 7%, whereas 5-shot accuracy improved by more than 2% on the CUB dataset. Most of the gain in 1-shot accuracy on the CUB dataset is from supervised contrastive learning (i.e., more than 5%), whereas the gain of the prototype estimation is less than 2%. Similarly, most of the gain in 5-shot accuracy on the CUB dataset is from supervised contrastive learning, whereas the gain of the prototype estimation is negligible. We assume that a 5-shot setting provides sufficient information to build a proper prototype for each class, even without the prototype estimation method.

F. DOMAIN-SHIFT

We measure the impact of the domain-shift and report the results in Table 5. All results used ResNet-18 as a backbone network model, and our results were averaged over 10,000 episodes. Domain A $\rightarrow$ B implies that the feature extractor is trained on dataset A, whereas the few-shot learning method is evaluated on dataset B, similar to the setting from [3]. Domain CUB $\rightarrow$ CUB is the baseline result without a domain-shift. Note that the domain-shift from a slightly large-sized dataset to a smaller one (i.e., Mini-ImageNet $\rightarrow$ CUB) drastically degrades the accuracy of the few-shot learning method. The results show a drop in 1-shot accuracy of approximately 29% and 19% in 5-shot accuracy. By comparison, the domain-shift from a much larger dataset (i.e., Tiered-ImageNet $\rightarrow$ CUB) is slightly better than the no domain-shift (i.e., CUB $\rightarrow$ CUB) baseline

[4]https://github.com/HobbitLong/SupContrast

TABLE 5. Summary of domain-shift setting results. Note that no domain-shift and domain-shift from the larger dataset are presented. The bold values are the best results among the domain settings. Note that domain-shift from the larger dataset (i.e., Tiered-ImageNet) improves accuracy on the CUB dataset. Our results score higher than any setting by a large margin on the CUB dataset without resorting to the larger dataset. Our results are averaged over 10,000 episodes.

Method	domain	Backbone	1-shot	5-shot
TIM-GD [1]	CUB →CUB	ResNet-18	82.2	90.8
	Mini-ImageNet →CUB	ResNet-18	53.04	71.04
	Tiered-ImageNet →CUB	ResNet-18	82.58	91.39
Ours-SPTA	CUB →CUB	ResNet-18	**88.81**	**93.11**
	Mini-ImageNet →CUB	ResNet-18	51.50	68.69
	Tiered-ImageNet →CUB	ResNet-18	82.80	90.70

TABLE 6. Results on increasing the number of ways on Mini-ImageNet. We report a more challenging setting of 10-way and 20-way few-shot classification accuracy. The bold values represent the best results among the algorithms. Our results score higher than existing methods by a large margin on 10-way and 20-way Mini-ImageNet datasets. Our results are averaged over 10,000 episodes.

		10-way		20-way	
Method	Backbone	1-shot	5-shot	1-shot	5-shot
baseline [3]	ResNet-18	-	55.00	-	42.03
baseline++ [3]	ResNet-18	-	63.40	-	50.85
TIM-ADM [1]	ResNet-18	56.0	72.9	39.5	58.8
TIM-GD [1]	ResNet-18	56.1	72.8	39.3	59.5
Ours-SPTA	ResNet-18	**61.15**	**77.12**	**43.29**	**64.23**

setting. It improves 1-shot accuracy by approximately 1%. The results show that the existing method requires a much larger dataset in the source domain to build an effective feature extractor under a domain-shift. By contrast, the proposed method provides better feature extraction when using a smaller dataset. Indeed, with our data-efficient augmentation method, CUB → CUB accuracy increases by a large margin surpassing that of Tiered-ImageNet → CUB setting. Our method improves 1-shot accuracy by approximately 6%, and 5-shot accuracy by more than 2%. Therefore, if our method is applied, it is possible to prepare a small base class dataset, and it can still achieve superior accuracy without resorting to the very large base class dataset. Note that our method suffers from more degradation with domain-shift. We conjecture that our method is highly dependent on the base dataset as discussed in Section IV.I.

G. INCREASING THE NUMBER OF WAYS

We investigated the effect of increasing the number of ways on Mini-ImageNet and report the results in Table 6. All results used ResNet-18 as a backbone network model, and our results were averaged over 10,000 episodes. These settings are more challenging than 5-way few-shot classification because there is a greater chance of misclassifying the input image. Our method's 10-way and 20-way few-shot classification accuracy scores are higher than those of existing methods by a large margin. For example, it improves the 10-way 1-shot accuracy by approximately 5%, 10-way 5-shot accuracy by more than 4%, 20-way 1-shot accuracy by approximately 4%, and 20-way 5-shot accuracy by more than 4% compared to the existing best method. This implies that our method improves the overall generalization performance of the few-shot learning method.

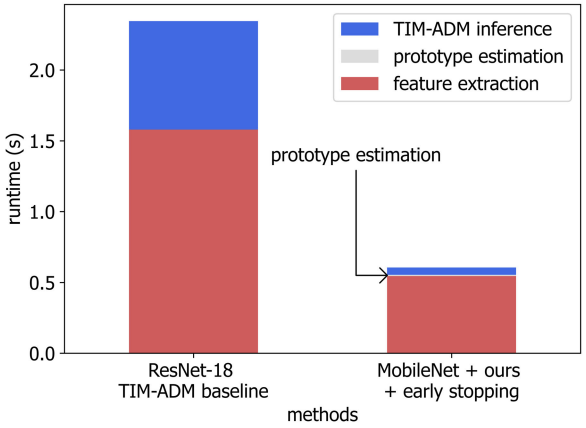

FIGURE 2. Runtime breakdown on NVIDIA Jetson TX2. Runtime measurement is based on a 5-shot standard where the feature extraction batch size is 100, and TIM-ADM algorithm runs a 5-shot classification. The baseline method uses the ResNet-18 backbone network model, no prototype estimation, and 150-iterations of TIM-ADM algorithm. Our method with the MobileNet backbone network model uses 10-iterations of TIM-ADM algorithm (i.e., early stopping). Note that the runtime of prototype estimation is negligible. Best viewed in color.

H. RUNTIME ANALYSIS

The accuracy gain of our method can be utilized for runtime reduction in few-shot learning, which could be especially useful in resource-constrained contexts such as mobile settings. We measured the latency of the methods on an NVIDIA Jetson TX2 to quantify the runtime impacts. The evaluation protocol included 100 warm-up runs, followed by 100 execution runs, and we reported the average over the execution runs. Figure 2 shows a breakdown of the algorithm runtime. The runtime is measured in 5-shot classification (i.e., the feature extraction batch size is 100, and the algorithm assumes 5-shot classification). The baseline method is the TIM-ADM algorithm with the ResNet-18 backbone, which scores a 1-shot accuracy of 73.6 and a 5-shot accuracy of 85.0 as reported in Table 7. Note that the feature extraction latency is larger than the TIM-ADM inference runtime for target task training, which confirms the importance of backbone network selection. We chose to use the MobileNet backbone network with our method under early stopping (i.e., 10 TIM-ADM iterations instead of 150 iterations) and obtained a 1-shot accuracy of 75.13 and a 5-shot accuracy of 85.01, which is still higher than the baseline. Thus, the accuracy gain enabled by our method could be translated to a runtime reduction of 3.87× without loss of accuracy.

TABLE 7. Summary results for the runtime analysis. MobileNet and early stopping are used to reduce the runtime of the algorithms. Supervised contrastive learning and prototype estimation are used to compensate for accuracy loss induced by the runtime reduction methods. Our results are averaged over 10,000 episodes.

Methods	1-shot	5-shot
ResNet-18 TIM-ADM baseline	73.6	85.0
MobileNet + ours + early stopping	75.13	85.01

I. LIMITATIONS

In domain-shift experiment, we observed that the feature extractor trained by our method did not improve the accuracy of the few-shot learning under the domain-shift setting (i.e., the last two rows in Table 5). This implies that our method is highly dependent on the base dataset as it consumes a high number of epochs (i.e., 1,000 epochs for supervised contrastive learning) with the base dataset. Therefore, we suggest that our method's application is limited to scenarios only when a domain-shift is not present. No domain-shift setting encourages a smaller base dataset in real-world implementations.

The cost of supervised contrastive learning is another limitation. A batch size larger than the number of classes in the base dataset is recommended to provide a sufficient number of positives in a single multiviewed batch. This implies many graphic processing units (GPUs) are required to implement and hinder extensive experiments. Specifically, we used two GTX 2080 Ti GPUs to six P100 GPUs to support a single run of the appropriate batch size for supervised contrastive learning. Therefore, we emphasize that a server with sufficient computing power is necessary to implement the pretraining stage. Note that once the pretraining stage and fine-tuning are completed, the remaining algorithm can be implemented in a resource-constrained environment.

In summary, both limitations indicate that our method has insufficient scalability in terms of dataset size, and hence, it is effective in small-scale applications (e.g., few-shot learning).

V. CONCLUSION

We proposed applying supervised contrastive learning for pretraining in the first stage of few-shot learning. The feature extractor was trained using supervised contrastive loss followed by fine-tuning, whereas the classifier performed adaptation using TIM loss. We report that our method is data-efficient (i.e., works well with a small dataset) while retaining competitive accuracy performance with a large dataset. Our experiment shows that we achieved new state-of-the-art results on Mini-ImageNet and CUB datasets.

REFERENCES

[1] M. Boudiaf, Z. I. Masud, J. Rony, J. Dolz, P. Piantanida, and I. B. Ayed, "Transductive information maximization for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–13.

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[3] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.

[4] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 3008–3017.

[5] G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto, "A baseline for few-shot image classification," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–20.

[6] N. Fei, Z. Lu, T. Xiang, and S. Huang, "MELR: Meta-learning via modeling episode-level relationships for few-shot learning," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–20.

[7] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[8] S. Gidaris, A. Bursuc, N. Komodakis, P. P. Perez, and M. Cord, "Boosting few-shot visual learning with self-supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8059–8068.

[9] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. Conf. Artif. Intell. Statist.*, 2010, pp. 297–304.

[10] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–24.

[11] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, "Cross attention network for few-shot classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4003–4014.

[12] S. X. Hu, G. P. Moreno, Y. Xiao, X. Shen, G. Obozinski, D. N. Lawrence, and A. Damianou, "Empirical Bayes transductive meta-learning with synthetic gradients," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–16.

[13] O. J. Hénaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. van den Oord, "Data-efficient image recognition with contrastive predictive coding," 2019, *arXiv:1905.09272*. [Online]. Available: http://arxiv.org/abs/1905.09272

[14] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–23.

[15] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. Int. Conf. Mach. Learn. Deep Learn. Workshop*, 2015, pp. 1–8.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[17] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proc. 33rd Annu. Conf. Cognit. Sci. Soc.*, 2011, pp. 1–7.

[18] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10657–10665.

[19] A. Li, W. Huang, X. Lan, J. Feng, Z. Li, and L. Wang, "Boosting few-shot learning with adaptive margin loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12576–12584.

[20] K. Li, Y. Zhang, K. Li, and Y. Fu, "Adversarial feature hallucination networks for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13470–13479.

[21] X. Li, Q. Sun, Y. Liu, Q. Zhou, S. Zheng, T.-S. Chua, and B. Schiele, "Learning to self-train for semi-supervised few-shot classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10276–10286.

[22] L. Fe-Fei, Fergus, and Perona, "A Bayesian approach to unsupervised one-shot learning of object categories," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 2, Oct. 2003, pp. 1134–1141.

[23] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[24] Y. Guo and N.-M. Cheung, "Attentive weights generation for few shot learning via information maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13499–13508.

[25] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, and H. Hu, "Negative margin matters: Understanding margin in few-shot classification," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 438–455.

[26] J. Liu, L. Song, and Y. Qin, "Prototype rectification for few-shot learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 741–756.

[27] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–14.

[28] A. Mnih and K. Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2265–2273.
[29] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*. [Online]. Available: http://arxiv.org/abs/1803.02999
[30] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
[31] L. Qiao, Y. Shi, J. Li, Y. Tian, T. Huang, and Y. Wang, "Transductive episodic-wise adaptive metric for few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3602–3611.
[32] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–11.
[33] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, B. J. Tenenbaum, H. Larochelle, and S. R. Zemel, "Meta-learning for semi-supervised few-shot classification," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.
[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[35] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.
[36] P. Sermanet, C. Lynch, J. Hsu, and S. Levine, "Time-contrastive networks: Self-supervised learning from multi-view observation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 486–487.
[37] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," 2017, *arXiv:1703.05175*. [Online]. Available: http://arxiv.org/abs/1703.05175
[38] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1857–1865.
[39] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 843–852.
[40] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 403–412.
[41] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," 2019, *arXiv:1906.05849*. [Online]. Available: http://arxiv.org/abs/1906.05849
[42] Y. Tian, Y. Wang, D. Krishnan, B. J. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: A good embedding is all you need?" in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 266–282.
[43] H. Touvron, A. Vedaldi, M. Douze, and H. Jegou, "Fixing the train-test resolution discrepancy," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8252–8262.
[44] O. Vinyals, C. Blundell, T. Lillicrap, K. kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3630–3638.
[45] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-UCSD birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.
[46] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten, "SimpleShot: Revisiting nearest-neighbor classification for few-shot learning," 2019, *arXiv:1911.04623*. [Online]. Available: http://arxiv.org/abs/1911.04623
[47] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7278–7286.
[48] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3733–3742.
[49] W. Xu, Y. xu, H. Wang, and Z. Tu, "Constellation nets for few-shot learning," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–16.
[50] L. Yang, L. Li, Z. Zhang, X. Zhou, E. Zhou, and Y. Liu, "DPGN: Distribution propagation graph network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13390–13399.
[51] S. Yang, L. Liu, and M. Xu, "Free lunch for few-shot learning: Distribution calibration," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–13.
[52] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8808–8817.
[53] Z. Yue, H. Zhang, Q. Sun, and X.-S. Hua, "Interventional few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–23.
[54] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031.
[55] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12203–12213.
[56] H. Zhang, M. Cisse, N. Y. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.
[57] J. Zhang, C. Zhao, B. Ni, M. Xu, and X. Yang, "Variational few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1685–1694.
[58] M. Zhang, J. Zhang, Z. Lu, T. Xiang, M. Ding, and S. Huang, "IEPT: Instance-level and episode-level pretext tasks for few-shot learning," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–16.
[59] M. M. Ziko, J. Dolz, E. Granger, and I. B. Ayed, "Laplacian regularized few-shot learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 11660–11670.

TAEMIN LEE (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from POSTECH, Pohang, South Korea, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree in computer science and engineering with Seoul National University, Seoul, South Korea. His research interest includes data-efficient implementation of deep learning algorithm.

SUNGJOO YOO (Senior Member, IEEE) received the Ph.D. degree from Seoul National University, in 2000. From 2000 to 2004, he was a Researcher with TIMA Laboratory. From 2004 to 2008, he was a Principal Engineer with System LSI, Samsung Electronics. From 2008 to 2015, he was an Associate Professor with POSTECH. In 2015, he joined the Seoul National University, where he is currently a Full Professor. He published more than 40 SCI (E) journals and 100 conference/workshop papers. He contributed to several conferences, such as DAC, DATE, and ESWEEK, and workshops, such as HENP, MPSOC, and RSP, as a reviewers and organizers. His research interest includes software/hardware co-design of deep neural networks.