Pushing the Limit of Fine-Tuning for Few-Shot Learning: Where Feature Reusing Meets Cross-Scale Attention

Ying-Yu Chen[1], **Jun-Wei Hsieh**[1]*, **Xin Li**[2], **Ming-Ching Chang**[2]

[1]College of Artificial Intelligence and Green Energy, National Yang Ming Chiao Tung University, Taiwan.
[2]Department of Computer Science, University at Albany - SUNY.
a0919903171@gmail.com, jwhsieh@nycu.edu.tw, Xin.Li@mail.wvu.edu, mchang2@albany.edu.

Abstract

Due to the scarcity of training samples, Few-Shot Learning (FSL) poses a significant challenge to capture discriminative object features effectively. The combination of transfer learning and meta-learning has recently been explored by pre-training the backbone features using labeled base data and subsequently fine-tuning the model with target data. However, existing meta-learning methods, which use embedding networks, suffer from scaling limitations when dealing with a few labeled samples, resulting in suboptimal results. Inspired by the latest advances in FSL, we further advance the approach of fine-tuning a pre-trained architecture by a strengthened hierarchical feature representation. The technical contributions of this work include: 1) a hybrid design named Intra-Block Fusion (IBF) to strengthen the extracted features within each convolution block; and 2) a novel Cross-Scale Attention (CSA) module to mitigate the scaling inconsistencies arising from the limited training samples, especially for cross-domain tasks. We conducted comprehensive evaluations on standard benchmarks, including three in-domain tasks (miniImageNet, CIFAR-FS, and FC100), as well as two cross-domain tasks (CDFSL and Meta-Dataset). The results have improved significantly over existing state-of-the-art approaches on all benchmark datasets. In particular, the FSL performance on the in-domain FC100 dataset is more than three points better than the latest PMF of Hu et al. 2022.

Introduction

When large and well-annotated datasets are available, deep learning including Convolutional Neural Networks (CNNs) and Vision Transformers has made substantial progress in many areas, including classification, detection, and segmentation. However, deep learning encounters challenges in real-world scenarios due to the cost of collecting sufficient data or even the impracticality to do so. Optimizing deep CNNs or Transformers with significantly larger parameters than the training dataset can result in severe overfitting. Few-Shot Learning (FSL) aims to solve this problem by allowing CNN or Transformer models to learn from very few annotated data and transfer the knowledge across different domains (Wang et al. 2020). FSL methods can be divided into two types: (1) *inductive few-shot*, where each prediction is made independently, and (2) *transductive few-shot*, where predictions are made taking into account the relationships among all labeled and unlabeled samples in a batch.

Many methods have been developed to address FSL in recent years, including transfer-based methods that achieve state-of-the-art (SOTA) performance. With the constraints of few samples, *transfer learning* aims to transfer the knowledge learned from a large labeled dataset (base case) to the unlabeled target dataset (novel case). Recent studies (Shalam and Korman 2022; Hu, Pateux, and Gripon 2022; Hu, Gripon, and Pateux 2021) claim that the main problem of transfer learning is the skewed distribution of feature maps extracted using the feature backbone pre-trained with the base case, which might deviate from that in the target case. To strengthen the quality of the extracted features, they preprocess the feature maps with PCA-like transforms and power transforms to fit a particular distribution ($i.e.$ Gaussian-like), then apply optimal transport to solve the classification problem. The current SOTA method of P>M>F (PMF) (Hu et al. 2022) demonstrates that large-scale models can also be feasible for FSL after pre-training with external data. Furthermore, fine-tuning the model on the target dataset can further boost the FSL performance. However, little is known about how much room is left for further optimization of FSL performance, partially due to the simplicity of the pipeline considered in PMF.

The motivation behind this work is two-fold. On the one hand, conventional wisdom in supervised learning requires abundant labeled training data, because each convolution block in the CNN model is finely trained to provide feature maps for the following convolution blocks. However, when it comes to few-shot classification tasks, the practicality of fine-tuning each convolution block using the novel target data becomes doubtful. Imprecise information generated by previous blocks may lead to drift in the subsequent blocks, and therefore degrades the FSL performance. On the other hand, FSL tasks demand the model to learn from a few labeled samples, which may not fully represent the entire distribution of the target classes. It is desirable to let the model combine local and global information for making predictions. How to handle few-shot examples on a local scale while also capturing higher-level patterns from the entire dataset remains an open question in the FSL literature (Song et al. 2023). Such scaling-related inconsistencies

*Corresponding author.

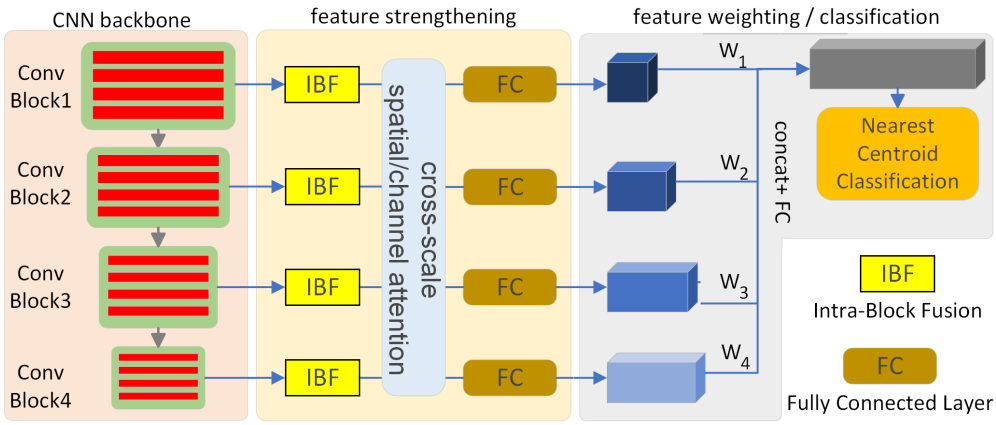

Figure 1: The proposed model architecture. Following CNN backbone, our model consists of two stages: feature strengthening and feature weighting. Feature strengthening includes IBF and CSA modules whose outputs are passed to feature weighting for nearest centroid classification.

become even more pronounced for cross-domain tasks (Fu et al. 2023) where generalization to novel classes by fine-tuning has not been optimized by a simple pipeline such as PMF (Hu et al. 2022). In summary, we identify and address two problems in conventional deep learning models that stem from the few training data in fine-tuning: (1) the inability of conventional models to sufficiently extract informative features for downstream classification tasks, and (2) the suboptimal fine-tuning due to the insufficient training samples as well as the lack of local-global consistency between extracted features.

This paper tackles the above two problems by pushing the limit of fine-tuning for FSL. For the former, we propose a feature-strengthening module, which consists of an **Intra-block Fusion** (IBF) to extract more informative features from the backbone and repair the imprecise features on each scale. We consider ResNet50 (He et al. 2016) and Swin Transformer (Liu et al. 2021) as our feature backbones in this paper to demonstrate the usefulness of this approach in both conventional CNN models and trendy transformer architectures. For the latter, we present how to reconcile scale inconsistency with a novel **cross-Scale Attention** (CSA) module. Fig. 1 shows an overview of our proposed architecture. As shown in Fig. 3(c), our model better concentrated the attention on the important areas. Our experimental results show that our approach has outperformed other competing approaches, including the latest PMF (Hu et al. 2022). In summary, the contributions of this paper are fourfold:

- We present an Intra-block Fusion module to fine-tune features within each convolution block in a pre-trained backbone to strengthen features extracted for FSL.
- We introduce a Cross-Scale Attention module to remedy the insufficiency of local-scale features by exploiting global-local consistencies for better fine-tuning.
- We propose a meta-testing stage for cross-domain FSL tasks, in contrast to the PMF of (Hu et al. 2022). Model fine-tuning with the aid of data augmentation from novel targets leads to improved generalization performance after several gradient updates.
- We achieve new SoTA results on three in-domain few-shot classification tasks with two settings and two cross-domain few-shot classification tasks with several settings. We achieve noticeable FSL performance on the in-domain FC100 dataset that is more than three points better than the latest PMF method.

Related Works

Few-Shot Learning is now a widely studied and active topic (Wang et al. 2020). A popular solution for FSL is meta-learning also known as *learning-to-learn* (Hospedales et al. 2021), which aims to find meta-weights that can quickly converge to the target task with a limited number of target data. This method consists of two phases: *meta-training* and *meta-testing*. In the meta-training phase, the training data are split into a series of episodes/tasks that simulate the target few-shot task, usually in a 5-way 5-shot or 5-way 1-shot form. The few-shot learner learns meta-weights with these tasks arriving in an episodic fashion. After meta-training, the learner undergoes meta-test evaluation on target few-shot tasks. The Model-Agnostic Meta-Learning (MAML) (Finn, Abbeel, and Levine 2017) selects optimal initial weights through gradient descent for the few-shot learner, making fine-tuning fast and simple. REPTILE (Nichol and Schulman 2018) uses the L2 loss to simplify complex computations in MAML. The Latent Embedding Optimization (LEO) network (Rusu et al. 2018) further reduces the complexity by employing a low-dimensional latent embedding optimization process.

Metric Learning (Kaya and Bilge 2019) aims at learning a feature backbone that maps input data into a high-dimensional feature space that preserves similarity among samples in each class. The similarity between the two feature maps is obtained via Cosine or Euclidean distance metrics. Metric learning is widely used in FSL (Vinyals et al. 2016; Sung et al. 2018; Snell, Swersky, and Zemel 2017). The Prototypical Network (ProtoNet) (Snell, Swersky, and

Zemel 2017) is a well-known metric-based FSL method, where prototypes (centroids) are generated by computing the average of every channel on all high-dimensional feature maps corresponding to each class. The feature backbone is trained to center the feature maps in the prototype of each class. For evaluation, the backbone maps the query set to the feature space and performs nearest-centroid classification.

Cross-attention allows the network to selectively attend to relevant parts of each input sequence while computing the representation for a specific element (Bahdanau, Cho, and Bengio 2014; Luong, Pham, and Manning 2015). Cross-attention has been applied in various natural language processing tasks, such as machine translation (Bahdanau, Cho, and Bengio 2014; Vaswani et al. 2017), image captioning (Anderson et al. 2018), and visual question answering (Antol et al. 2015). The Transformer is a popular architecture for implementing cross-attention, which consists of a stack of self-attention and cross-attention layers (Vaswani et al. 2017). Cross-attention is a powerful mechanism to model the relationships between multiple data modalities with SoTA performance in various tasks (Anderson et al. 2018; Tan and Bansal 2019). Most recently, *cross-scale attention* has been applied to the transformers *e.g.*, CrossViT (Chen, Fan, and Panda 2021) and Crossformer (Wang et al. 2023).

Cross-domain few-shot learning handles the challenging but practically relevant cases, where objects and scenes are significantly different between the source and target datasets. In contrast, the within-domain few-shot setting is limited by the assumption that all source data sets are within the same domain. Cross-Domain FSL (CDFSL) (Guo et al. 2020) and Meta-Dataset (Triantafillou et al. 2019) are two major benchmarks of cross-domain few-shot learning that reflect real-world FSL scenarios.

The Proposed Method

This paper aims to address two generic issues for model fine-tuning in the FSL setting: (1) the inability of pre-trained models to accurately extract informative features from target data; and (2) suboptimal fine-tuning of the convolution blocks due to the scarce training samples, leading to a strong bias in the extracted features. We derive two approaches, namely the Intra-Block Fusion (IBF) and Cross-Scale Attention (CSA) modules to address these issues.

Problem Setup

We focus on the few-shot classification tasks, where the classification model must learn from only a few annotated target samples. In a transfer-based few-shot classification task, a well-labeled base dataset $\mathbf{D}_{base}$ is first used to train a backbone F for feature extraction parameterized by θ. The pre-trained model F_θ will be evaluated later with a series of few-shot classification tasks constructed with a novel dataset $\mathbf{D}_{novel}$. Note that $\mathbf{D}_{base}$ and $\mathbf{D}_{novel}$ are completely disjoint.

Each few-shot classification task is built up with a support set $\mathbf{S}_\tau$ and a query set $\mathbf{Q}_\tau$ to form an N-way K-shot classification task τ, where K denotes the number of samples of each class in $\mathbf{S}_\tau$. Both $\mathbf{S}_\tau$ and $\mathbf{Q}_\tau$ consist of N different classes from $\mathbf{D}_{novel}$. Let Q be the number of samples of each class in $\mathbf{Q}_\tau$. After pre-training with $\mathbf{D}_{base}$, we further fine-tune the pre-trained feature backbone F_θ with $\mathbf{S}_\tau$ to better fit the evaluation task and perform inference on $\mathbf{Q}_\tau$.

For the case of *transductive* few-shot learning, prediction is performed by considering all $N \times (K + Q)$ samples together. In contrast, for the case of *inductive* few-shot learning, the prediction is performed independently on each of the $N \times Q$ samples. Here, the prediction does not rely on other query samples, which align better with real-world usage scenarios. Therefore, we aim to solve 5-way 1/5/20/50-shot within-domain and cross-domain few-shot tasks with an inductive setting. The key motivation behind our approach is to improve the fine-tuning by heuristics related to feature engineering *i.e.*, *how best to optimize the feature representation during fine-tuning with the FSL constraint?*

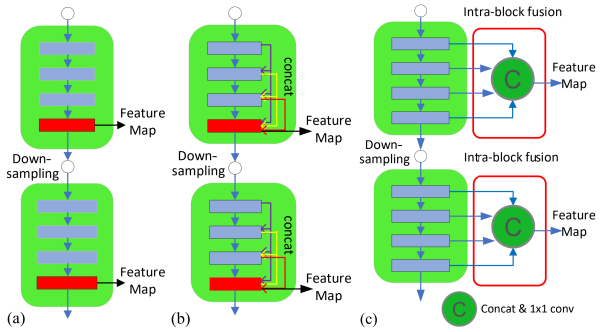

Figure 2: (a) Traditional convolution block only leverages the feature of the last layer from the previous convolution block, which is not informative. (b) Dense block connects each layer to every other layer, which suffers from high computational cost. (c) IBF leverages the features from all layers of a block in a hybrid manner.

Architecture Overview

In a general CNN model, the classification task is conducted mainly with a fully connected layer based on the feature map produced by the convolution blocks followed by a Global-Average Pooling (GAP) layer. For example, ResNet (He et al. 2016) is mainly built up with four convolution blocks, followed by a GAP layer and a fully connected layer. After being passed through each convolution block, the size of the feature map becomes smaller, but with a larger channel size so that deeper features are embedded with a larger receptive field. To leverage multiscale features, our overall architecture takes advantage of the feature maps of each convolution block. Assume that there are B branches of convolution blocks in a backbone F. Four convolution blocks are adopted for feature extraction. Let $\mathbf{B}_b$ denote the b-th convolution block in F; and we use f_b^l to denote the l-th feature layer in $\mathbf{B}_b$, where f_b^L represents the last feature layer in $\mathbf{B}_b$. Most SoTA methods (Wang et al. 2020; Hospedales et al. 2021) adopt only the last feature layer f_b^L in each $\mathbf{B}_b$ to perform FSL tasks.

We propose a two-stage approach (feature strengthening and feature weighting) to better leverage the cross-scale fea-

tures for FSL. Fig. 1 shows the proposed model architecture, which consists of two stages: *feature strengthening* via the IBF module and *feature weighting* via the CSA module. One contribution of this paper is to introduce the idea of IBF to fine-tune features within a convolution block from a pre-trained model to form better informative feature maps for few-shot learning. Another contribution is the CSA module, which remedies the insufficiency of local-scale features from other scales. These fine-tuned feature maps are then fed to the FC layers to integrate the features and are weighted by a trainable parameter W_b, which represents the importance of the features on each scale. These weighted features are then concatenated together and further weighted by a one-by-one convolutional layer to assign importance to each channel. Next, we will elaborate on the design of the IBF and CSA modules.

Intra-Block Fusion (IBF)

The IBF module is designed with the objective of strengthening features while maintaining low cost. To address the problem of few-shot classification, an intuitive strategy is to increase the amount of information that the model can extract from the limited number of samples. Although various convolutions followed one by one are applied to extract various features, only the feature map at the last layer of each convolution block is fed to the next layer (denoted by the red stripe in Fig. 2(a)). However, not only the last layer but also previous layers within the convolution block can provide fine-grained features to generate a more accurate feature map for few-shot learning. One remedy is through the dense block in DenseNet that connects each layer to every other layer in a feed-forward fashion; see Fig. 2(b). Although dense connections can help mitigate the vanishing gradient problem and make the network deep, it is not necessary to build an extremely deep network for few-shot learning. In addition, dense block suffers from a high computational cost. The above observations inspire us to take a hybrid approach and design an IBF module to better leverage the features from all layers within the same block; see Fig. 2(c). Before sending the feature map at the last layer out, a 1×1 convolution is first adopted to fuse all feature layers within the same convolution block (denoted by the yellow box) and then the fused result within more fine-grained information is sent to the decoder for better few-shot learning. We add this technique to the fine-tuning steps to extract more informative features. Using feature-strengthening techniques, our IBF module can generate fine-grained features in a more computationally efficient manner than DenseNet (Huang et al. 2017).

Cross-Scale Attention (CSA)

The CSA module is designed to further strengthen the feature by multiscale attention-guided fusion. In deep learning models, feature maps are propagated from shallow to deep layers during feature extraction. In supervised learning, deep learning models are trained with a large amount of data, allowing each layer to contain rich and informative features. However, the limited amount of training data in the FSL setting prevents the model from being finely trained, resulting in suboptimal feature extraction in each layer. Inspired by recent work of CrossViT (Chen, Fan, and Panda 2021) and Crossformer++ (Wang et al. 2023), we propose a cross-scale attention (CSA) module to compensate for the lack of use of single-scale features, as shown in Fig. 3(a,b). Unlike CrossViT (Chen, Fan, and Panda 2021) dealing only with two levels, we focus on the fusion of weighted features on multiple scales similar to Crossformer++ (Wang et al. 2023), using both spatial and channel attention. Spatial attention produces attention maps to help highlight informative locations and downplay the opposite ones in the feature maps, and channel attention is renowned for enhancing the performance of the models by selectively weighting feature maps in each channel of each feature map.

Unlike other Transformer architectures (Chen, Fan, and Panda 2021; Liu et al. 2021) that pay attention to only feature maps at the same scale, the CSA module accepts multiscale feature maps as input and selectively highlights informative features within them. When the CSA module operates on a feature map f_b of dimensions $C_b \times H_b \times W_b$ obtained from the b-th convolution block in the backbone network F, CSA takes into account feature maps from all scales ($f_{1\sim B}$) and calculates an attention map for f_b by performing cross-attention to generate a new feature map f'_b. As shown in Fig. 3(b), all operations involved are detailed in the CSA block (CSAB), as shown in Fig. 3(a). Cross-attention involves the query vector(q), key vector(k), and value vector(v) from different-scale sources, as opposed to self-attention which operates q, k, and v generated from sources at the same scale. In this context, the feature maps from other scales($f_{1\sim B}$, except for f_b) are first resized to $H_b \times W_b$ to match the size of f_b using interpolation, then transformed by one-by-one convolutions to form the keys and values. f_b is transformed by one-by-one convolutions to form the queries for each scale, the keys, and the values. The subsequent steps are similar to typical self-attention. These queries, keys, and values are reshaped to $C_b \times N_b$, while $N_b = H_b \times W_b$.

For the cross-scale spatial attention module, q is transposed to shape $N_b \times C_b$, then the matrix is multiplied by k to form a $N_b \times N_b$ spatial attention map. This spatial attention map highlights the informative regions of the input feature map by taking into account feature maps from multiple scales. For the cross-scale channel attention module, k is transposed to shape $N_b \times C_b$, then q is matrix multiplied by k to form a $C_b \times C_b$ channel attention map. Then channel attention map assigns importance to each channel in a feature map by considering feature maps from multiple scales. These attention maps are generated using a cross-scale attention mechanism that computes attention weights based on channel-wise and spatial-wise relationships among the input feature maps from different scales. After calculating attention maps, v value from each scale is matrix-multiplied by the attention map. These values from each scale are then concatenated with the v generated from the scale of f_b followed by a one-by-one convolution to generate the output feature map for the scale of f_b. The output feature maps involve the feature maps from each scale. The CSA module enhances the spatial/channel relations within the input fea-

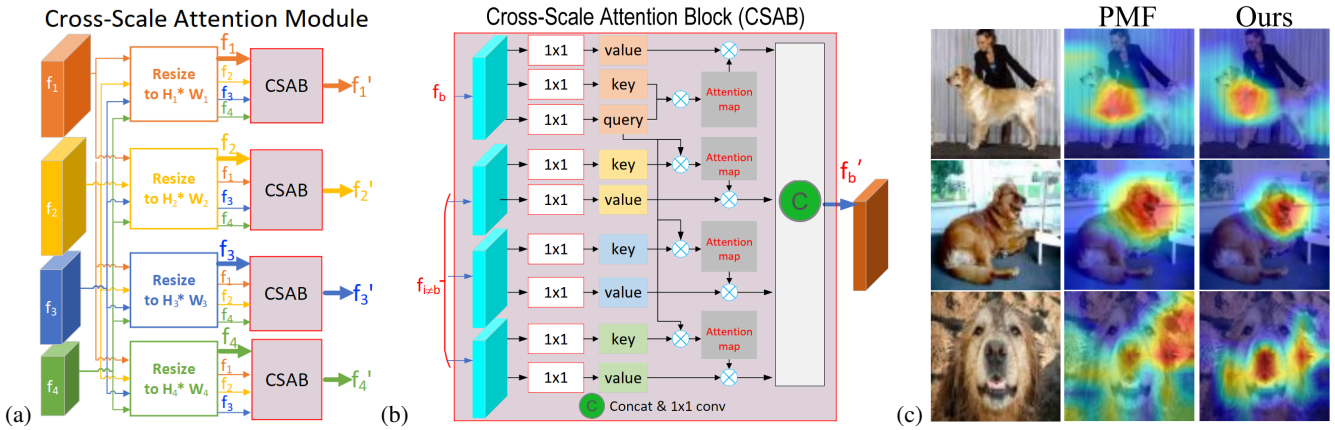

Figure 3: (a) The proposed Cross-Scale Attention (CSA) module consists of several CSA Blocks, each of which corresponds to a resized feature map; (b) The CSA Block concatenates v values from each scale and passes through a one-by-one convolution to generate the output feature map. (c) Class Activation Maps (CAM) of three images from the novel domain: (Left) raw image, (Middle) CAM generated using PMF (Hu et al. 2022), (Right) CAM generated using our CSA module.

ture map with the help of cross-scale features, allowing for more effective feature extraction and classification. In summary, the proposed CSA module can function as both channel and spatial attention with minimal adjustment.

Pre-training, Meta-training, and Meta-testing

Similarly to (Hu et al. 2022), we have pre-training and meta-training (fine-tuning) for FSL. However, for cross-domain FSL tasks, fine-tuning becomes more challenging due to novel classes. Inspired by MAML (Finn, Abbeel, and Levine 2017), we propose a meta-testing to achieve permutation-invariant FSL (Ye and Chao 2021).

Pre-training. We opt for traditional supervised learning and DINO (Caron et al. 2021) as our pre-training algorithms, using ImageNet as the pre-training foundation, following the approach in (Hu et al. 2022). DINO is a well-known self-supervised learning algorithm utilizing the consistency in feature embedding capability. It achieves this by predicting relationships between a large cropped region and several smaller cropped regions within an image. The large crop captures the entire foreground object, while the smaller ones only capture portions of it. Pre-training on a large external dataset is crucial for enhancing model flexibility, particularly for FSL tasks.

Meta-training. We adopt an effective meta-training process using episodic tasks. Each episodic task follows the N-way K-shot setup identical to the target task. Through episodic training, the model learns a meta-weight capable of rapid adaptation to the classes presented in the target task.

Meta-testing. For standard within-domain few-shot tasks, we directly apply the meta-trained model to all new target instances. However, in cross-domain few-shot tasks where target instances belong to previously unseen domains, disrupting the learned feature representation, we fine-tune the model. This involves data augmentation through several gradient steps, akin to the procedure outlined in (Hu et al. 2022). Due to limited labeled data, we utilize the support set for model fine-tuning. Initially, prototypes for each class are computed using the original support set. The model is then updated using a loss derived from a pseudo-query set, generated by augmenting the support set in various ways.

Experimental Results

Experimental Setup

We tested our approach using the standard inductive setting: a 5-way, $1/5$-shot, 15-query classification scenario, common for within-domain few-shot classification tasks. In addition, we use 5-way, $5/20/50$-shot, and 15-query classification settings for the cross-domain few-shot classification task; that is, $N = 5, K = 1/5, Q = 15$, and $N = 5, K = 5/20/50, Q = 15$. The average prediction accuracy among each randomly sampled $\mathbf{Q}_\tau$ is used as the evaluation metric.

In-domain few-shot classification Evaluation is performed on three standard *in-domain* benchmarks: miniImageNet (Vinyals et al. 2016), CIFAR-100 Few-Shots (CIFAR-FS) (Bertinetto et al. 2018), and FC100 (Oreshkin, Rodríguez López, and Lacoste 2018). The miniImageNet contains 100 randomly chosen classes from ILSVRC-12, which are then divided into 64 training, 16 validation, and 20 testing classes. Each class contains 600 images of 84×84. CIFAR-FS and FC100 are two different split datasets based on CIFAR-100, each containing 60 base classes for training, 20 classes for validation, and 20 novel classes for evaluation. Each class consists of 600 images of 32×32.

Cross-domain few-shot classification Evaluation is performed on two *cross-domain* benchmarks: CDFSL (Guo et al. 2020) and Meta-Dataset (Triantafillou et al. 2019).

CDFSL comprises four distinct datasets, each exclusive to a particular domain: ChestX (X-ray images), ISIC2018 (dermoscopic images of skin lesions), EuroSAT (satellite images), and CropDisease (plant disease images). These datasets reflect real-world FSL use cases, capturing the challenges and cost associated with data collection. The model undergoes evaluation simulating real-world usage scenarios.

Method (backbone)	Ext. data	CIFAR-FS		miniImageNet		FC100	
		5w1s	5w5s	5w1s	5w5s	5w1s	5w5s
Matching Networks (CNN-4-64) (Vinyals et al. 2016)		-	-	43.5	55.3	-	-
MAML (CNN-4-64) (Finn, Abbeel, and Levine 2017)		-	-	48.7	63.1	-	-
ProtoNet (CNN-4-64) (Snell, Swersky, and Zemel 2017)		55.5	72.0	49.4	68.2	37.5	51.4
MetaOpt-SVM (RN12) (Lee et al. 2019)		72.0	84.2	62.6	78.6	49.8	67.2
Meta-Baseline (RN12) (Chen et al. 2021b)		-	-	68.6	83.7	-	-
EASY 3xResNet12 (RN12) (Bendou et al. 2022)		75.24	89.0	71.75	87.15	48.0	64.7
Baseline++ (WRN-28-10) (Chen et al. 2019)		67.5	80.1	57.5	73.0	-	-
S2M2R (WRN-28-10) (Mangla et al. 2020)		74.8	87.5	64.9	83.2	-	-
HCTransformers (ViT-S) (He et al. 2022)		79.9	90.5	74.7	89.2	48.3	66.4
AMDIM (AmdimNet) (Chen et al. 2021a)	✓	-	-	76.8	91.0	-	-
P>M>F (IN1K, Sup., RN50)† (Hu et al. 2022)	✓	76.73	87.60	83.74	94.33	58.91	74.01
P>M>F (IN1K, Sup., Swin-S)† (Hu et al. 2022)	✓	84.41	92.16	95.26	98.20	69.92	82.98
Ours (1N1K, Sup., RN50)	✓	78.62	89.15	88.78	96.76	64.03	79.64
Ours (1N1K, Sup., Swin-S)	✓	**87.06**	**93.62**	**97.03**	**99.04**	**73.07**	**86.02**

Table 1: Accuracy comparisons with other methods, including the SoTA with *in-domain* few-shot settings. Methods using external data are marked with ✓. † indicates results reproduced by us using the publicly released codes from the original work.

	ChestX			ISIC			EuroSAT			CropDisease		
	5w5s	5w20s	5w50s	5w5s	5w20s	5w50s	5w5s	5w20s	5w50s	5w5s	5w20s	5w50s
ProtroNet (RN10)	24.05	28.21	29.32	39.57	49.50	51.99	73.29	82.27	80.48	79.72	88.15	90.81
RelationNet (RN10)	22.92	26.63	28.45	39.41	41.77	49.32	61.31	74.43	74.91	68.99	80.45	85.08
MetaOptNet (RN10)	22.53	25.53	29.35	36.28	49.42	54.80	64.44	79.19	83.62	68.41	82.89	91.76
Finetune (RN10)	26.97	31.32	35.49	48.11	59.31	66.48	79.08	87.64	90.89	89.25	95.51	97.68
CHEF (RN10)	24.72	29.71	31.25	41.26	54.30	60.86	74.15	83.31	86.55	86.87	94.78	96.77
DeepCluster2 (1N1K, RN50)	26.51	31.51	34.17	40.73	49.91	53.65	**88.39**	92.02	93.07	93.63	96.63	97.04
P>M>F (IN1K, Sup., RN50)	27.04	35.33	41.32	46.97	61.51	69.72	85.57	92.30	95.55	93.14	96.75	98.04
P>M>F (IN1K, DINO, RN50)	26.56	32.98	36.07	45.03	60.34	67.94	86.10	93.90	95.79	93.98	97.74	98.10
Ours (IN1K, Sup., RN50)	**27.83**	**34.20**	**43.18**	**48.04**	**62.36**	**70.51**	86.98	93.58	95.72	94.39	97.81	**98.86**
Ours (IN1K, DINO, RN50)	27.43	33.62	36.64	47.10	61.90	69.62	88.23	**94.26**	**96.08**	**95.21**	**97.90**	98.54

Table 2: Accuracy comparison with others including the SoTA on the CDFSL dataset with *cross-domain* few-shot settings.

The Meta-Dataset comprises 10 diverse datasets spanning a broad range of domains: ImageNet-1k (INet), Omniglot (Omglot), FGVC-Aircraft (Acraft), CUB-200-2011 (CUB), Describable Textures (DTD), QuickDraw (QDraw), FGVCx Fungi (Fungi), VGG Flower (Flower), Traffic Signs (Sign), and MSCOCO (COCO), where abbreviations are shown in parentheses. Each domain includes a train/val/test set. Two training protocols are conducted: (1) Throughout the meta-training and validation procedures, the train/val splits of the first eight in-domain datasets are utilized, while the test splits of all ten datasets are leveraged for meta-testing. (2) Only ImageNet-1k is used for meta-training and validation procedures, while all other configurations remain the same.

Implementation Details

Training details. We start by pre-training our model with ImageNet-1k through supervised learning and DINO (Caron et al. 2021). Subsequently, we apply a standard meta-training algorithm on the pre-trained model. The meta-train process spans 100 epochs, with each epoch comprising 2,000 training episodes/tasks. At the end of each epoch, we conduct validation with 1,000 validation episodes/tasks. The training episodes are randomly selected from the training classes in $\mathbf{D}_{base}$ for each epoch, while validation episodes are randomly selected from the validation classes in $\mathbf{D}_{base}$ and remain constant across all epochs. The complete training duration is 100 epochs, with the learning rate starting at 10^{-6} and increasing to 10^{-5} over 5 epochs as warm-up, followed by a decrease to 10^{-6} using the learning rate of cosine annealing (Loshchilov and Hutter 2017).

Evaluation details. For the standard within-domain few-shot tasks, we verify our meta-trained model with 2000 testing episodes/tasks randomly selected from $\mathbf{D}_{novel}$. For CDFSL, we transfer the model trained with miniImageNet to the new target. The fine-tuning step heavily relies on the learning rate. For each domain, we carefully select the learning rate by comparing model results within a reasonable range (e.g., 0.001, 0.005, 0.0001, 0) across 600 randomly chosen episodes. Subsequently, we fine-tune the model and assess its performance on another set of 600 randomly chosen episodes, using the chosen learning rate. For Meta-Dataset, we uniformly and randomly select ways, shots, and query images based on dataset specifications, except for ImageNet-1k and Omniglot.

Benchmark Results

We report the results of the evaluation against existing methods, including the SoTA PMF method (Hu et al.

8 in-domain datasets	In-domain								Out-of-domain		
	INet	Omglot	Acraft	CUB	DTD	QDraw	Fungi	Flower	Sign	COCO	Avg.
SUR (RN18)	57.20	93.20	**90.10**	82.30	73.50	81.90	67.90	88.40	67.40	51.30	75.32
URL (RN18)	57.51	94.51	88.59	80.54	76.17	81.94	68.75	92.11	63.34	54.03	75.75
ITA (RN18)	57.35	**94.96**	89.33	81.42	76.74	**82.01**	67.40	92.18	83.55	55.75	78.07
P>M>F (RN50)	67.51	85.91	80.30	81.67	**87.08**	72.84	60.03	94.69	87.17	**58.92**	77.61
Ours (RN50)	**75.68**	91.67	87.93	**90.72**	85.02	73.43	**69.22**	**96.10**	**90.31**	56.07	**81.61**
	In-domain	Out-of-domain									
In-domain=INet	INet	Omglot	Acraft	CUB	DTD	QDraw	Fungi	Flower	Sign	COCO	Avg.
ALFA+FP-MAML (RN12)	52.80	61.87	63.43	69.75	70.78	59.17	41.49	85.96	60.78	48.11	61.41
BOHB (RN18)	51.92	67.57	54.12	70.69	68.34	50.33	41.38	87.34	51.80	48.03	59.15
CTX (RN34)	62.76	**82.21**	79.49	80.63	75.57	**72.68**	51.58	95.34	82.65	**59.90**	74.28
P>M>F (RN50)	67.08	75.33	75.39	72.08	**86.42**	66.79	50.53	94.14	86.54	58.20	73.25
Ours (RN50)	**75.96**	80.58	**81.75**	**86.05**	83.24	67.27	**58.37**	**95.75**	**89.08**	52.15	**77.02**

Table 3: Accuracy comparison with others including the SoTA on the Meta-Dataset with **cross-domain** few-shot settings.

			CIFAR-FS		miniImageNet		FC100	
	IBF	CSA	5w1s	5w5s	5w1s	5w5s	5w1s	5w5s
RN50	✓		77.98	87.94	86.36	94.91	61.01	77.09
		✓	78.04	88.35	87.08	96.31	63.24	78.95
	✓	✓	**78.62**	**89.15**	**88.78**	**96.76**	**64.03**	**79.64**
Swin-S	✓		86.38	92.23	95.92	97.61	71.65	84.87
		✓	86.70	92.41	96.58	98.50	72.57	85.49
	✓	✓	**87.06**	**93.62**	**97.03**	**99.04**	**73.07**	**86.02**

Table 4: Comparison of effectiveness of the Intra-block Fusion (IBF) and the Cross-Scale Attention (CSA) modules.

	CIFAR-FS		miniImageNet		FC100	
RN50	5w1s	5w5s	5w1s	5w5s	5w1s	5w5s
local	77.73	88.75	85.93	95.24	59.71	77.48
cross	**78.62**	**89.15**	**88.78**	**96.76**	**64.03**	**79.64**
Swin-S	5w1s	5w5s	5w1s	5w5s	5w1s	5w5s
local	85.97	92.58	95.51	97.81	70.28	84.19
cross	**87.06**	**93.62**	**97.03**	**99.04**	**73.07**	**86.02**

Table 5: Ablation study comparing the use of local-scale *vs.* cross-scale features.

2022) on common benchmarks for both *within-domain* and *cross-domain* few-shot settings. Best-performing scores are marked in bold. RN50 and Swin-S are the abbreviations of ResNet50 and Swin Transformer small, respectively.

Table 1 shows *within-domain* performance comparison. Our method outperforms all comparison methods including SoTA PMF on CIFAR-FS, miniImageNet, and FC100 by about 2.0% to 2.5% in accuracy, when evaluated with 5-way 5-shot and 5-way 1-shot settings.

Tables 2 and 3 show *cross-domain* performance comparisons on the CDFSL dataset and Meta-Dataset, respectively. Table 2 shows that our method outperforms others with all settings except for EuroSAT with 5-way 5-shot setting, where our accuracy is only 0.16% lower than DeepCluster2 (Ericsson, Gouk, and Hospedales 2021). Table 3 shows that our method achieves the highest average accuracy with a significant improvement of 3.77%. These results provide compelling evidence of the significant adaptability of our method in divergent scenarios.

Ablation Study

We conducted ablation studies on two aspects, namely (1) the effectiveness of the IBF and CSA modules and (2) the distinction between the use of local-scale features *vs.* cross-scale features. Evaluations are performed on CIFAR-FS, miniImageNet, and FC100 with 5-way 5-shot and 5-way 1-shot settings, with the same setting as our main method.

Table 4 shows that both the IBF and CSA modules demonstrated performance enhancements. When both modules were removed, the performance decreased compared to the baseline models. This makes sense because the shallower layers are trained in the baseline models to provide features for the deeper layers. Incorporating these features directly into the final output could introduce inaccuracies. Therefore, our introduced modules enhance cross-scale features, making them more valuable for downstream tasks and thus effectively capitalizing on the hierarchically learned representations to boost overall performance.

We swap the CSA module with self-attention layers for each scale. Table 5 shows that harnessing cross-scale features to amplify features from each scale unanimously resulted in improved performance for two different backbones and three benchmark datasets. The largest performance improvement is up to 4.32 points (RN50 on FC100).

Conclusion

We identify and tackle the limitations associated with conventional CNN models when applied to few-shot classification tasks. To address these limitations, we propose two feature strengthening methods: Intra-Block Fusion (IBF) to preserve cross-layer features within each convolution block, and the Cross-Scale Attention (CSA) module to alleviate the constraints imposed by using a single scale. Our architecture achieves new state-of-the-art results on three within-domain benchmarks and two cross-domain benchmarks, substantiating its effectiveness in addressing the challenges of few-shot classification. Future works will involve further evaluation of the IBF and CSA modules on additional computer vision tasks, such as semantic segmentation and crowd counting.

References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 6077–6086.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *ICCV*, 2425–2433.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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

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

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *ICCV*, 9650–9660.

Chen, C.-F. R.; Fan, Q.; and Panda, R. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *ICCV*, 357–366.

Chen, D.; Chen, Y.; Li, Y.; Mao, F.; He, Y.; and Xue, H. 2021a. Self-supervised learning for few-shot image classification. In *ICASSP*, 1745–1749. IEEE.

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C. F.; and Huang, J.-B. 2019. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*.

Chen, Y.; Liu, Z.; Xu, H.; Darrell, T.; and Wang, X. 2021b. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *ICCV*, 9062–9071.

Ericsson, L.; Gouk, H.; and Hospedales, T. M. 2021. How well do self-supervised models transfer? In *CVPR*, 5414–5423.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 1126–1135.

Fu, Y.; Xie, Y.; Fu, Y.; and Jiang, Y.-G. 2023. StyleAdv: Meta Style Adversarial Training for Cross-Domain Few-Shot Learning. In *CVPR*, 24575–24584.

Guo, Y.; Codella, N. C.; Karlinsky, L.; Codella, J. V.; Smith, J. R.; Saenko, K.; Rosing, T.; and Feris, R. 2020. A broader study of cross-domain few-shot learning. In *ECCV*, 124–141. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, Y.; Liang, W.; Zhao, D.; Zhou, H.-Y.; Ge, W.; Yu, Y.; and Zhang, W. 2022. Attribute Surrogates Learning and Spectral Tokens Pooling in Transformers for Few-shot Learning. In *CVPR*, 9119–9129.

Hospedales, T.; Antoniou, A.; Micaelli, P.; and Storkey, A. 2021. Meta-learning in neural networks: A survey. *IEEE PAMI*, 44(9): 5149–5169.

Hu, S. X.; Li, D.; Stühmer, J.; Kim, M.; and Hospedales, T. M. 2022. Pushing the Limits of Simple Pipelines for Few-Shot Learning: External Data and Fine-Tuning Make a Difference. In *CVPR*, 9068–9077.

Hu, Y.; Gripon, V.; and Pateux, S. 2021. Leveraging the feature distribution in transfer-based few-shot learning. In *ICANN*, 487–499. Springer.

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

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *CVPR*, 4700–4708.

Kaya, M.; and Bilge, H. Ş. 2019. Deep metric learning: A survey. *Symmetry*, 11(9): 1066.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-learning with differentiable convex optimization. In *CVPR*, 10657–10665.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Mangla, P.; Kumari, N.; Sinha, A.; Singh, M.; Krishnamurthy, B.; and Balasubramanian, V. N. 2020. Charting the right manifold: Manifold mixup for few-shot learning. In *WACV*, 2218–2227.

Nichol, A.; and Schulman, J. 2018. Reptile: a scalable metalearning algorithm. *arXiv:1803.02999*, 2(3): 4.

Oreshkin, B.; Rodríguez López, P.; and Lacoste, A. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. *NeurIPS*, 31.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2018. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*.

Shalam, D.; and Korman, S. 2022. The Self-Optimal-Transport Feature Transform. *arXiv:2204.03065*.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *NeurIPS*, 30.

Song, Y.; Wang, T.; Cai, P.; Mondal, S. K.; and Sahoo, J. P. 2023. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*, 1199–1208.

Tan, H.; and Bansal, M. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Triantafillou, E.; Zhu, T.; Dumoulin, V.; Lamblin, P.; Evci, U.; Xu, K.; Goroshin, R.; Gelada, C.; Swersky, K.; Manzagol, P.-A.; et al. 2019. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*, 30.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *NeurIPS*, 29.

Wang, W.; Chen, W.; Qiu, Q.; Chen, L.; Wu, B.; Lin, B.; He, X.; and Liu, W. 2023. Crossformer++: A versatile vision transformer hinging on cross-scale attention. *arXiv preprint arXiv:2303.06908*.

Wang, Y.; Yao, Q.; Kwok, J. T.; and Ni, L. M. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys*, 53(3): 1–34.

Ye, H.-J.; and Chao, W.-L. 2021. How to Train Your MAML to Excel in Few-Shot Classification. In *ICLR*.