Revisiting Learnable Affines for Batch Norm in Few-Shot Transfer Learning

Moslem Yazdanpanah* [1,5] Aamer Abdul Rahman* [2,5] Muawiz Chaudhary [4,5] Christian Desrosiers [2]
Mohammad Havaei [3] Eugene Belilovsky †[4,5] Samira Ebrahimi Kahou †[2,5]

[1]University of Kurdistan; [2]École de technologie supérieure; [3]Imagia; [4]Concordia University, [5]Mila

Abstract

Batch normalization is a staple of computer vision models, including those employed in few-shot learning. Batch normalization layers in convolutional neural networks are composed of a normalization step, followed by a shift and scale of these normalized features applied via the per-channel trainable affine parameters γ and β. These affine parameters were introduced to maintain the expressive powers of the model following normalization. While this hypothesis holds true for classification within the same domain, this work illustrates that these parameters are detrimental to downstream performance on common few-shot transfer tasks. This effect is studied with multiple methods on well-known benchmarks such as few-shot classification on miniImageNet, cross-domain few-shot learning (CD-FSL) and META-DATASET. Experiments reveal consistent performance improvements on CNNs with affine unaccompanied batch normalization layers; particularly in large domain-shift few-shot transfer settings. As opposed to common practices in few-shot transfer learning where the affine parameters are fixed during the adaptation phase, we show fine-tuning them can lead to improved performance.

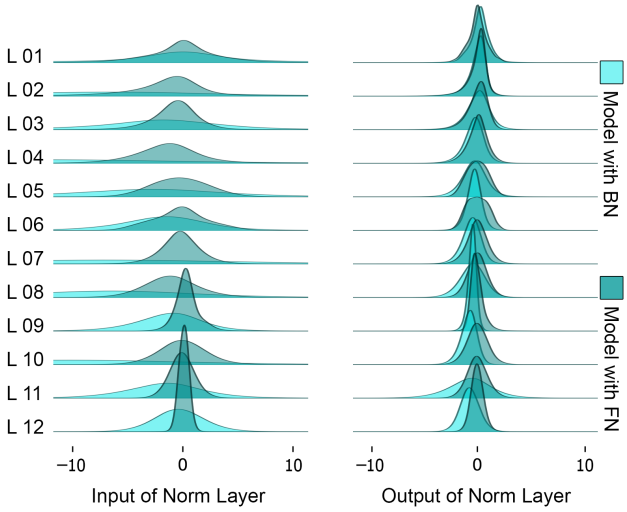

Figure 1. Aggregated distributions of normalization layers from a ResNet10 model, pre-trained on miniImageNet, and fed with samples from EuroSat. Although the input distributions differ, the model with FN appears to accomodate the role of the affine parameters, resulting in a more centered input to the normalization layer (left) with relatively similar output distributions (right).

1. Introduction

Over the last decade, the growing availability of data has allowed deep neural networks to achieve remarkable performance on various visual recognition tasks [10, 12, 13]. However, the size and variability of the dataset can have a huge impact on the effectiveness of these models. Deep neural networks trained on datasets from a specific distribution often fail to generalise their performance to new domains, creating a compelling need for large-scale datasets [33]. Developments in few-shot learning (FSL) have enabled deep neural networks to draw data representations from target classes with just a few labelled samples [7, 29, 35].

Throughout the literature, batch normalization (BN) [14] layers are ubiquitous in FSL techniques. They speed up model convergence and are believed to add regularization [22]. Adding BN layers to deep learning models stabilizes the distribution of layer input features by modulating their mean and variance [14]. This results in a smoother optimization landscape and improved performance across a variety of computer vision problems [28]. Despite these achievements, there persists a poor understanding of the source of effectiveness from BN layers. Moreover, recent work has revealed that these gains may not be the result of alleviating internal covariate shift, as initially believed [28].

BN layers typically consist of two steps. First, the input features are normalized by the mean and standard deviation

*Equal contributions. †Equal senior author contribution. This research was partially funded by NSERC Discovery Grants [*E.B., S.E.K.*]; and CIFAR AI Chair [*S.E.K*]. [*M.C.*] is funded by IVADO PRF Grant. We thank Compute Canada and Calcul Quebéc for computational resources. Correspondence to: Moslem.Yazdanpanah@gmail.com, aamer.abdul-rahman.1@ens.etsmtl.net

over the spatial dimensions of each channel across a mini-batch. These normalized features are then scaled and shifted by the trainable coefficient γ and bias β (the affine parameters). In this paper, we refer to the initial step as "Feature Normalization" (FN). The affine parameters acting in the second step serve to preserve the expressive capabilities of the neural network following the normalization of features.

In order to bridge distributional gaps between the source and target datasets, notable efforts have been directed towards the area of domain adaptation. Li et al. [21] state that the label information is usually stored in the network's weight matrix while the statistics of the BN layer represent domain-related knowledge. This interpretation leads to a reasonable question – upon facing a novel target distribution, are the BN's affine parameters still helpful? Recent work has touched upon the auxiliary benefits of these affine parameters towards weight layers [8]. However, the negative effect of this biased adaptation to training labels when facing novel labels of a distant target domain is yet to be explored.

In this work, we investigate the effect of replacing BN with FN layers towards the generalizability of convolutional neural networks (CNN) in few-shot transfer learning. Our experiments on multiple few-shot transfer benchmarks such as miniImageNet [35], cross-domain few-shot learning (CDFSL) [11] and META-DATASET [34] confirm that using batch normalization when learning on the source domain harms few-shot generalization on the target domain. We show Feature Normalization achieves significantly better results in similar settings. We hypothesize the decrease in performance in models using BN could be related to BN's sparsifying effect in conjunction with the ReLU (See section 4.4 for quantitative evaluation). Ablation studies are conducted to determine the isolated influence of γ and β towards few-shot transfer tasks.

To learn more generalizable features from the source domain and to better adapt to the target domain, we develop a novel methodology for few-shot transfer where we apply Feature Normalization during representation learning on the source domain (we refer to this learning phase as "base training") and batch normalization when adapting to the target domain—we refer to this technique as "Fine-Affine". With this methodology, we gain from the best of both worlds and achieve an overall better result.

The rest of this paper is structured as follows. Few-shot transfer and normalization-based approaches are reviewed in Section 2. Formal definitions for Feature Normalization and Fine-Affine are presented in Section 3, Section 4 describes the benchmarks and experimental setups as well as the evaluation results. Finally, we draw conclusions in Section 5.

2. Related work

2.1. Few-Shot learning

In recent years, significant efforts have been directed towards the development of few-shot learning (FSL) [3, 7, 9, 18, 25, 30, 35]. FSL aims to adapt learners to novel classes using only a limited number of labelled samples. Research in FSL has typically been predicated to settings with limited domain shifts between the source and novel classes. Meta-learning techniques have garnered significant attention in FSL based on their coherent and simplistic qualities. Current meta-learning methods can be broadly classified into metric and optimization-based approaches. Metric-based approaches [15, 26, 29, 31, 35] utilize the distance between embeddings of the support and query samples to classify the novel query images, wherein only the classifier is adapted to the downstream task. Optimization approaches [7, 25] incorporate the entire network within the adaptation phase. Furthermore, several works propose a transfer learning [3, 17, 32, 37] approach following the hypothesis that the base and novel classes share discriminative features. Other methods instead employ model initialization techniques to speed up convergence and improve the classifier, based on the assumption that the initialization which works well on the source domain will be effective on the novel target domain [18, 30].

Recently, research in FSL has focused on settings where there is a significant domain gap between the source and target data [11, 34]. Despite the popularity of meta-learning, Guo et al. [11] demonstrated that the standard transfer learning and fine-tuning approach outperforms current state-of-the-art meta-learning methods when facing a large distribution shift. Furthermore, several methods utilize unlabelled data from the target domain in the evaluation stages in order to reduce the distributional shift [19, 26, 27, 38]. Progress in self-training [39] and self-supervised learning [6] methods have led to promising solutions for CDFSL problems. STARTUP [24] is a notable state-of-the-art approach in distant tasks which employs a combination of self-supervised and self-training components for CDFSL.

2.2. Batch Normalization

The introduction of batch normalization layers [14] have sped up model convergence and enabled the training of deeper networks. The initial hypothesis stated that BN alleviates the issue of internal covariate shift following the notion that the standardization of features reduces dramatic shifts to the inputs of convolutional layers [14]. Since then, this explanation has been cast into doubt in [28], where internal covariate shift was induced in BN layers to find a negligible effect on BN effectiveness. Another study suggests that the optimization of weight magnitude and direction is decoupled by BN [16]. Empirical experiments demonstrate that

BN layers smoothen the optimization landscape [28]; while providing a slight regularization effect [22] and aiding in deterring the exploding activations problem [1].

Our work investigates the role of BN and its affine parameters when facing extreme domain shift, particularly in few-shot settings. Li et al. [20] use BN layers towards domain adaptation in their AdaBN method. This method assumes that data from different domains will be transformed into representations with similar distributions following standardization. The authors of AdaBN present its benefits through empirical experiments carried out on CNNs for image classification tasks. MetaNorm [5] is a BN-based domain adaptation technique that utilizes a meta-learning approach to predict domain-specific BN statistics for domain-independent batch normalization. Frankle et al. [8] highlight the expressive powers of the BN affine parameters. They conduct experiments that show that BN affine parameters play a positive role in improving model performance. However, their work does not take into consideration settings where there is a distributional gap between the training and target data.

In this paper, we explore the role of affine parameters towards the generalizability of few-shot learners in the presence of extreme distribution shift between the source and target data. We perform experiments on state-of-the-art methods such as STARTUP. Furthermore, we adapt AdaBN to an FSL environment to study the effect of the affine parameters on BN-based domain adaptation techniques on cross-domain few-shot transfer.

3. Methods

3.1. Definitions

Notations here are adopted from the survey paper [36]. Domain $\mathcal{D}$ consists of a feature space $\mathcal{X}$ and marginal probability distribution $P(X)$, where $X = \{x_1, ..., x_n\} \in \mathcal{X}$.

3.2. Feature Normalization

Let S be a batch of labelled examples $\{(x_i^s, y_i^s)\}_{i=1}^N$ of size N from a source domain $\mathcal{D}^s$ where $x_i^s \in \mathcal{X}^s$ and $y_i^s \in \mathcal{Y}^s$, and Θ be a deep convolutional neural network consisting of L layers with weight matrices θ^l where l represents the layer index. If h represents the intermediate features of Θ for layer l, the Feature Normalization layer at layer l is computed for each channel and can be defined as[1]:

$$\mathrm{FN}(h_c) = \frac{h_c - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}. \tag{1}$$

Here, subscript c represents the channel index, ϵ is a small number added to prevent divisions by zero, μ_c and σ_c are the first and second moments of h_c respectively defined as:

$$\mu_c = \frac{1}{NHW} \sum\nolimits_{n,h,w} h_{nchw} \tag{2}$$

$$\sigma_c = \sqrt{\frac{1}{NHW} \sum\nolimits_{n,h,w} (h_{nchw} - \mu_c)^2}, \tag{3}$$

where H and W are the spatial dimensions of h_c.

3.3. Fine-tuning affines (Fine-Affine)

In much of the few-shot learning literature [3, 24], only the linear classifier is adapted in the fine-tuning stage, leaving the backbone frozen. Typically this is done to allow for rapid adaptation, but also because fine-tuning the backbone does not improve performance as the model becomes over-parameterized. In another work [23], the affine parameters are utilized to provide task specific conditioning. The affines represent a small number of parameters and may allow the model to adapt without overfitting to the few samples presented in the few-shot fine-tuning stage. It is thus natural to consider adapting both the linear layer and the affine parameters. In this paper, we refer to the joint fine-tuning of the linear classifier and the affine parameters as Fine-Affine.

	BN	FN	BN✗	FN✗
	5-WAY, 1-SHOT			
EuroSAT	65.17±0.46	**67.04±0.44**	66.32±0.46	**68.69±0.45**
CropDisease	72.98±0.47	**76.97±0.44**	74.01±0.46	**77.52±0.43**
ISIC	29.33±0.29	**30.89±0.31**	31.08±0.32	**31.40±0.31**
ChestX	22.37±0.22	**22.67±0.23**	22.28±0.22	**22.71±0.22**
	5-WAY, 5-SHOT			
EuroSAT	84.32±0.31	**86.43±0.28**	84.07±0.34	**86.75±0.29**
CropDisease	91.86±0.25	**93.59±0.23**	91.92±0.25	**94.02±0.22**
ISIC	42.11±0.32	**45.12±0.33**	**47.50±0.36**	46.39±0.33
ChestX	25.38±0.23	**26.22±0.24**	25.21±0.23	**26.39±0.24**
	5-WAY, 20-SHOT			
EuroSAT	91.32±0.20	**92.49±0.19**	92.43±0.19	**93.02±0.19**
CropDisease	96.80±0.15	**97.65±0.13**	97.48±0.15	**98.01±0.12**
ISIC	54.53±0.33	**56.92±0.33**	**62.00±0.35**	60.04±0.33
ChestX	29.55±0.24	**30.73±0.24**	30.20±0.26	**31.77±0.26**
	5-WAY, 50-SHOT			
EuroSAT	93.55±0.17	**94.34±0.15**	**95.18±0.15**	95.15±0.14
CropDisease	98.09±0.10	**98.62±0.09**	98.86±0.07	**98.88±0.07**
ISIC	60.78±0.31	**63.16±0.31**	**69.05±0.32**	68.25±0.32
ChestX	32.33±0.25	**33.64±0.25**	34.36±0.28	**35.85±0.27**

Table 1. Fine-tuning the linear classifier versus affines + linear classifier (methods marked with ✗: stands for Fine-Affine). All methods make use of a ResNet18 pre-trained on ImageNet and evaluated over 2000 episodes. BN: BN configuration, linear classifier finetuned; BN✗: BN configuration, linear classifier + affines fine-tuned; FN: FN configuration, linear classifier fine-tuned; FN✗: FN configuration, linear classifier + affines fine-tuned.

[1]For the sake of simplicity, we implement the Feature Normalize layer using standard Batch Norm modules with disabled affine parameters.

		EuroSAT	CropDisease	ISIC	ChestX	Adapt Time	Base-Train Time
				5-WAY, 1-SHOT			
Baseline	BN	61.54±0.89	68.87±0.84	31.96±0.60	**22.43±0.40**	1.00	1.00
	FN	**62.61±0.87**	**70.91±0.85**	**32.80±0.61**	22.20±0.40	1.00	1.00
Baseline ✗	BN	61.49±0.91	68.94±0.85	31.77±0.58	**22.54±0.40**	1.00	1.00
	FN	**61.81±0.87**	**71.11±0.86**	**32.58±0.60**	22.33±0.40	1.00	1.00
AdaBN	BN	59.44±0.84	68.07±0.85	**33.82±0.62**	**22.41±0.40**	7.25	1.00
	FN	**63.27±0.86**	**71.50±0.85**	33.67±0.63	22.11±0.39	7.25	1.00
AdaBN ✗	BN	60.40±0.87	68.04±0.85	33.31±0.61	**22.32±0.40**	7.25	1.00
	FN	**63.29±0.88**	**71.32±0.86**	**33.43±0.63**	22.14±0.40	7.25	1.00
STARTUP	BN	63.88±0.84	**75.93±0.80**	32.70±0.60	**23.09±0.43**	1251	1.00
	FN	**64.00±0.88**	74.56±0.85	**35.12±0.64**	22.93±0.43	1251	1.00
				5-WAY, 5-SHOT			
MAML*	BN	71.70±0.72	78.05±0.68	40.13±0.58	23.48±0.96	0.70	4.83
ProtoNet*	BN	73.29±0.71	79.72±0.67	39.57±0.57	24.05±1.01	0.35	4.18
Baseline	BN	79.90±0.69	89.93±0.52	43.47±0.60	**26.17±0.43**	1.00	1.00
	FN	**80.51±0.67**	**91.14±0.49**	**45.03±0.62**	25.90±0.43	1.00	1.00
Baseline ✗	BN	79.81±0.71	90.15±0.51	43.11±0.58	**26.39±0.43**	1.00	1.00
	FN	**80.03±0.70**	**91.11±0.49**	**45.34±0.60**	25.78±0.42	1.00	1.00
AdaBN	BN	80.47±0.63	90.11±0.52	**47.97±0.64**	**26.00±0.42**	7.25	1.00
	FN	**82.34±0.62**	**91.29±0.49**	47.92±0.64	25.87±0.43	7.25	1.00
AdaBN ✗	BN	80.39±0.65	89.95±0.51	**46.74±0.61**	**25.93±0.43**	7.25	1.00
	FN	**82.00±0.64**	**90.99±0.50**	47.20±0.62	25.86±0.43	7.25	1.00
STARTUP	BN	82.29±0.60	**93.02±0.45**	47.20±0.61	26.94±0.44	1251	1.00
	FN	**82.51±0.62**	92.86±0.43	**48.54±0.63**	**27.17±0.44**	1251	1.00
				5-WAY, 20-SHOT			
MAML*	BN	81.95±0.55	89.75±0.42	52.36±0.57	27.53±0.43	0.70	4.83
ProtoNet*	BN	82.27±0.57	88.15±0.51	49.50±0.55	28.21±1.15	0.35	4.18
Baseline	BN	87.59±0.45	95.83±0.29	54.67±0.58	**32.24±0.46**	1.00	1.00
	FN	**88.31±0.46**	**96.50±0.27**	**56.71±0.59**	32.11±0.46	1.00	1.00
Baseline ✗	BN	88.31±0.48	96.06±0.28	56.62±0.57	**32.58±0.46**	1.00	1.00
	FN	**88.94±0.46**	**96.62±0.26**	**58.92±0.57**	31.88±0.46	1.00	1.00
AdaBN	BN	88.90±0.45	96.03±0.28	59.04±0.60	31.33±0.46	7.25	1.00
	FN	**89.95±0.42**	**96.68±0.27**	**59.65±0.60**	**31.57±0.45**	7.25	1.00
AdaBN ✗	BN	88.87±0.46	95.99±0.28	58.23±0.58	31.58±0.46	7.25	1.00
	FN	**89.91±0.43**	**96.55±0.27**	**59.24±0.59**	**31.68±0.47**	7.25	1.00
STARTUP	BN	89.26±0.43	**97.51±0.21**	58.60±0.58	33.19±0.46	1251	1.00
	FN	**89.63±0.43**	97.43±0.23	**59.98±0.59**	**33.54±0.46**	1251	1.00
				5-WAY, 50-SHOT			
ProtoNet*	BN	80.48±0.57	90.81±0.43	51.99±0.52	29.32±1.12	0.35	4.18
Baseline	BN	90.43±0.41	97.58±0.21	60.84±0.56	35.71±0.47	1.00	1.00
	FN	**91.10±0.39**	**98.03±0.19**	**63.17±0.56**	**35.80±0.47**	1.00	1.00
Baseline ✗	BN	91.64±0.39	97.85±0.19	64.29±0.57	**36.25±0.48**	1.00	1.00
	FN	**92.34±0.36**	**98.27±0.17**	**65.90±0.58**	34.81±0.49	1.00	1.00
AdaBN	BN	91.75±0.37	97.77±0.20	63.69±0.58	34.36±0.47	7.25	1.00
	FN	**92.73±0.34**	**98.13±0.19**	**64.56±0.58**	**35.09±0.47**	7.25	1.00
AdaBN ✗	BN	92.04±0.37	97.73±0.20	64.15±0.56	35.08±0.47	7.25	1.00
	FN	**92.86±0.34**	**98.11±0.18**	**65.28±0.56**	**35.18±0.48**	7.25	1.00
STARTUP	BN	91.99±0.36	98.45±0.17	64.20±0.58	36.91±0.50	1251	1.00
	FN	**92.59±0.33**	**98.53±0.16**	**65.90±0.56**	**37.67±0.47**	1251	1.00

Table 2. Few-shot transfer results under extreme distribution shift. All methods make use of a ResNet10 backbone evaluated over 600 episodes. (BN): BN configuration, linear classifier fine-tuned; (FN): FN configuration, linear classifier fine-tuned; methods marked with ✗: stands for Fine-Affine, linear classifier + affines fine-tuned.; The affines of (FN Fine-Affine) are restored prior to the fine-tuning stage. * Results from [11].

	BN (γ, β)	BN (γ)	BN (β)	FN (ours)
	5-WAY, 1-SHOT			
EuroSAT	65.17±0.46	66.67±0.80	66.69±0.80	**67.04±0.44**
CropDisease	72.98±0.47	75.32±0.88	75.68±0.84	**76.97±0.44**
ISIC	29.33±0.29	30.11±0.54	29.41±0.55	**30.89±0.31**
ChestX	22.37±0.22	22.62±0.39	22.47±0.41	**22.67±0.23**
	5-WAY, 5-SHOT			
EuroSAT	84.32±0.31	85.56±0.52	86.18±0.52	**86.43±0.28**
CropDisease	91.86±0.25	92.91±0.47	93.09±0.43	**93.59±0.23**
ISIC	42.11±0.32	44.48±0.58	43.26±0.59	**45.12±0.33**
ChestX	25.38±0.23	26.09±0.43	26.01±0.44	**26.22±0.24**
	5-WAY, 20-SHOT			
EuroSAT	91.32±0.20	91.73±0.35	92.11±0.34	**92.49±0.19**
CropDisease	96.80±0.15	97.26±0.26	97.51±0.23	**97.65±0.13**
ISIC	54.53±0.33	56.41±0.59	56.25±0.60	**56.92±0.33**
ChestX	29.55±0.24	30.26±0.43	30.15±0.44	**30.73±0.24**
	5-WAY, 50-SHOT			
EuroSAT	93.55±0.17	93.59±0.29	94.11±0.27	**94.34±0.15**
CropDisease	98.09±0.10	98.31±0.19	98.57±0.16	**98.62±0.09**
ISIC	60.78±0.31	62.46±0.58	**63.25±0.57**	63.16±0.31
ChestX	32.33±0.25	33.03±0.45	32.60±0.46	**33.64±0.25**

Table 3. Ablation studies on the affine parameters of the BN layer. All methods utilize a ResNet18 backbone pre-trained on the ImageNet dataset and evaluated over 2000 episodes. BN (γ, β): Standard BN configuration; FN: FN configuration; BN (γ): BN with disabled β; BN (β): BN with disabled γ.

4. Experiments

We study the effect of Feature Normalization (FN) applied on state-of-the-art few-shot learning frameworks, such as STARTUP [24], and evaluate FN in few-shot transfer settings. We adapt AdaBN [20], a BN-based domain adaptation technique, to FSL setup and investigate the effect of replacing BN with FN. Ablation studies are carried out on the BN affine parameters γ and β to evaluate their isolated influence towards performance in cross-domain few-shot transfer. The adaptation time overhead relative to the baseline is calculated for all methods to emphasize on the computational cost of more complex methods while achieving similar performance gains to FN. We compare the sparsity of feature representations trained with BN and FN across different datasets. Finally, we investigate the effect of Fine-Affine (*i.e.* reactivating the affine parameters while fine-tuning on the target domain).

4.1. Benchmarks

CDFSL The challenging CDFSL benchmark [11] is used as the basis for our experiments. MiniImageNet [35], which consists of images based on object recognition tasks, is utilized as the base training dataset (source). Experiments are conducted on the more extensive ImageNet [4] dataset as well. The benchmark's target data is composed of four datasets, each from very different domains with respect to the source images of miniImageNet and ImageNet. These datasets consist of EuroSAT (satellite imagery to determine land usage), CropDiseases (plant images to identify botanical diseases), ChestX (chest X-rays to detect pathology), and ISIC2018 (images of skin lesions to detect melanoma).

Following [24], for methods with an unsupervised component namely STARTUP and AdaBN, we randomly sample 20% of the unlabelled images from novel classes in the target dataset to be used in the base training. The remaining examples are used for inference. Similar to [11], we perform experiments in FSL classification setting where the support set is composed of 5 classes with k samples per class (5-way k-shot), where k $\in$ {1, 5, 20, 50}. Evaluation of models pre-trained on source miniImageNet are carried out over 600 target episodes, and reported with mean accuracy and 95% confidence intervals. Models that were pre-trained on source ImageNet are evaluated in a similar fashion, except over 2000 target episodes.

META-DATASET Further empirical experiments are carried out on META-DATASET [34]. Here, ImageNet is used as the base representation learning dataset. The target dataset comprises of Omniglot, Aircraft, Birds, VGG Flower, Quickdraw, Fungi, Textures, TrafficSigns and MSCOCO. In addition to the domain shifts between the source and target datasets, there are additional challenges with META-DATASET in that task generation does not follow the standard K-way N-Shot tasks. The tasks are generated with a random number of ways, support and query shots. Additional details on the task generation process for META-DATASET can be found in [34].

4.2. Implementation and Evaluation Details

The few-shot transfer experiments in Table 2 are carried out on the publicly available CDFSL benchmark [11]. The Baseline is standard transfer learning, trained for 400 epochs on miniImageNet with a batch size of 128. STARTUP's teacher model is trained for 400 epochs on miniImageNet and its student model is trained for 1000 epochs on unlabelled samples from 20% of each target dataset, both using a batch size of 256. The remaining 80% of target datasets are utilized for fine-tuning, as described in Sec. 4.1. All methods in Table 2 make use of the ResNet10 architecture [12]. The experiments on META-DATASET in Table 4 were carried on ResNet18 models [12], based on the implementation in [2]. The experiments in this paper were carried out using the Tesla V100 SXM2 16 GB GPU.

For minImageNet source cases with very-low shot cases such as 1-shot, we observe a high variance in results across different seeds. For instance, on 5 different seeds, the fine-tuned baseline trained in [24] produced the following mean accuracies for 5-way 1-shot classification on EuroSAT:

{63.11%; 63.01 %; 61.50%; 62.68 %; 61.91 %}, each with 95% confidence interval of about 0.9 across episodes. We note as well some reported improvements are often in the range of 2-3% in the mean [24], thus we can see the variance due to the training procedure can be higher than typically assumed. In order to take into consideration this high variance that has been unaccounted for in other studies, we average the results obtained from experiments carried out over 5 seeds.

	BN	FN
Omniglot	60.73±1.35	**65.86±1.34**
Aircraft	51.96±1.03	**54.74±1.04**
Birds	**63.51±1.03**	62.93±1.02
Textures	73.86±0.77	**74.52±0.75**
QuickDraw	58.02±1.06	**63.96±0.99**
Fungi	34.77±1.03	**36.67±1.04**
VGG Flower	82.97±0.81	**85.44±0.78**
Traffic signs	54.80±1.13	**58.18±1.09**
MSCOCO	40.66±1.13	**41.88±1.14**

Table 4. Evaluation of FN and BN on META-DATASET. Both methods make use of a ResNet18 backbone pre-trained on source ImageNet and finetuned to the target tasks. We observe substantial gains with the FN based ResNet18.

Overhead calculation It is worth noting that different methods have varying computational demands and complexity. To make a fair comparison of the computational costs relative to performance gains, we calculate the base training and adaptation times as a means of elaborating cost differences among the evaluated methods. The base train time for each method relative to that of the Baseline is presented in Table 2. Having an overhead of 1 shows equivalency to the base training of the Baseline, while 0.75 indicates that the method only requires 75% of the Baseline training time. Using the same approach, the adaptation time ratio is calculated as the time needed to adapt a single sample of the target domain $\mathcal{D}^t$, either supervised or unsupervised, for each episode relative to the amount of time taken by the Baseline.

Evaluation setting Any inference technique that is dependent on a feature representation and is built with BN in its backbone can be used agnostically with FN. For fairness and simplicity, in this work, we follow the same evaluation setting used for experiments on the CDFSL benchmark [11] and STARTUP [24]. For META-DATASET, we follow the evaluation settings used in [2]. Here, the weights of the feature extractor are frozen after base training on the source dataset. A linear classifier is then trained on the support set of the down stream task. Finally, the model is evaluated on the task query set.

4.3. BatchNorm related methods

Our work focuses on batch normalization and thus we consider comparison with AdaBN, an approach that is not commonly used in the few-shot literature, to facilitate more rigid comparisons. AdaBN is based on adjusting BN statistics to the statistics of the target domain. In the following paragraph we describe how AdaBN was adapted to an FSL paradigm.

AdaBN few-shot setup AdaBN, introduced in [20], is a lightweight BN-based domain adaptation technique that has been shown to improve performance on transfer learning methods towards image classification tasks. The method is an unsupervised technique that utilizes unlabelled data from the target domain and adapts the BN statistics to bridge the domain gap between the source and target distributions. Despite the efficacy of this approach in transfer learning, it has been neglected in the few-shot literature. In this study, we evaluate AdaBN in few-shot settings both in near-domain and when facing a significant domain shift, with both BN and FN configurations. AdaBN utilizes the standard Baseline model pre-trained on the source dataset, and adapts for an additional few epochs of forward passes on unlabelled samples $\mathcal{D}^t$. Here, the statistics of the model's normalization layer are updated based on the target feature distribution $p(x)^t$ while the learnable parameters of the model remain frozen.

4.4. Few-Shot Learning Results

In this section we first run a study to give insight into how affine parameters affect the distribution of features under domain shift. We then present our results in multiple few shot transfer tasks.

Post-activation distributions We hypothesize that the issue with BN affine parameters under domain shift is related to the sparsifying properties of ReLU. Due to the thresholding property of ReLU, a potentially small shift in a neuron's pre-activation output distribution, for example the distribution becoming more peaked, can result in substantial shifts in the post-activation distribution. Moreover excessive thresholding can lead to information loss. To obtain further insights, we investigate the average number of non-zero entries (the sparsity) in the feature representations of the penultimate layer of imagenet trained ResNet18 and mini-imagenet ResNet10 models under distribution shift. For each model we compute its sparsity on the source data (ImageNet or miniImageNet) and subsequently compare this to the sparsity of other datasets from the CDFSL benchmark. Furthermore, as seen from Table 6, distribution shift (going from imagenet to CDFSL data) tends to induce substantially sparser representations relative to in-distribution data. We

hypothesize this excessive sparsity leads to degraded performance and less general features. On the other hand, the centered distributions produced by the FN trained models do not have as high a sparsity both for source data and for target datasets, motivating their potential for alleviating this issue with affine parameters and distribution shifts.

	1-shot	5-shot	20-shot
Baseline (BN)	54.56±0.84	76.18±0.69	84.53±0.52
Baseline (FN)	55.16±0.83	76.03±0.67	84.23±0.53
AdaBN (BN)	54.21±0.85	76.10±0.68	84.43±0.53
AdaBN (FN)	55.10±0.84	76.06±0.67	84.16±0.54

Table 5. Near-domain few-shot evaluation on Baseline and AdaBN. Models are pre-trained on miniImageNet and evaluated on novel classes of ImageNet over 600 episodes.

	ImageNet	Eurosat	ISIC	ChestXRay	CropDisease
ResNet18 (BN)	53.5	37.2	47.3	58.4	54.2
ResNet18 (FN)	60.9	53.7	58.9	64.5	62.1
ResNet10 (BN)	30.0	16.9	16.2	20.7	30.4
ResNet10 (FN)	50.7	26.3	27.6	37.3	47.7

Table 6. The percentage of non-zero entries in the feature maps is computed after the final ReLU activation in each pre-trained model. Small changes in the continuous distribution lead to large changes in the discrete distribution. From in-domain to cross-domain transfer, we find that sparsity increases as we move cross-domain. ResNet10 and ResNet18 are pre-trained on miniImageNet and ImageNet respectively.

Feature distribution analysis We evaluate the cross-domain feature distribution before and after the BN (light green) and FN (dark green) layers, as presented in Figure 1. In both cases, models are pre-trained on miniImagenet while the recorded distributions are samples from EuroSat. For the sake of simplicity, per-sample channel wise spatial means are aggregated to one distribution per layer in order to negate the visual biases of the channel distributions. The left and right column represent the normalization layer's input and output distributions respectively.

Cross-domain few-shot transfer Table 2 reports the results of our experiments on the CDFSL benchmark. Across all datasets and 1, 5, 20, and 50 shot settings (consistent with the CDFSL benchmark), the average performance of the models configured with FN exceed that of the BN models. Notably, there is an average improvement of 2.04% for 1 shot classification on the CropDisease dataset when the the Baseline is equipped with FN across 5 seeds. Simply configuring the Baseline model with FN obtains results that rival (within error bars) the more complex and computationally expensive STARTUP, which employs a large amount of unlabelled data to bridge the domain gap. Relative performance gains can be observed across all three methods (Baseline, AdaBN and STARTUP) when equipped with FN. The best overall results were produced by STARTUP with FN. The results of the experiments on the META-DATASET, presented in Table 4, show that FN brings significant improvements on this benchmark was well. The superior results produced by FN models indicate that the BN affine parameters, γ and β, have a generally negative impact on downstream few-shot transfer tasks when facing a significant domain shift.

Near-domain few-shot transfer Further analysis was carried out on few-shot transfer tasks to determine FN's effectiveness on target data that are not as distant from the source training data as datasets from the CDFSL benchmark. In this experiment, we used miniImageNet as the source dataset and novel unseen classes from ImageNet as the target data. Even though these source and target images are essentially from the same dataset, the unseen classes of the target presents a task with some domain shift from the source. Upon inspecting the results presented in Table 5, it can be observed that FN does not improve on the performance of BN. Moreover, BN produced better results than FN on in-domain validation data while training on ImageNet, as seen from Figure 2. These results support the hypothesis that FN is more beneficial for Few-Shot transfer tasks when facing a significant domain shift.

Fine-Affine (Fine-tuning the γ and β) The results of the affine fine-tuning experiment are presented in Table 1. Baseline models equipped with both BN and FN were evaluated with the Fine-Affine configuration. After the affine parameters of FN Fine-Affine were disabled during the base training phase, they are restored and initialized to 1 and 0 for γ and β. The ImageNet dataset was chosen as the source domain on which both the BN and FN models are pre-trained on. It can be observed from the results that there are strong performance gains as a result of the Fine-Affine setup on both BN and FN models, but that FN models still outperform BN models. Improvements are noted for 1, 5, 20 and 50 shot classification across all four datasets, with noteworthy gains of 7.57% on 20-shot classification of ISIC and 2.21% on 50-shot classification of ChestX by the BN Fine-Affine and FN Fine-Affine models respectively. These results suggest that affine parameters are useful towards task specific adaption in few-shot transfer settings, without causing the models to overfit to the small number of samples presented in few-shot environments. The Fine-Affine adaptation was not as effective when using miniImageNet as the source dataset, as observed from Table 2. However, on both ImageNet and miniImageNet base datasets, FN provides a marked improvement over BN on the Fine-Affine method.

Computational overhead The computation overhead for each evaluated method is presented in Table 1. From a practical perspective, even though STARTUP produced the overall best results, its adaptation time ratio is 1251 times than that of the Baseline approach. This is due to an expensive unsupervised learning step. This makes such computationally complex methods inapplicable in tight situations. On the other hand, despite the slow paced base training for MAML and ProtoNet (time ratios to Baseline are 4.83 and 4.18 respectively), they are relatively faster in adaptation time with a lower ratio for MAML (0.70) and a considerably small portion relative to the Baseline time for ProtoNet (0.35). AdaBN is a computationally expensive method with overhead adapt time larger than that of MAML, ProtoNet and the Baseline. In practice, the adaptation time is not of the same scale as that of base training; the adaptation happens on small number of annotated samples from the target domain. It can thus be considered as a negligible overhead compared to base training which benefits from a large supervised sample set. Finally, the proposed FN modification, results in improving all methods, without imposing extra overhead cost. It is noteworthy that FN even slightly decreases the base-train time overhead due to its reduced number of parameters.

AdaBN AdaBN is a domain adaptation technique based on batch noramalization that has been adapted to a few-shot learning in this paper. The evaluation of AdaBN on cross-domain few-shot transfer can be viewed in Table 2. The results indicate that AdaBN, with both BN and FN configurations, produce considerable improvements on ISIC few-shot benchmark, with a notable 4.86% gain over the baseline on 5-shot classification. However, on the rest of the target datasets, AdaBN produced more marginal gains relative to the Baseline. In terms of AdaBN with the BN and FN configuration, FN consistently outperforms BN on most of the experiments. Further analysis was carried out on near-dormain target data with unseen novel classes from ImageNet. The results, presented in Table 5, show that AdaBN does not produce any benefits over the standard Baseline. Furthermore, AdaBN with FN does not improve over the BN version in near-domain experiment. This shows that replacing BN with FN can produce substantial gains for BN-based domain adaptation techniques when facing a large domain shift, but is not effective towards small domain shift tasks.

4.5. Ablation studies

As described in Section 1, the batch normalization layer consists of two learnable affine parameters whereas the Feature Normalization layer performs normalization in the absence of these affines. In this section, we carry out ablation experiments on these parameters to determine their isolated influence on few-shot transfer performance. The results of

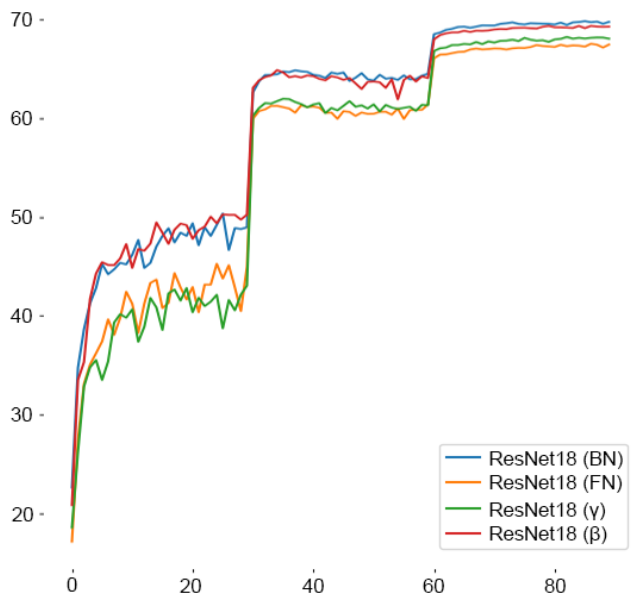

Figure 2. Top 1 validation accuracy while training on the ImageNet dataset over 90 epochs. ResNet18 (BN): BN configuration; ResNet18 (FN): FN configuration; ResNet18 (γ): BN with disabled β; ResNet18 (β): BN with disabled γ. Even though source performance is lower, the few shot transfer performance is higher as seen in Table 3

the ablation experiments on the CDFSL benchmark are presented in Table 3. It can be observed that both BN(γ) and BN(β) produce more accurate classification than BN across 1, 5, 20 and 50 shots on all four datasets. The margin of improvement is higher on BN(β) relative to BN(γ). Feature Normalization, where both γ and β are removed, is the best performing configuration for distant domain few-shot transfer.

5. Conclusion

Feature Normalization layers improve few-shot generalization performance on shifted domains, leveraging a smaller number of model parameters. By stabilizing the output distribution of convolutional layers, Feature Normalization improves robustness against distribution shifts. It captures and normalizes the statistical distribution of data features while preventing the affines from overfitting to the training source labels. Feature Normalization is consistent with widely used batch normalization implementations and can be easily integrated into existing CNN architectures. It is observed that the proposed normalization technique only helps in few-shot transfer and the effect is more pronounced as the data distribution shift increases.

References

[1] Johan Bjorck, Carla Gomes, Bart Selman, and Kilian Q. Weinberger. Understanding batch normalization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 7705–7716, 2018. 3

[2] Malik Boudiaf, Ziko Imtiaz Masud, Jérôme Rony, Jose Dolz, Ismail Ben Ayed, and Pablo Piantanida. Mutual-information based few-shot classification, 2021. 5, 6

[3] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019. 2, 3

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009. 5

[5] Ying-Jun Du, Xiantong Zhen, Ling Shao, and Cees G. M. Snoek. Metanorm: Learning to normalize few-shot batches across domains. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. 3

[6] Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M Hospedales. Self-supervised representation learning: Introduction, advances and challenges. *arXiv preprint arXiv:2110.09327*, 2021. 2

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 1, 2

[8] Jonathan Frankle, David J Schwab, and Ari S Morcos. Training batchnorm and only batchnorm: On the expressive power of random features in cnns. *arXiv preprint arXiv:2003.00152*, 2020. 2, 3

[9] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 2

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. 1

[11] Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *European Conference on Computer Vision*, pages 124–141. Springer, 2020. 2, 4, 5, 6

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 1, 2

[15] Gregory R. Koch. Siamese neural networks for one-shot image recognition. In *ICML Deep learning workshop*, 2015. 2

[16] Jonas Moritz Kohler, Hadi Daneshmand, Aurélien Lucchi, Thomas Hofmann, M. Zhou, and Klaus Neymeyr. Exponential convergence rates for batch normalization: The power of length-direction decoupling in non-convex optimization. In *Proceedings of machine learning research*, 2019. 2

[17] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 491–507. Springer, 2020. 2

[18] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019. 2

[19] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. *Advances in Neural Information Processing Systems*, 32:10276–10286, 2019. 2

[20] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018. 3, 5, 6

[21] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016. 2

[22] Ping Luo, Xinjiang Wang, Wenqi Shao, and Zhanglin Peng. Towards understanding regularization in batch normalization. *ArXiv*, abs/1809.00846, 2019. 1, 3

[23] Boris N Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *arXiv preprint arXiv:1805.10123*, 2018. 3

[24] Cheng Perng Phoo and Bharath Hariharan. Self-training for few-shot transfer across extreme task differences. *arXiv preprint arXiv:2010.07734*, 2020. 2, 3, 5, 6

[25] Sachin Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2

[26] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018. 2

[27] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding propagation: Smoother manifold for few-shot classification. In *European Conference on Computer Vision*, pages 121–138. Springer, 2020. 2

[28] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Mądry. How does batch normalization help optimization? In *Proceedings of the 32nd international conference on neural information processing systems*, pages 2488–2498, 2018. 1, 2, 3

[29] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proceedings of the 31st*

International conference on neural information processing systems, pages 4080–4090, 2017. 1, 2

[30] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019. 2

[31] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018. 2

[32] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 266–282. Springer, 2020. 2

[33] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. 1

[34] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019. 2, 5

[35] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638, 2016. 1, 2, 5

[36] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 3

[37] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019. 2

[38] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12836–12845, 2020. 2

[39] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2