Strong Baselines for Parameter-Efficient Few-Shot Fine-Tuning

Samyadeep Basu[1], **Shell Hu**[3], **Daniela Massiceti**[2], **Soheil Feizi**[1]
[1]University of Maryland, College Park
[2]Microsoft Research, Cambridge
[3]Samsung Research, Cambridge
sbasu12@cs.umd.edu

Abstract

Few-shot classification (FSC) entails learning novel classes given only a few examples per class after a pre-training (or meta-training) phase on a set of base classes. Recent works have shown that simply fine-tuning a pre-trained Vision Transformer (ViT) on new test classes is a strong approach for FSC. Fine-tuning ViTs, however, is expensive in time, compute and storage. This has motivated the design of parameter efficient fine-tuning (PEFT) methods which fine-tune only a fraction of the Transformer's parameters. While these methods have shown promise, inconsistencies in experimental conditions make it difficult to disentangle their advantage from other experimental factors including the feature extractor architecture, pre-trained initialization and fine-tuning algorithm, amongst others. In our paper, we conduct a large-scale, experimentally consistent, empirical analysis to study PEFTs for few-shot image classification. Through a battery of over $1.8k$ controlled experiments on large-scale few-shot benchmarks including META-DATASET (MD) and ORBIT, we uncover novel insights on PEFTs that cast light on their efficacy in fine-tuning ViTs for few-shot classification. Through our controlled empirical study, we have two main findings: (i) Fine-tuning just the LayerNorm parameters (which we call LN-TUNE) during few-shot adaptation is an extremely strong baseline across ViTs pre-trained with both self-supervised and supervised objectives, (ii) For self-supervised ViTs, we find that simply learning a set of scaling parameters for each attention matrix (which we call ATTNSCALE) along with a domain-residual adapter (DRA) module leads to state-of-the-art performance (while being $\sim 9\times$ more parameter-efficient) on MD. Our empirical findings set strong baselines and call for rethinking the current design of PEFT methods for FSC.

1 Introduction

Few-shot classification (FSC) involves learning a new classification task given only a few labelled training examples from each of the novel classes. It has a large number of mainstream applications such as drug-discovery (Stanley et al. 2021), robotics (Ren et al. 2020) and personalized object recognition (Massiceti et al. 2021) among others. Usually, a given few-shot classification task consists of a few-labelled examples from the new classes (support set) and a testing set of unlabeled held-out examples of those classes (query set). Recent works (Hu et al. 2022; Li, Liu, and Bilen 2021; Xu et al. 2022) have shown that fine-tuning a large pre-trained Vision Transformer (ViT) on the support set of new test tasks achieves state-of-the-art performance on large-scale few-shot classification benchmarks such as META-DATASET (MD). Because of their high number of parameters, however, fine-tuning ViTs is extremely expensive in terms of storage, compute, and time. This limits the ability to learn new downstream tasks in real-world applications where resources are constrained (e.g., personalization on edge or mobile devices) since (i) storing the task's fine-tuned parameters on the edge may be unfeasible, especially for a large number of downstream tasks and (ii) fine-tuning on each new task takes long.

As a result, much recent progress has been made in designing light-weight, fast and parameter-efficient fine-tuning (PEFT) methods (Xu et al. 2022; Jia et al. 2022). These reduce the computational requirements to adapt a ViT to a new test task by fine-tuning only a fraction of the ViT's total parameters. However, inconsistencies in experimental setups make it difficult to disentangle the benefit of PEFT methods from other experimental factors, including pre-training initialization, feature extractor architecture, fine-tuning algorithm, downstream dataset and other hyperparameters. Prompt-tuning (Jia et al. 2022), for example, is the state-of-the-art PEFT method on the transfer learning benchmark VTAB (Zhai et al. 2019), while eTT (Xu et al. 2022) performs strongly on few-shot classification in MD. Both, however, use distinct feature extractors, pre-training initializations, fine-tuning algorithms, and hyperparameters, thus limiting our understanding of the generalizability of these PEFT methods across different setups.

To address this, we perform a large-scale empirical analysis of top-performing PEFT methods on two large-scale few-shot image classification benchmarks, META-DATASET (Triantafillou et al. 2019) and ORBIT (Massiceti et al. 2021). Our experimentation involves $\sim 1.8k$ fine-tuning experiments which quantify the performance of PEFT methods under experimentally controlled settings including ViT architectures, pre-training objectives, and fine-tuning algorithms. This enables us to compare PEFT methods in a fair and consistent way and also draw out novel insights on the interaction between these different components in the fine-tuning pipeline.

Our main finding is that the embarrassingly simple approach of fine-tuning just a ViT's LayerNorm parameters

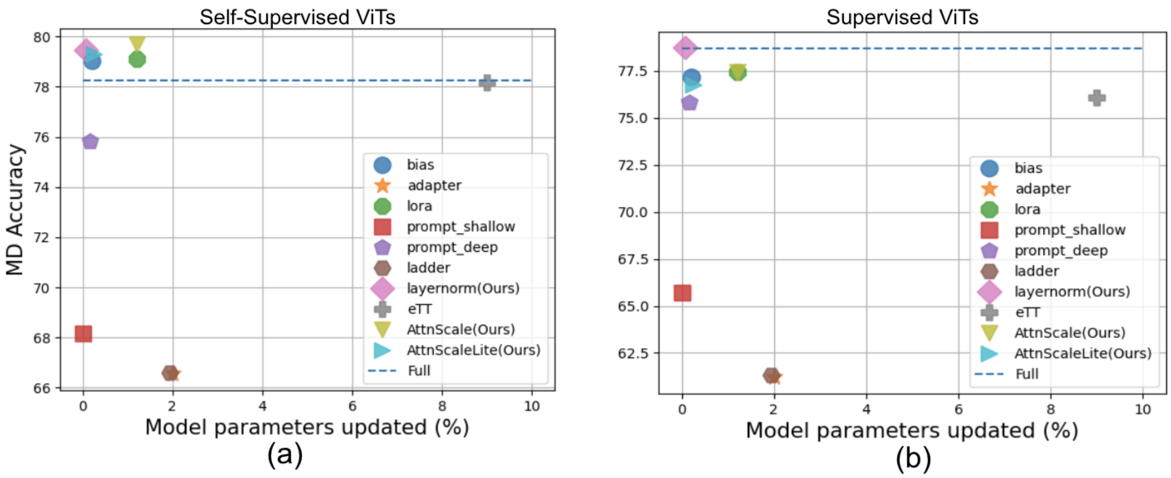

Figure 1: ATTNSCALE leads to SoTA performance on MD with self-supervised ViTs and LN-TUNE leads to SoTA performance for supervised ViTs. Pareto-Plot comparing the average MD accuracy with the model parameters updated during few-shot adaptation: (a) Averaged across self-supervised ViT-S/16 and ViT-B/16 (DINO); (b) Averaged across supervised ViT-S/16(DeiT), ViT-B/16(DeiT) and ViT-B/16(ImageNet-21k). We find that the recently proposed eTT (Xu et al. 2022) does not generalize well to supervised objectives and two simple but *strong* baselines LN-TUNE and ATTNSCALE outperform existing PEFT methods.

(only 0.08% of total parameters) on a new test task leads to better performance than with full model fine-tuning and other PEFT methods on MD and ORBIT. We call this baseline LN-TUNE. We also find that the recently proposed eTT (Xu et al. 2022), primarily designed for self-supervised ViTs, lags behind some of the PEFT methods which we evaluate in our empirical study. In lieu of this, we propose a new strong baseline called ATTNSCALE which leads to improved few-shot performance over eTT and other PEFT methods for self-supervised ViTs. In particular, ATTNSCALE learns only a scaling parameter for each entry in the attention matrices along with a domain-residual module during few-shot adaptation, making it ~ 9x more parameter-efficient than eTT. Importantly, ATTNSCALE is extremely simple to implement, requires less than 6 lines of code, and can be easily integrated with *any* ViT architecture. These approaches establish two new, strong PEFT baselines for few-shot classification, however our empirical study also reveals several interesting insights: (i) None of the carefully designed existing PEFT methods show consistent performance rankings across different pre-training methods (Sec 6.1). (ii) We find that for different degrees of domain shifts, distinct PEFT methods are preferred highlighting that the need for surgically designing PEFT methods for different domain shifts (Sec 6.3). (iii) Dropping PEFT methods from earlier layers in the ViT for large domain shifts (e.g. Omniglot, Quickdraw, Traffic-Sign) is detrimental to few-shot performance (Sec 6.4). In summary, our contributions are as follows:

- A large-scale, experimentally consistent, empirical analysis of a wide-range of PEFT methods for few-shot classification on 2 challenging large-scale benchmarks, META-DATASET and ORBIT.
- An embarrassingly simple PEFT baseline, LN-TUNE, which fine-tunes less than 0.08% of a ViT's parameters outperforming all existing PEFT methods on MD amongst supervised ViTs.
- An easy-to-implement method, ATTNSCALE, which sets a new state-of-the-art on MD amongst self-supervised ViTs while fine-tuning $< 1.2\%$ of the ViT's parameters.

Our findings highlight that there is no one-size-fits-all PEFT method and simple parameter-efficient fine-tuning baselines should not be overlooked.

2 Related Works

ViTs in few-shot classification. CNNs have primarily been used as the feature extractor backbone in few-shot classification methods (Finn, Abbeel, and Levine 2017; Snell, Swersky, and Zemel 2017; Chen et al. 2020; Hospedales et al. 2020; Vinyals et al. 2016), however, recently ViTs have replaced them as the state-of-the-art (Hu et al. 2022) in challenging few-shot classification benchmarks like META-DATASET. In these methods, the ViT is typically pre-trained with a self-supervised (or meta-learning) objective on a large dataset and then fine-tuned on new test tasks.

PEFT methods for few-shot classification. Parameter efficient fine-tuning methods have been extensively studied in Transformers for NLP tasks with adapters (Houlsby et al. 2019), LoRA (Hu et al. 2021), prefix-tuning (Li and Liang 2021) and prompt-tuning (Lester, Al-Rfou, and Constant 2021) serving as strong alternatives to fine-tuning all the Transformer's parameters. PEFTs have also been explored in Vision Transformers for computer vision tasks, with methods like visual prompt tuning (Jia et al. 2022) for transfer learning which work by tuning prefixes attached to the input and eTT (Xu et al. 2022) which tune prefixes attached to key and value matrices in the self-attention layers.

(Xu et al. 2022) show that eTT results in performance close to full model tuning for ViTs pre-trained using DINO using only 9% of the total model parameters on MD.

3 Few-Shot Classification Preliminaries

In few-shot classification, the goal is to adapt a classifier to a new task at test time using a small number of training examples of each new class. In fine-tuning-based approaches, this adaptation process is done by fine-tuning the model on the training examples, before then evaluating it on a held-out set of test examples.

Formally, given a pre-trained feature extractor f_θ, a few-shot task is sampled from a test dataset $\mathcal{D}$. The task is composed of a support set $\mathcal{S}$ (of training examples) and a query set Q (of held-out test examples). Generally, N unique classes are first sampled from the underlying dataset $\mathcal{D}$. For each class $j \in [1, N]$, k_s^j examples are sampled for the support set $\mathcal{S}$ and k_q^j examples are sampled for the query set $\mathcal{Q}$. If $k_s^j = k$ is fixed for $\forall j \in [1, N]$ classes, then the task is known as a N-way, k-shot task. When given a new test task, the objective is to fine-tune the underlying feature extractor f_θ or the parameter-efficient module p_ϕ on the task's support set $\mathcal{S}$ using a fine-tuning algorithm $\mathcal{F}$. In parameter-efficient fine-tuning approaches, f_θ is frozen and only the parameters in p_ϕ are fine-tuned. More specifically, we can formalize the fine-tuning procedure as follows:

$$\phi^* = \min_\phi \ell(f_\theta, p_\phi, \mathcal{F}(\mathcal{S})), \tag{1}$$

Inference on the query examples is done depending on the fine-tuning algorithm $\mathcal{F}$ (see Sec 4) for details). We follow the variable-way, variable way sampling protocol from (Triantafillou et al. 2019) where k_s^j, k_q^j and N vary for each sampled few-shot task. This setting generates class-imbalanced few-shot tasks which make it challenging as the model needs to handle tasks of varying sizes.

4 Large-Scale Empirical Study Design

PEFT methods have been widely used to make few-shot adaptation more computationally efficient (Jia et al. 2022; Xu et al. 2022; Shysheya et al. 2022), however, inconsistencies in experimental setups make it difficult to disentangle the gain from PEFT methods versus other experimental factors. To address this, we conduct a wide-scale experimentally controlled study of over $1.8k$ experiments. We control for the pre-trained model (including pre-training objective and architecture), PEFT module type, position of the PEFT module, fine-tuning algorithm, learning hyperparameters and downstream dataset. Below we provide details of each of these components:

Pre-trained models. For pre-training objectives we consider the self-supervised objective DINO (Caron et al. 2021) and the supervised objective DeiT (Touvron et al. 2020). For architectures, we consider ViT-S/16 and ViT-B/16 (Touvron et al. 2020). These architectures are pre-trained using the given objectives on ImageNet-1k. In addition, we also consider ViT-B/16, which is pre-trained on the large-scale ImageNet-21k. These objectives and architectures were chosen as they lead in downstream few-shot performance (Hu et al. 2022) on MD. More details on pre-training are included in the Appendix.

PEFT methods. We consider the following 7 existing methods for parameter-efficient fine-tuning: adapters (Houlsby et al. 2019), LoRA (Hu et al. 2021), shallow prompt-tuning and deep prompt-tuning (Jia et al. 2022), eTT (Xu et al. 2022), ladder tuning (Sung, Cho, and Bansal 2022), and bias tuning (Zaken, Ravfogel, and Goldberg 2021). We also compare to full model fine-tuning (Hu et al. 2022) and our 2 strong baselines: fine-tuning only the ViT's LayerNorm parameters (LN-TUNE), and learning a simple scaling factor for the elements in the attention matrices (ATTNSCALE) (see Sec 5.2). Of the existing methods, adapters and LoRA have been extensively used for fine-tuning Transformers in few-shot NLP tasks. Ladder tuning is a more recent memory-efficient as well as parameter-efficient fine-tuning method for language models like T5 (Raffel et al. 2019). Ladder is tuning is memory-efficient as it avoids back-propagation through the entire feature-extractor backbone. Shallow and deep prompt tuning are adaptations of (Lester, Al-Rfou, and Constant 2021) for transfer learning in vision. eTT (Xu et al. 2022) fine-tunes only the prefixes attached to the key and value matrices in a ViT's self-attention layers. eTT is also the only method to have been tested on the large-scale META-DATASET benchmark. Note, we omit the prototype regularization used in eTT to ensure fair comparison to other PEFT methods where prototype regularization is not used. We provide further information for each of these methods in the Appendix.

Position of PEFT methods. We consider two configurations in which the PEFTs are inserted in the ViT: (i) We insert PEFTs in each of the layers, including the final; (ii) We insert PEFT in the final layer and in one of the layers between the first and the final layer, leading to two layers in total. For (ii) each fine-tuning experiment is repeated 12 times (see Sec 6.4 for analyses).

Fine-tuning algorithms . We consider 3 fine-tuning algorithms given a new test task: (i) LINEAR: We attach a linear classification layer after the final layer of the ViT and fine-tune both the PEFT's and this layer's parameters using a cross-entropy loss. (ii) PROTOAUG: Following the state-of-the-art fine-tuning approach in (Hu et al. 2022), we use the examples from the task's support set to initialize class prototypes, similar to ProtoNets (Snell, Swersky, and Zemel 2017), and then use a query set to fine-tune the ViT. where the query set is an augmented version of the support set. In particular, we apply color-jitter and translation augmentations on the support set to generate the query set. (iii) PROTONCC: Following (Li, Liu, and Bilen 2021; Xu et al. 2022), we do not apply augmentations to generate the query set and instead treat the query set as a copy of the support set, and fine-tune the ViT in a similar way to PROTOAUG.

Hyperparameters. We standardize the hyperparameters across our entire experimental setup. Following (Hu et al. 2022), we choose a learning rate from $\{0.0001, 0.001, 0.01, 0.1\}$ and select the rate that gives the best performance on the validation set. The validation set is a fixed set of 5 few-shot tasks sampled from the downstream

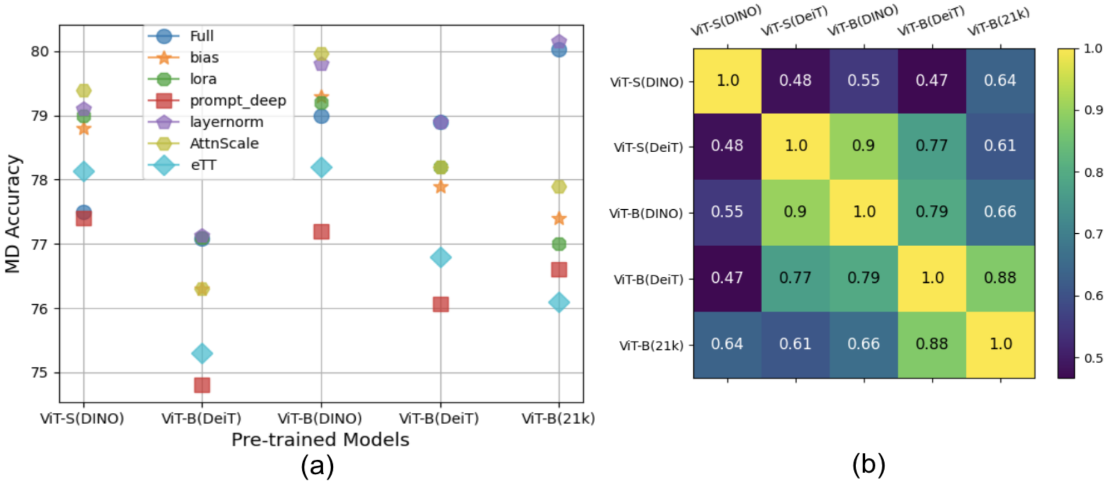

Figure 2: PEFT methods (except LN-TUNE) lack consistency across different pre-training paradigms. (a) The ranks of the 7 top-performing PEFT methods on META-DATASET change across different pre-training paradigms when measured under controlled settings; (b) The Spearman correlations between the different pre-trained models with respect to the performance rank of all 10 PEFT methods are not consistently high. Evaluation across all domains in MD except ImageNet.

dataset to which the ViT is being adapted. For each few-shot task, we fine-tune for 40 steps with Adam (Kingma and Ba 2014) using the selected learning rate.

Downstream datasets. We run all our experiments on two challenging large-scale few-shot classification benchmarks (i) META-DATASET (Triantafillou et al. 2019) and (ii) ORBIT (Massiceti et al. 2021). META-DATASET consists of 10 different sub-datasets, and is currently the most widely used few-shot classification benchmark. Note, we remove the ilsvrc_2012 sub-dataset from META-DATASET as our ViT models have been pre-trained on ImageNet. ORBIT is a few-shot classification benchmark containing noisy, real-world videos of everyday objects across 17 test users. In accordance with (Triantafillou et al. 2019), we sample 600 few-shot tasks per sub-dataset in META-DATASET while for ORBIT, we sample 50 tasks per user. In total, each experimental analysis is performed on 6250 few-shot tasks.

5 Strong Baselines for Few-Shot Fine-tuning

Our standardised large-scale empirical study led us to discover two embarrassingly simple but strong baselines for parameter-efficient few-shot fine-tuning: LN-TUNE and ATTNSCALE. Both of these methods perform better than full model fine-tuning and all other existing PEFT methods on MD at a fraction of the computational cost. Below we describe each of these strong baselines:

5.1 LN-TUNE

LN-TUNE works by fine-tuning *only* the ViT's LayerNorm parameters on a task's support set. Formally, for a given ViT with L layers, the i^{th} layer has two LayerNorm blocks – one before its attention block and one before its MLP block. Given an input vector $a \in \mathbb{R}^d$ from the previous layer or block, the operation of the first block can defined as $\text{LayerNorm}_1^i(\text{a}) = \gamma_1^i \odot (a - \mu)/\sigma + \beta_1^i$, and the operation of the second block as $\text{LayerNorm}_2^i(\text{a}) = \gamma_2^i \odot (a - \mu)/\sigma + \beta_2^i$. Here $\{\gamma_1^i, \beta_1^i, \gamma_2^i, \beta_2^i\} \in \mathbb{R}^d$ are the only learnable parameters for the i^{th} layer. For a given task, these parameters across all L layers are fine-tuned using the task's support set $\mathcal{S}$. As a result, LN-TUNE is extremely light-weight when compared to the other PEFT methods. For e.g., a ViT-S/16 has only $\sim 18.6k$ LayerNorm parameters, while a ViT-B/16 has only $\sim 37k$. Since ViT-S/16 and ViT-B/16 have $\sim$ 22M and $\sim$ 76M parameters, respectively, this accounts for less than 0.08% of the total parameters.

5.2 ATTNSCALE

As a second strong baseline, we introduce ATTNSCALE, a modification to the recently proposed eTT (Xu et al. 2022). Here, we replace the attentive prefix tuning part in eTT with a learnable scaling parameter on each element in the attention matrices, which we tune along with eTT's DRA module, reducing the number of learnable parameters by $\sim$9x. Given a ViT with L layers, n_h attention heads and n tokens, the weight matrices in the i^{th} layer's attention block for the j^{th} head are defined as $W_q^{ij} \in \mathbb{R}^{d \times d_e}$, $W_k^{ij} \in \mathbb{R}^{d \times d_e}$ and $W_v^{ij} \in \mathbb{R}^{d \times d_e}$. Here d is the dimension of the token embeddings and d_e is the dimension of the tokens after the weight matrix projection. $Q^{ij} \in \mathbb{R}^{n \times d}$, $K^{ij} \in \mathbb{R}^{n \times d}$, $V^{ij} \in \mathbb{R}^{n \times d}$ are defined as the query, key and value tokens, respectively. The attention matrix in the i^{th} layer for the j^{th} head can be defined as:

$$A^{ij} = softmax((Q^{ij}W_q^{ij})(K^{ij}W_k^{ij})^T/\sqrt(d_e)), \quad (2)$$

where $A^{ij} \in \mathbb{R}^{n \times n}$. ATTNSCALE applies a point-wise scaling factor to each element in the attention matrix before the softmax operation. These scaling factors are learned during fine-tuning on the task's support set $\mathcal{S}$. In particular,

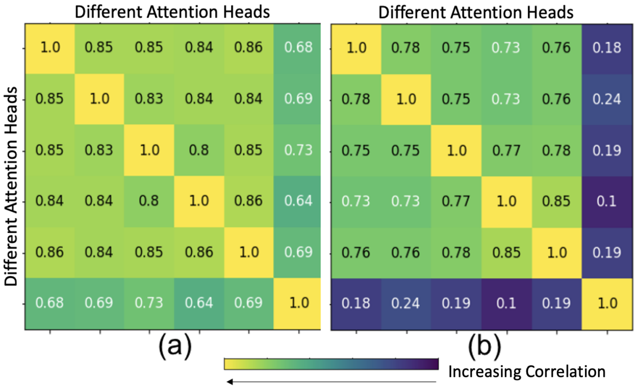

Figure 3: Different attention heads encode similar attention maps in self-supervised ViTs – (a) ViT-S/16(DINO); (b) ViT-S/16(DeiT). We compute the Pearson correlation between the attention scores of different heads: $h\_i, \;\; \forall i \in [1, n_h]$. Self-supervised ViTs encode attention across different heads more similarly than supervised ViTs. Correlation is averaged across examples from 100 tasks in MD.

we define a learnable scaling tensor $A_\alpha \in \mathbb{R}^{n \times n \times L \times n_h}$. A_α can be reshaped as $\{A^i_\alpha\}^L_{i=1}$ where $A^i_\alpha \in \mathbb{R}^{n \times n \times n_h}$ is the scaling tensor for each i^{th} layer. For each attention head $j \in [1, n_h]$, the scaling matrix is defined as $A^{ij}_\alpha \in \mathbb{R}^{n \times n}$.

$$A^{ij} = softmax(A^{ij}_\alpha \odot (Q^{ij}W^{ij}_q)(K^{ij}W^{ij}_k)^T/\sqrt(d_e)), \tag{3}$$

During few-shot adaptation, only A^{ij}_α is learned along with the parameters in the DRA module from eTT. Note, $\{W^{ij}_q, W^{ij}_k, W^{ij}_v\}$ are kept frozen for each i^{th} layer and j^{th} attention head. In principle, the scaling factor A_α replaces the attentive-prefix tuning (APT) module in eTT. This APT module uses ∼9% model parameters, whereas ATTNSCALE uses only ∼1.2% but still gives improved MD performance.

We also propose a light-weight extension of ATTNSCALE, called ATTNSCALELITE, which learns the same scaling parameters across *all* n_h attention heads in a given layer, rather than different ones for each head. This is motivated by an observation that all n_h attention heads in a layer have similar attention maps. We show this in Fig 3 where we plot the pairwise Pearson correlation (Benesty et al. 2009) between the attention values of different heads. Here, for self-supervised ViTs, we see strong correlation values between different heads in a given layer indicating that different heads encode similar kinds of attention maps. This is similar for supervised ViTs, however, the correlation values are slightly lower. Formally, for ATTNSCALELITE, we define the scaling parameter for the i^{th} layer as $A^i_\alpha \in \mathbb{R}^{n \times n}$ and $A^{ij}_\alpha = A^i_\alpha, \;\; \forall j \in [1, n_h]$. ATTNSCALELITE requires only 0.25% of the total parameters for ViT-S/16 and only 0.09% for ViT-B/16 which makes it an extremely light-weight module. In Sec 6, we provide fine-grained results on the efficacy of both ATTNSCALE and ATTNSCALELITE for downstream few-shot adaptation. We provide a PyTorch-like implementation in the Appendix.

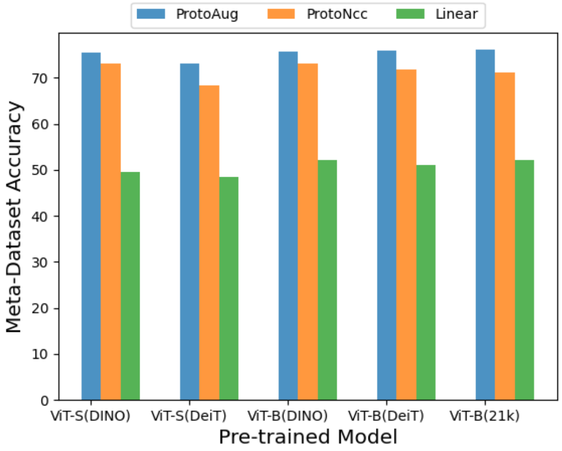

Figure 4: With PEFT methods, we find PROTOAUG to have the best performance on META-DATASET, while LINEAR performs the worst.MD accuracy averaged over all 10 PEFT methods with different fine-tuning algorithms.

6 Empirical Results on META-DATASET

We use our wide-scale empirical study to derive novel insights on PEFT methods for few-shot classification. In particular, we use our results on MD to answer the following key questions: ① Do PEFT methods rank similarly across different pre-training architectures and learning objectives? ② How does the fine-tuning algorithm influence the performance of a PEFT method? ③ Is the optimal PEFT method different for different data domains? ④ Can PEFT modules be dropped from certain positions in the feature extractor? This can lead to significant memory and storage savings during few-shot deployment. These are critical factors when deploying a few-shot classifier in the wild. We also show that our two simple but *strong* baselines, LN-TUNE and ATTNSCALE, perform better than full fine-tuning and all top-performing PEFT methods.

6.1 Consistency Across Pre-Training Models

We analyse the influence of pre-training model by ranking the performance of different PEFT methods across the different pre-training objectives and architectures described in Sec 4. To isolate the role of the pre-trained model, for each run, we keep all other variables constant including the fine-tuning algorithm, position of the modules, and hyperparameters. We report the results using the PROTOAUG fine-tuning algorithm in Fig 2, and include results for PROTONCC and LINEAR in the Appendix.

Existing PEFT methods. In Fig 2-(a), we find that PEFT methods rank inconsistently, with no single best approach, across the different pre-trained models. In Fig 2-(b), we plot the Spearman correlation of the PEFT method's ranking between different pre-trained models. We observe that the correlation values across all pairs of pre-trained models are not consistently high, suggesting that existing PEFT methods do *not* generalize similarly for different pre-trained archi-

PEFT	MSCOCO	Traffic-Sign	Omniglot	Aircraft	DTD	VGG-Flower	Quickdraw	Cu-birds	Fungi	Overall
Full	61.5	87.3	78.7	75.4	86.9	94.2	73.6	85.4	54.7	77.5
Adapter	55.8	52.2	54.7	60.0	83.8	94.6	60.5	84.8	55.9	66.8
Bias	63.4	90.4	80.4	77.5	84.7	95.1	74.3	85.6	58.9	78.8
LoRA	62.1	88.1	80.8	**80.8**	**86.8**	94.8	72.7	85.8	59.8	78.9
Ladder	55.7	52.2	54.7	60.01	83.8	94.6	60.5	84.8	55.9	67.0
Prompt-Shallow	52.7	58.9	61.8	62.9	83.0	94.2	66.0	83.4	55.5	68.7
Prompt-Deep	62.8	85.6	77.0	73.3	85.3	96.2	73.2	86.1	58.2	77.5
eTT	61.5	89.1	78.9	75.8	85.1	95.1	73.5	86.1	58.2	78.1
LN-Tune	**64.2**	91.2	77.9	75.3	84.4	**96.9**	**74.7**	**87.5**	**59.9**	**79.1**
AttnScale	61.9	**91.4**	**80.9**	78.8	85.8	95.9	74.4	86.7	59.01	**79.4**
AttnScaleLite	61.6	91.0	80.2	77.9	85.8	96.0	73.9	86.7	59.0	**79.1**

Table 1: Our strong baselines, LN-Tune and AttnScale, rank in the top 2 of all PEFT methods on the few-shot classification benchmark, META-DATASET. Results shown for a ViT-S/16 (DINO), and exclude the ImageNet split.

tectures and objectives. We also find that adapters, ladder-tuning and shallow prompt-tuning all have sub-par performances on MD ($\sim$ 10% drop) when compared to LoRA, bias-tuning, eTT and deep prompt-tuning. We also highlight that shallow prompt-tuning struggles with few-shot classification on MD despite performing competitively on transfer learning natural tasks in VTAB (Jia et al. 2022). Deep prompt-tuning, which is the state-of-the-art PEFT module on VTAB, performs competitively on MD across all pre-trained models, but falls short of methods like eTT, LoRA, bias-tuning and full model-tuning (see Fig 2). This result highlights that strongly performing PEFT methods for transfer learning *do not* generalize well to the challenging few-shot setting of MD. eTT (Xu et al. 2022) for ViT-S/16(DINO) outperforms full model-tuning, but also lags behind LoRA and bias-tuning. Overall, we find bias-tuning (Zaken, Ravfogel, and Goldberg 2021) to consistently rank amongst the top 4 across all the pre-training models, outperforming many of the more complex PEFT methods.

Our strong baselines. From Fig 2, we find that our strong baselines, LN-Tune and AttnScale, perform strongly across all the pre-trained models on MD. In particular, LN-Tune performs the best for supervised ViTs (pre-trained on ImageNet-1k and ImageNet-21k) consistently. We also highlight that for supervised ViTs, none of the PEFT methods except LN-Tune reaches performance close to full fine-tuning. AttnScale, which is around 9x more parameter-efficient than eTT, has the best few-shot performance for self-supervised ViTs. For self-supervised ViTs, LN-Tune performs closely to AttnScale and ranks in the top 2 methods.

6.2 Effect of Fine-tuning Algorithm

We quantify the impact of 3 different algorithms for fine-tuning the parameters in PEFTs: Linear, ProtoAug and ProtoNCC. We find that ProtoAug outperforms ProtoNCC and strongly outperforms Linear across all pre-training objectives and PEFT methods including full model tuning (Fig 4). In some cases, ProtoAug and ProtoNCC outperform Linear by as much as 20%. We also find that for self-supervised pre-training objectives like DINO (Caron et al. 2021), the gap between ProtoAug and ProtoNCC is $\sim$2.2%, whereas for supervised objectives like DeiT (Touvron et al. 2020) this gap is higher at $\sim$4.7% (for both ImageNet-1k and ImageNet-21k initializations). Since the only difference between ProtoAug and ProtoNCC is that the query set is an augmented version of the support set, this suggests that applying augmentations during few-shot (meta) fine-tuning is more effective with supervised than self-supervised objectives. We also note that when using full model fine-tuning, ProtoAug outperforms ProtoNCC by $\sim$5% for DINO and by $\sim$6.7% for DeiT objectives. This gap is higher than when used with other PEFT methods (see Table 3). This suggests that ProtoAug's efficacy decreases when used in conjunction with PEFT methods.

6.3 Comparing Performance Across Domains

We leverage the distinct sub-datasets in MD to compare the performance of PEFT methods across domains. Since each sub-datasets has a different degree of domains shifts from the pre-training dataset (ImageNet), we also evaluate the robustness of different PEFT methods to these shifts. In Table 1, we show these results with a ViT-S/16 pre-trained with DINO, and observe that none of the PEFT methods are consistently the best across domains. We show similar results for other pre-trained ViTs in the Appendix.

Existing PEFT methods. We observe that deep prompt-tuning is the best PEFT method for domains with smaller degrees of shift from ImageNet such as Cu-Birds and VGG-Flower. It is second best on MS-COCO, which is also similar to ImageNet. We find, however, that for larger domain shifts such as Omniglot, Quickdraw and Traffic-Sign it struggles, with LoRA and bias-tuning showing stronger performance. This is similarly the case for adapters, LoRA, and ladder-tuning which also perform poorly on larger domain shifts and have the lowest average performance on MD generally.

Our strong baselines. We find that LN-Tune in Table 1 outperforms all existing PEFT methods in 5 out of the 9 domains, with AttnScale lagging behind it only slightly in these 5 domains. However, for domains with a larger shift (e.g., Omniglot, Traffic-Sign), AttnScale performs better than LN-Tune. Even for Quickdraw, where there is a significant shift, AttnScale and LN-Tune perform almost similarly. Overall on MD, AttnScale ranks the best

Model	Full	Adapter	Bias	LoRA	Ladder	Prompt-D	Prompt-S	eTT	LN-Tune	AttnScale	AttnScaleLite
ViT-S(DINO)	63.1	62.6	67.1	66.4	62.7	65.7	51.8	65.6	**67.8**	67.2	66.9
ViT-S(DeiT)	66.6	66.8	66.4	67.6	66.9	66.7	63.4	68.4	**68.8**	67.1	66.2

Table 2: LN-Tune results in the best performance on ORBIT while AttnScale is extremely competitive. Prompt-D: Prompt-Deep; Prompt-S: Prompt-Shallow.

Method	ProtoAug	ProtoNCC	Performance Gap
Full Tuning (DINO)	77.2	72.2	Δ **5.0**%
All PEFTs (DINO)	75.4	73.2	Δ 2.2%
Full Tuning (DeiT)	78.1	71.38	Δ **6.7**%
All PEFTs (DeiT)	73.1	68.4	Δ4.7%

Table 3: The performance gap between ProtoAug and ProtoNCC is more with full fine-tuning than when used with PEFT methods.

in terms of few-shot performance. These results suggest that our two strong baselines can be used complementarily: when the domain shift from the pre-training dataset is high, AttnScale is better suited, whereas when the domain shift is low, LN-Tune is the stronger approach. Our results highlight that current PEFT methods are not robust to varying degree of domain shifts and requires rethinking the current designs to be uniformly robust to all domain shifts.

Performance of AttnScaleLite. We observe from Table 1 that AttnScaleLite performs similarly to LN-Tune but slightly worse than AttnScale (by around $0.5 - 0.7\%$) on larger domain shifts for self-supervised ViT-S/16(DINO). For smaller domain shifts, AttnScaleLite matches the performance of AttnScale. For supervised ViTs, we find that AttnScaleLite lags behind AttnScale by a larger margin of $1.2 - 1.8\%$ for large domain shifts (see Appendix for results). The decrease in the effectiveness of AttnScaleLite for supervised ViTs can be attributed to the fact, that different heads encode attention maps less similarly than self-supervised ViTs. Therefore, learning a separate set of scaling parameters for different heads is more beneficial for few-shot adaptation.

6.4 Can We Drop PEFTs from ViT Layers?

In Secs. 6.3 and 6.2, the PEFT modules are inserted in each of the 12 layers of the ViT. In this section, we use our strong baselines to examine if dropping PEFT modules from the majority of layers impacts performance. Specifically, we insert a PEFT module in the final layer of the ViT and another in 1 other layer (between 1-11). We vary the position of the second PEFT and observe its impact on performance (Fig 5).

Results. From Fig 5, we find that inserting the PEFT into the later layers improves the performance more than inserting it in the earlier layers for domains with a small degree of shift from ImageNet (e.g., MSCOCO, DTD, VGG-Flower, Cu_birds). However, for large domain shifts such as in Traffic-Sign, Quickdraw and Omniglot, we find that inserting LN-Tune in the earlier layers is crucial. In particular for these domains, we find that inserting LN-Tune *only* in the later layers results in $\sim10\%$ drop in accuracy.

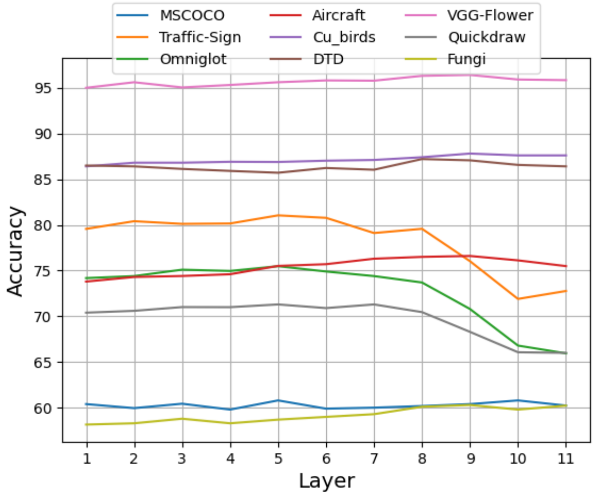

Figure 5: Dropping LN-Tune from earlier layers in the ViT for large domain shifts (e.g., Traffic-Sign, Quickdraw, Omniglot) leads to a large drop in accuracy.

7 Results on Tasks from ORBIT

From Table 2, we find that bias-tuning and eTT have the best performances amongst the existing PEFT methods for ViT-S/16 (DINO) and ViT-S/16 (DeiT), respectively. These results reinforce our previous finding that different PEFT methods may be suited to different pre-training objectives. Overall, we find that LN-Tune results in the best few-shot performance for both self-supervised and supervised objectives across all PEFT methods. AttnScale ranks in the top 2 for DINO, however, for DeiT we find its performance slightly drops but still ranks within the top 4 PEFT methods.

8 Conclusion

In this paper, we perform a large-scale empirical study of a range of top-performing PEFT methods across large-scale benchmarks such as MD and ORBIT. Our main finding is that two embarrassingly simple approaches – LN-Tune and AttnScale – beat all PEFTs we evaluated and set new state-of-the-art results on MD, while being easy-to-implement, significantly less complex and parameter-intensive. The scale of our empirical study also uncovers several novel empirical insights, including that there is no one-size-fits-all PEFT method across different pre-training architectures, objectives, and downstream domains. Together, our experimentally consistent suite of experiments and *strong* baselines supports the future study of PEFT approaches for few-shot classification, but calls for rethinking current practices in light of simple but effective baselines.

Acknowledgements

This project was supported in part by a grant from an NSF CAREER AWARD 1942230, ONR YIP award N00014-22-1-2271, ARO's Early Career Program Award 310902-00001, Meta grant 23010098, HR00112090132 (DARPA/RED), HR001119S0026 (DARPA/GARD), Army Grant No. W911NF2120076, NIST 60NANB20D134, the NSF award CCF2212458, an Amazon Research Award and an award from Capital One.

References

Benesty, J.; Chen, J.; Huang, Y.; and Cohen, I. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*, 37–40. Springer.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. *CoRR*, abs/2104.14294.

Chen, Y.; Wang, X.; Liu, Z.; Xu, H.; and Darrell, T. 2020. A New Meta-Baseline for Few-Shot Learning. *CoRR*, abs/2003.04390.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *CoRR*, abs/1703.03400.

Hospedales, T. M.; Antoniou, A.; Micaelli, P.; and Storkey, A. J. 2020. Meta-Learning in Neural Networks: A Survey. *CoRR*, abs/2004.05439.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; de Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-Efficient Transfer Learning for NLP.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *CoRR*, abs/2106.09685.

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.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual Prompt Tuning.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. *CoRR*, abs/2104.08691.

Li, W.-H.; Liu, X.; and Bilen, H. 2021. Cross-domain Few-shot Learning with Task-specific Adapters.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. *CoRR*, abs/2101.00190.

Massiceti, D.; Theodorou, L.; Zintgraf, L.; Harris, M. T.; Stumpf, S.; Morrison, C.; Cutrell, E.; and Hofmann, K. 2021. ORBIT: A real-world few-shot dataset for teachable object recognition collected from people who are blind or low vision.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *CoRR*, abs/1910.10683.

Ren, M.; Iuzzolino, M. L.; Mozer, M. C.; and Zemel, R. S. 2020. Wandering Within a World: Online Contextualized Few-Shot Learning. *CoRR*, abs/2007.04546.

Shysheya, A.; Bronskill, J.; Patacchiola, M.; Nowozin, S.; and Turner, R. E. 2022. FiT: Parameter Efficient Few-shot Transfer Learning for Personalized and Federated Image Classification.

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical Networks for Few-shot Learning.

Stanley, M.; Bronskill, J. F.; Maziarz, K.; Misztela, H.; Lanini, J.; Segler, M.; Schneider, N.; and Brockschmidt, M. 2021. FS-Mol: A Few-Shot Learning Dataset of Molecules. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Sung, Y.-L.; Cho, J.; and Bansal, M. 2022. LST: Ladder Side-Tuning for Parameter and Memory Efficient Transfer Learning.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2020. Training data-efficient image transformers & distillation through attention. *CoRR*, abs/2012.12877.

Triantafillou, E.; Zhu, T.; Dumoulin, V.; Lamblin, P.; Xu, K.; Goroshin, R.; Gelada, C.; Swersky, K.; Manzagol, P.; and Larochelle, H. 2019. Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples. *CoRR*, abs/1903.03096.

Vinyals, O.; Blundell, C.; Lillicrap, T. P.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. *CoRR*, abs/1606.04080.

Xu, C.; Yang, S.; Wang, Y.; Wang, Z.; Fu, Y.; and Xue, X. 2022. Exploring Efficient Few-shot Adaptation for Vision Transformers. *Transactions of Machine Learning Research*.

Zaken, E. B.; Ravfogel, S.; and Goldberg, Y. 2021. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. *CoRR*, abs/2106.10199.

Zhai, X.; Puigcerver, J.; Kolesnikov, A.; Ruyssen, P.; Riquelme, C.; Lucic, M.; Djolonga, J.; Pinto, A. S.; Neumann, M.; Dosovitskiy, A.; Beyer, L.; Bachem, O.; Tschannen, M.; Michalski, M.; Bousquet, O.; Gelly, S.; and Houlsby, N. 2019. The Visual Task Adaptation Benchmark. *CoRR*, abs/1910.04867.