Unleashing the Power of Meta-tuning for Few-shot Generalization Through Sparse Interpolated Experts

Shengzhuang Chen [1] **Jihoon Tack** [2] **Yunqiao Yang** [1] **Yee Whye Teh** [3] **Jonathan Richard Schwarz** [* 4] **Ying Wei** [* 5]

Abstract

Recent successes suggest that parameter-efficient fine-tuning of foundation models is becoming the state-of-the-art method for transfer learning in vision, gradually replacing the rich literature of alternatives such as meta-learning. In trying to harness the best of both worlds, meta-tuning introduces a subsequent optimization stage of foundation models but has so far only shown limited success and crucially tends to underperform on out-of-distribution (OOD) tasks. In this paper, we introduce Sparse MetA-Tuning (SMAT), a method inspired by sparse mixture-of-experts approaches and trained to isolate subsets of pre-trained parameters automatically for meta-tuning on each task. SMAT successfully overcomes OOD sensitivity and delivers on the promise of enhancing the transfer abilities of vision foundation models beyond parameter-efficient finetuning. We establish new state-of-the-art results on a challenging combination of Meta-Dataset augmented with additional OOD tasks in both zero-shot and gradient-based adaptation settings. In addition, we provide a thorough analysis of the superiority of learned over hand-designed sparsity patterns for sparse expert methods and the pivotal importance of the sparsity level in balancing between in-distribution and out-of-distribution generalization. Our `code` and `models` are publicly available.

1. Introduction

The emergence of foundation models (Bommasani et al., 2021) has marked a new chapter in machine learning, with

Figure 1. Average testing accuracy during meta-training for meta-tuning methods. SMAT yields better in-distribution (ID) and out-of-distribution (OOD) results and shows an attractive learning speedup.

pre-trained models in established domains (e.g., vision or language) becoming virtually indispensable and a vibrant research landscape developing around the design and training of foundation models for new modalities and problems, ranging from the life sciences (Lin et al., 2023) to spectral data (Hong et al., 2023), time series (Yeh et al., 2023), graphs (Liu et al., 2023a) and combinations thereof in multi-modal systems (Yu et al., 2023). Foundation models have seemingly also signaled the convergence of decades of research on transfer learning (see Zhuang et al. (2020) for a survey) to the simple yet powerful paradigm of full or parameter-efficient fine-tuning (Perez et al., 2018; Hu et al., 2021) of the best foundation model available. As with many breakthroughs in science, this convergence runs contrary to an attractive hypothesis: That the explicit formulation of objectives, algorithms, and optimization procedures targeted directly at downstream performance will result in the best transfer learner (most directly advocated for in the learning to learn, or meta-learning community (Thrun & Pratt, 2012; Finn et al., 2017)). Instead, the field's general belief has shifted towards self-supervised objectives such as autoregressive losses (e.g. Mikolov et al., 2013) or contrastive learning (e.g. Radford et al., 2021) with large models and big data as the best strategy for generalist models with the

---

*Equal contribution [1]City University of Hong Kong [2]Korea Advanced Institute of Science and Technology [3]University of Oxford [4]Harvard University [5]Nanyang Technological University. Correspondence to: Shengzhuang Chen <szchen9-c@my.cityu.edu.hk>, Jonathan Richard Schwarz <schwarzjn@gmail.com>, Ying Wei <ying.wei@ntu.edu.sg>.

potential to transfer to a wide variety of tasks.

In this paper, rather than committing fully to this view, we instead join a nascent group of researchers aiming to find a middle point between both paradigms, harnessing their strengths and aiming to find a synergy. Indeed, the recent popularity of instruction tuning for large language models (Zhang et al., 2023) takes a similar view and has emerged as a promising avenue to not only narrow the gap between pre-training and downstream objectives but also enhance zero-shot generalization of pre-trained models. Similarly, *meta-tuning* aims to enhance the transferability of foundation models through a secondary meta-learning stage initiated once pre-training has converged. Indeed, existing research in the field of Natural Language Processing (NLP) has substantiated the advantages of meta-tuning over traditional fine-tuning and transfer learning approaches, particularly in zero-shot and few-shot testing scenarios (Gao et al., 2021; Min et al., 2021; Chen et al., 2021).

Despite initial progress made, the exploration of meta-tuning in vision still remains notably limited to date. Hu et al. propose a pre-training → meta-training → fine-tuning pipeline, dubbed PMF, for enhancing the few-shot learning performance of the resulting model relative to the default pre-training → fine-tuning approach. With the design principle of simplicity in mind, PMF meta-trains all parameters in a vision transformer using Prototypical Networks (Snell et al., 2017) starting from a pre-trained initialization, yielding the state-of-the-art performance on popular meta-learning benchmarks such as Meta-Dataset (Triantafillou et al., 2020). Despite such promising reported results, we find that this intuitive approach tends to underperform on downstream few-shot tasks, particularly when testing for out-of-distribution (OOD) tasks (i.e., tasks dissimilar to the ones presented during the meta-training stage).

We hypothesize that this low generalization performance on OOD tasks stems from two major factors. (i) The strong emphasis on learning from small amounts of data using a limited number of optimization steps in meta-learning can lead to algorithms that are "greedy" w.r.t. the distribution of tasks presented, sacrificing more generalizable features for performance on the distribution at hand. This leads to a risk of meta-overfitting, a phenomenon previously observed (e.g. Zintgraf et al., 2019; Yao et al., 2021; Chen et al., 2022). (ii) When meta-tuning tasks are diverse, the default setting of updating all parameters suffers from task interference, making optimization unstable and thereby reducing generalization performance.

In introducing our method, we thus explicitly design core model components to overcome these challenges. We address (i) by taking inspiration from recent work (Ilharco et al., 2022; Panigrahi et al., 2023a; Wortsman et al., 2021), noticing that interpolation between pre-trained and fine-tuned weights leads to a trade-off between ID and OOD generalization performance, with an optimal point usually existing between the extremes. We implement this trade-off through a learned gated interpolation implemented with a sparsity constraint. This particular choice also has the added benefit of addressing (ii) by considering a Mixture-of-Experts inspired approach (with each expert defined through sparse masks), which guarantees expressiveness while alleviating task interference. Finally, sparsity has an additional regularizing effect, further reducing the chance of meta-overfitting and thus counteracting poor OOD generalization observed for standard Meta-Tuning (see Figure 1 for a direct comparison with the aforementioned PMF).

In summary, we propose a reformulation of meta-tuning as a process wherein a hypernetwork undergoes meta-training to select a combination of sparse experts based on few-shot examples, which are subsequently interpolated with the pre-trained model to tailor a powerful foundation model for downstream performance on each specific task. The integration of an interpolation strategy alongside specialized experts not only preserves the pre-trained model's generalization capabilities but also consolidates the knowledge acquired from all meta-tuned tasks without interference. This synergy contributes significantly to our *strong performance across both in-distribution and out-of-distribution few-shot generalization scenarios*. Furthermore, we showcase the *interpretability* on task relationship through the experts selected, and *compatibility* of our approach with both full fine-tuning and parameter-efficient fine-tuning methods, such as LoRA (Hu et al., 2021).

2. Related Work

Few-shot learning and meta-tuning. Much of few-shot learning (FSL) relies on meta-learning across so-called N-way-K-shot meta-training episodes, which can then be utilized for data-efficient learning during (Hospedales et al., 2020). Meta-learned inductive biases may take the form of a model initialization (Finn et al., 2017), a learned metric (Snell et al., 2017), a Bayesian prior (Grant et al., 2018) or an optimization strategy (Li et al., 2017). A particular sub-domain of FSL, namely, cross-domain FSL algorithms (Li et al., 2022; Triantafillou et al., 2020; Liu et al., 2021; Bateni et al., 2019), explicitly deals with task-distribution shift between meta-training and -testing. Nevertheless, most architectures used in existing work are limited in scale and without large-scale pre-training. Transitioning into the LLM era, (Min et al., 2021) first study meta-training a pre-trained LLM on a large collection of few-shot in-context learning tasks. Their results highlight the effectiveness of meta-training on improving few-shot in-context learning generalization of powerful pre-trained transformers; motivating several follow-up works in the field of meta-tuning

in NLP (Gao et al., 2021; Min et al., 2021; Chen et al., 2021). In computer vision, Hu et al. propose the simple pre-training, meta-training then fine-tuning (PMF) pipeline and achieve SOTA performance (2022).

Sparse mixture-of-experts (MoE) The key idea of sparse MoE is the selective activation of expert modules, usually MLP layers, for each input token during training and inference, thereby achieving graceful scaling. Earlier MoE methods make discrete expert-to-token assignments through a token-choice router scoring experts and selecting the top-k for each token (Shazeer et al., 2017; Riquelme et al., 2021; Lepikhin et al., 2021; Fedus et al., 2021). Alternatively, methods may choose the top-k scored tokens for each expert (Zhou et al., 2022), use stochastic or fixed routers(Roller et al., 2021; Zuo et al., 2022), and more advanced routing techniques (Lewis et al., 2021; Liu et al., 2023b). However, the discrete nature of the assignment poses a serious challenge to stability in training and optimization (Mustafa et al., 2022; Dai et al., 2022). To this end, more recent works on soft MoE consider a soft relaxation or approximation to the otherwise discrete expert-token assignment (Puigcerver et al., 2023), as well as other MoE works that employ a weighted sum of experts in the parameter space (Muqeeth et al., 2023).

Partitioned meta-learning. The importance of isolating a subset of parameters with high plasticity for optimization-based meta-learning is well-established. A common feature is thus the partitioning of parameters into a set of shared parameters optimized in the outer loop and a (typically smaller) parameter set that implements task adaptation, thus reducing meta-overfitting and memory usage. Early work in this direction (e.g. Raghu et al., 2019; Oh et al., 2020) rely on partitioning heuristics (such as only updating the last layer) or introduce additional context parameters (Zintgraf et al., 2019) which are concatenated with the input vector.

Since then, an increasing amount of attention has been placed on adapters, i.e. compact, parameter-efficient modules which have been shown to be particularly impactful when fine-tuning foundation models, thus being particularly suitable for Meta-Tuning. Popular adapter types such as FiLM (Perez et al., 2018), LoRA (Hu et al., 2021) as well as various alternatives feature in various meta-learning and episodic-training methods (e.g. Requeima et al., 2019; Triantafillou et al., 2021a; Shysheya et al., 2022; Schwarz et al., 2023).

Most closely related to our method is a line of work utilizing sparsification to isolate and train a subset of parameters for rapid adaptation, thus increasing their plasticity for fine-tuning. Most closely related to our method is the aforementioned MSCN (Schwarz & Teh, 2022), although the authors focus their experiments on a more specialized compression problem and do not address the problem of how to tackle a specific sparsity level. Alternative approaches feature sparsification in the outer loop through magnitude-based pruning (Lee et al., 2021), which, while simple, may overly constrain the representational capacity. Similar to over hyper-network inspired approach (Schwarz et al., 2023) predict sparse masks that index into model weights, although they still rely on second-order meta-learning. Finally, the work in (Von Oswald et al., 2021) presents a first-order method for gradient sparsity, demonstrating the approach in traditional meta-learning as well as Continual Learning.

3. Preliminary: Meta-tuning

Meta-tuning aims to improve the few-show learning performance of a pre-trained model on downstream few-shot testing tasks - usually by directly meta-training the pre-trained model over a collection of labeled few-shot training task episodes (Hu et al., 2022; Min et al., 2021).

Specifically, we assume the availability of a pre-trained model $f_{\theta^{\text{pre}}}$ as initialization for meta-training, and a training task distribution $\mathcal{P}_{ID}(\mathcal{T})$ from which we may sample fully labeled few-shot training tasks $\mathcal{T}_i \sim \mathcal{P}_{ID}(\mathcal{T})$. Note that we explicitly denote this as in-distribution (ID). In particular, in the supervised setting, each training task $\mathcal{T}_i$ takes the form of $\mathcal{T}_i := \{\mathcal{L}_i, \mathcal{T}_i^s, \mathcal{T}_i^q\}$, where $\mathcal{L}_i$ is the task loss to be minimized, $\mathcal{T}_i^s := \{\mathbf{x}_{i,j}^s, \mathbf{y}_{i,j}^s\}_{j=1}^{N_i^s}$ and $\mathcal{T}_i^q := \{\mathbf{x}_{i,j}^q, \mathbf{y}_{i,j}^q\}_{j=1}^{N^q}$ are labeled support and query sets of N_i^s, N_i^q input-target pairs, respectively. We use the shorthand notations $\mathbf{X}$ and $\mathbf{Y}$ to represent a set of inputs and labels, respectively. Meta-tuning is then realized through the typical episodic-learning setting familiar from meta-learning, i.e., the direct optimization of θ on the few-shot learning objective which considers minimizing the task loss on the query predictions given information of the support set, i.e., $\theta^* := \arg\min_{\theta \in \Theta} \mathbb{E}_{\mathcal{P}_{ID}}[\mathcal{L}_i(f_\theta(\mathbf{X}_i^q, \mathcal{T}_i^s), \mathbf{Y}_i^q)]$, over training task episodes sampled from $\mathcal{P}_{ID}(\mathcal{T})$. At test time, we expect to encounter both ID and OOD testing tasks i.e., $\tilde{\mathcal{T}} := \{\tilde{\mathcal{T}}_i^s, \tilde{\mathbf{X}}_i^q\} \sim \mathcal{P}_{ID} \cup \mathcal{P}_{OOD}$, where $\mathcal{P}_{OOD} \neq \mathcal{P}_{ID}$ is an unseen OOD task distribution. For each testing task, we are given a few labeled support samples and unlabeled query inputs to assess the few-shot generalization performance of the meta-tuned model. Hence, our objective is to develop a meta-tuning algorithm that enables the meta-tuned θ^* to attain optimal few-shot generalization performance across both ID and OOD downstream testing tasks.

4. SMAT: Sparse MetA-Tuning

4.1. Meta-training with SMAT

As discussed in Section 1, naively sharing and updating all pre-trained parameters across all tasks in meta-tuning leads to task interference in optimization (Yu et al., 2020; Wang et al., 2020). To address this issue, we instead hypothesize
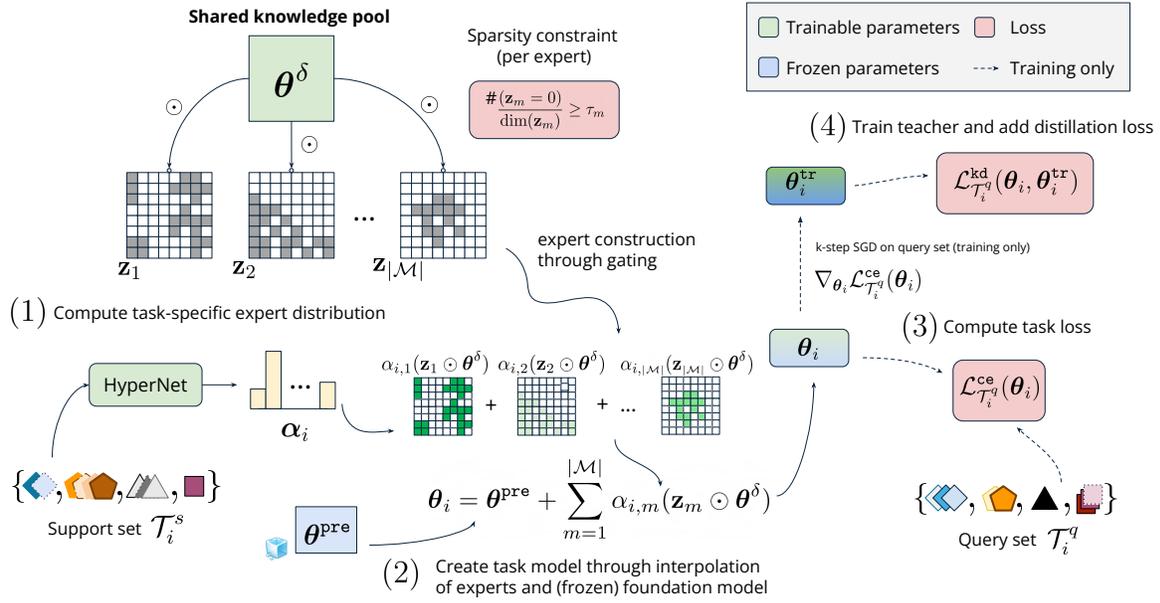
Figure 2. Overview of the proposed **S**parse **M**et**A**-**T**uning approach, showing meta-training and inference procedures for a single task $\mathcal{T}_i$. SMAT meta-learns a shared knowledge pool $\mathcal{M}$ consisting of $|\mathcal{M}|$ sparse interpolated experts characterized by a **common**, learnable set of dense parameters $\boldsymbol{\theta}^\delta$ and **distinct**, learnable sets of gating masks $\{\boldsymbol{z}_m\}_{m=1}^{|\mathcal{M}|}$ with sparsity constraints. To construct each task-specific model $\boldsymbol{\theta}_i$ for both meta-training and inference, **(1)** SMAT first combines experts via a weighted-sum with merging weights $\boldsymbol{\alpha}_i$ generated by a meta-learned hypernetwork h_ζ based on the task's support set $\mathcal{T}_i^s$. **(2)** The experts are then subsequently combined with the frozen pre-trained model $\boldsymbol{\theta}^{\text{pre}}$ to enhance both in-distribution (ID) and out-of-distribution (OOD) generalization performance. Alongside **(3)** the query prediction loss $\mathcal{L}_{\mathcal{T}_i^q}^{\text{ce}}(\boldsymbol{\theta}_i)$, **(4)** knowledge distillation with task-specific dense teachers $\mathcal{L}_{\mathcal{T}_i^q}^{\text{kd}}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^{\text{tr}})$ is introduced during meta-training to promote specialization and cooperation of the sparse interpolated experts, ensuring optimization success.

that the solution for each task (ID or OOD) comprises a task-specific mixture of a common pool of knowledge covering a broad range of tasks. The knowledge pool is represented by distinct sets of model parameters (i.e., experts), which can be combined cooperatively as a complementary addition to the pre-trained model to promote systematic generalization. Formally, we assume that each task-specific model $\boldsymbol{\theta}_i$ is derived from aggregating the experts via a task-specific weighted sum in the parameter space:

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}^{\text{pre}} + \sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m} \boldsymbol{\theta}_m^\delta, \quad (1)$$

where $\boldsymbol{\theta}_m^\delta$ represents the m-th expert in the pool $\mathcal{M}$, and $\boldsymbol{\alpha}_i := [\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,|\mathcal{M}|}]$ are the weights. This way of merging experts in Eqn. (1) is appealing due to its high expressiveness, as supported by recent findings revealing that merging multiple sets of specialized parameters through simple arithmetic can lead to a better overall multi-task solution (Ilharco et al., 2023; Matena & Raffel, 2021). Moreover, the complementary addition of experts to the pre-trained backbone effectively decouples the two for optimization - meta-training $\{\boldsymbol{\theta}_m^\delta\}^{|\mathcal{M}|}$ while keeping $\boldsymbol{\theta}^{\text{pre}}$ frozen provides sufficient capacity and preserves pre-trained knowledge.

That said, meta-tuning, in our case, is seen as a procedure for discovering a generalizable selection rule for assigning appropriate experts to tasks. We present an overview of our proposed method in Fig. 2.

Sparsification of experts. At the core of our method lies the question on **where** and **how** to find a set of experts in the model during meta-tuning. Previous studies on partitioned meta-learning, as well as recent works on Mixture-of-Experts, propose a seemingly reasonable solution: Experts are incorporated into a heuristically (and a-priori) chosen *subset* of modules in the model. For example, the batch-norm (BN) layers in ResNets (Triantafillou et al., 2021b) or the MLP layers in Vision Transformers (ViT) (Puigcerver et al., 2023). Although these approaches based on fixed partitioning are more memory-efficient and generally result in improved performance compared to their non-expert counterparts, they suffer from a significant bias due to their manually crafted selection (i.e., the concrete choice of $\boldsymbol{\theta}_m^\delta$). This bias may be suboptimal when it conflicts with the intricate dynamics of meta-training and, in addition, is not always model-agnostic, e.g., there are no BN layers in ViT.

Thus, instead of specifying experts as prior knowledge, we propose to automatically identify the sparsity patterns in ex-

perts through meta-tuning using a maximum likelihood objective while considering sparsity constraints on the experts' capacity, thus encouraging specialization. Meta-learning sparsity patterns not only shows minimal bias but also allows for differentiation in the architecture of experts, giving our model a wider range of capabilities to better handle various types of distribution shifts. (Lee et al., 2022).

Specifically, we induce a learnable sparsity pattern in the m-th expert in the form of a sparse reparameterization $\boldsymbol{\theta}_m^\delta \odot \boldsymbol{z}_m$ of the dense expert, where $\boldsymbol{z}_m$ denotes a learnable binary mask with dimension $|\boldsymbol{z}_m| = |\boldsymbol{\theta}_m^\delta|$, and $\odot$ denotes element-wise multiplication. To learn binary masks through gradient-based optimization, we take inspiration from (Schwarz & Teh, 2022), and reparameterize the m-th binary mask using an underlying continuous distribution $q_{\boldsymbol{\phi}_m}$ with parameters $\boldsymbol{\phi}_m$. The reparameterization samples, i.e., $\mathbf{s}_m \sim q_{\boldsymbol{\phi}_m}(\mathbf{s})$, can be transformed to have values *exactly* 0 and 1 through a hard rectification $\mathbf{z}_m = g(\mathbf{s}_m) := \min(1, \max(0, \mathbf{s}_m))$. As a result, sparsity in $\boldsymbol{z}_m$ can be enforced by limiting the probability of $\boldsymbol{s}_m$ being non-zero which can be straight-forwardly expressed through the CDF Q of $q_{\boldsymbol{\phi}_m}$, i.e. $1 - Q_{\boldsymbol{\phi}_m}(\mathbf{s} \le 0)$. Choosing $q_{\boldsymbol{\phi}_m}$ as the stretched hard concrete distribution (Louizos et al., 2018) enables both gradient-based optimization through reparameterization as well as analytical evaluation of the CDF.

From sparse experts to sparse interpolated experts. Though the aforementioned sparse reparameterization eventually reduces the final parameter counts by removing zeros in expert parameters, it also leads to a strong increase in the number of parameters compared to its non-sparse counterpart at the beginning of meta-tuning – due to learning $\boldsymbol{z}_m$ and $\boldsymbol{\theta}_m^\delta$ concurrently. Moreover, assigning each expert its own dense underlying parameters $\boldsymbol{\theta}_m^\delta$ greatly hinders knowledge transfer among experts, which contradicts the principles of partitioned meta-learning emphasizing collaboration of both task-agnostic and task-specific components.

To tackle these issues, we propose sharing dense parameters across sparse reparameterization of different experts, i.e., $\boldsymbol{\theta}_m^\delta = \boldsymbol{\theta}^\delta, \forall m$. By rearranging Eqn. (1) slightly, the experts now essentially become different sparse interpolations between the same pre-trained and meta-tuned models (more details in Appendix D). Although this formulation of MoE may initially appear bold, it is well-supported by recent works (Panigrahi et al., 2023b) which suggest that multiple task-specific optimal points can coexist between the same set of pre-trained and meta-tuned models, offering favorable trade-offs for both in-distribution (ID) and some out-of-distribution (OOD) performance. Therefore, our approach of learning sparse interpolated experts can be seen as an inductive bias that promotes the recovery of these optimal interpolation points through meta-tuning. Favorably, this

partitioning allows for knowledge transfer among experts through the overlapping regions in their masks.

Meta-learning expert selection through a hypernetwork. In theory, there are two possible approaches to achieve task-specific inference for the expert merging scores $\boldsymbol{\alpha}_i$. The first one is a meta-learned merging score initialization $\boldsymbol{\alpha}_0$ combined with inner-loop gradient-descent on the task support set; the second involves a meta-learned global hypernetwork, $h_{\boldsymbol{\zeta}}(\mathcal{T}_i^s)$, parameterized by $\boldsymbol{\zeta}$, that directly outputs $\boldsymbol{\alpha}_i$ conditioned on the task support set $\mathcal{T}_i^s$. We opt for the latter approach as it scales better with larger model sizes. For implementation, we use the pre-trained model $f_{\boldsymbol{\theta}^{\mathrm{pre}}}$ to encode each support image $\mathbf{x}_{i,j}^s \in \mathcal{T}_i^s$ into a vector embedding $f_{\boldsymbol{\theta}^{\mathrm{pre}}}(\mathbf{x}_{i,j}^s)$. The support set embeddings are aggregated into class prototypes which are then concatenated into a sequence and fed into a single trainable transformer block to obtain $\boldsymbol{\alpha}_i' \in \mathbb{R}^{|\mathcal{M}|}$ as the output. We treat $\boldsymbol{\alpha}_i'$ as the logits for activating the experts. Instead of choosing the top-k activated ones, which can cause training instability issues and, more importantly, restrict the number of experts per task, we employ the Gumbel-Sigmoid trick (Jang et al., 2017) to sample a soft activation value $\in (0, 1)$ for each expert, followed by normalization to obtain $\boldsymbol{\alpha}_i \in (0, 1)^{|\mathcal{M}|}$.

Enhancing expert specialization through task-specific dense teachers. Specialization and cooperation among experts play a crucial role in a MoE model. One way to promote specialization is by penalizing the similarity between experts. For example, an orthogonal penalty can be applied to pairs of experts. However, incorporating such explicit penalties makes the optimization problem in the meta-objective more challenging, as it introduces a trade-off with respect to the few-shot prediction performance. To circumvent this trade-off, we propose an alternative approach that utilizes a knowledge distillation loss $\mathcal{L}_{\mathcal{T}_i^q}^{\mathrm{kd}}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i^{\mathrm{tr}})$ (Hinton et al., 2015) between the merged MoE model $\boldsymbol{\theta}_i$ and a teacher network $\boldsymbol{\theta}_i^{\mathrm{tr}}$. By using a *highly task-specific* teacher, the distillation loss imposes the weighted-sum of experts that constitute each $\boldsymbol{\theta}_i$ to mimic the behaviour of the teacher, thus enhancing both specialization and cooperation implicitly through knowledge transfer. Furthermore, the predictive performance of $\boldsymbol{\theta}_i$ is not impeded as the loss consistently encourages the current student to achieve a better task-specific generalization on the query set, similar to that of the teacher.

We generate the teacher $\boldsymbol{\theta}_i^{\mathrm{tr}}$ dynamically in each meta-training episode by performing K-step gradient descent starting from $\boldsymbol{\theta}_i$ on the query loss $\mathcal{L}_{\mathcal{T}_i^q}^{\mathrm{ce}}$ (we find $K = 1$ is sufficient in practice). It is worth noting that unlike $\boldsymbol{\theta}_i$, we do not limit the capacity of $\boldsymbol{\theta}_i^{\mathrm{tr}}$ through sparse regularization. This results in a teacher model that is both dense in modulation (i.e., $\boldsymbol{\theta}_i^{\mathrm{tr}} - \boldsymbol{\theta}^{\mathrm{pre}}$), making it expressive and highly task-specific. Importantly, we do not propagate the loss gradients through $\boldsymbol{\theta}_i^{\mathrm{tr}} \rightarrow \boldsymbol{\theta}_i$. This approach is thus

similar to the use of bootstrapping in optimization-based meta-learning (Flennerhag et al., 2022; Tack et al., 2024).

Meta-optimization with controlled expert sparsity. We are now ready to state our optimization problem for meta-tuning. To enable precise control of expert sparsity- which is pivotal for controlling the trade-offs between ID and OOD generalization performance, we solve an optimization problem for few-shot learning performance under sparsity constraints, namely, $1 - \frac{L_0(z_m)}{\dim(z_m)} \geq \tau_m, \forall m \in \big[|\mathcal{M}|\big]$ where $\tau_m \in [0, 1]$ are the targeted sparsity levels. In practice, we optimize the Lagrangian associated with the constraint optimization problem (with Lagrangian multipliers λ) during meta-tuning:

$$\min_{\theta^\delta, \zeta, \Phi} \max_{\lambda \geq 0} \mathbb{E}_{\mathcal{T}_i \sim \mathcal{P}_{ID}} \big[\mathcal{L}^{\mathrm{ce}}_{\mathcal{T}_i^q}(\theta_i) + \mathcal{L}^{\mathrm{kd}}_{\mathcal{T}_i^q}(\theta_i, \theta_i^{\mathrm{tr}}) \big]$$

$$+ \sum_{m=1}^{|\mathcal{M}|} \lambda_m \big(\frac{1}{|\phi_m|} \sum_{k=1}^{|\phi_m|} \tau - Q_{\phi_m}(s_k \leq 0) \big),$$

$$\text{where} \quad \theta_i = \theta^{\mathrm{pre}} + \sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m}(z_m \odot \theta^\delta),$$

$$z_m \sim q_{\phi_m}; \alpha_i \sim h_\zeta(\mathcal{T}_i^s), \qquad (2)$$

in which the objective is the minimization problem in the first line while the aforementioned sparsity constraints translate to the maximization problem in the second. We put the sparsity constraints on individual masks, hence experts, as opposed to the overall θ_i to allow task-dependent model capacity in θ_i through selective merging. For simplicity, we set a common constraint for all masks i.e., $\tau_m = \tau, \forall m$, which is treated as a hyperparameter. We use simultaneous gradient descent and projected gradient ascent for optimizing Eqn. (2). To avoid over-penalizing the model capacity from surpassing the sparsity constraints, we reset λ_m to zero after its associated sparsity constraint is met (Gallego-Posada et al., 2022). This results in final sparsity close to the target τ, enabling precise control of the expert sparsity levels. The pseudocodes for meta-training can be found in Appendix A.1.

4.2. Meta-testing time adaptation

Gradient-free optimization for expert selection. Prior work has discovered that further task-specific adaptation of a meta-trained model is essential for improving its performance on OOD tasks during meta-testing (Hu et al., 2022; Chen et al., 2023; Li et al., 2022). Although effective, one major limitation of these adaptation methods is the reliance on back-propagation of the gradients, which can be expensive, making these methods inefficient and potentially impractical due to poor scaling with model size. To this end, we propose an adaptation strategy, specifically designed for SMAT, that *bypasses the gradient computation*. At the core of our method lies the hypothesis that each expert selection

score can be descreatzed i.e, either 0 or 1, which aligns with the intuition that each expert knowledge is either required or not for solving each tasks. We can thus optimize the expert selection score in a binary hypothesize space i.e, $\alpha_i \in \{0, 1\}^{|\mathcal{M}|}$. A more detailed description of the algorithm can be found in Appendix A.2.

Gradient-based fine-tuning with SMAT. Following (Hu et al., 2022), we also consider gradient-based fine-tuning of our meta-tuned model at meta-testing time. With only few changes, our method is fully compatible with existing full fine-tuning (i.e., fine-tuning the entire model), and parameter-efficient fine-tuning (PEFT) techniques. Specifically, we use θ_i in Eqn. (2) as the task-specific model initialization, before applying any off-the-shelf fine-tuning technique for further optimizing θ_i on the support set of each task. The pseudocode, as well as ablation results on fine-tuning SMAT, at meta-testing time, using our proposed methods are shown in the Appendix.

5. Experiment

We now verify the efficacy and competitiveness of SMAT on standard meta-learning benchmarks. Additional details and results can be found in Appendix B and C.

Experimental setup. We conduct meta-tuning experiments on Meta-dataset (**MD**) (Triantafillou et al., 2020), which is a widely studied large-scale cross-domain few-shot learning benchmark. As in PMF (Hu et al., 2022), we adhere to the official guidelines and employ the standard meta-training and meta-testing splits for meta-tuning and meta-testing. We select all hyperparameters and the meta-tuned checkpoint for testing using the official meta-validation split. In order to obtain a more comprehensive evaluation of the meta-tuned models, we introduce additional OOD datasets for *meta-testing only*, which were not used during the meta-tuning process on MD.

Baselines. We compare SMAT to two types of baselines: **(a)** the **Pre**-trained model without meta-tuning, and **(b)** meta-tuning methods: **PMF** (Hu et al., 2022), which is the SOTA on MD. To compare against a MoE baseline for meta-tuning, we adopt the recently proposed SMEAR (Muqeeth et al., 2023) which implements soft merging of experts in the parameter space, and denote this baseline by **SoftMerge**. All methods use DINO-ViT-Small (Caron et al., 2021) as the pre-trained backbone.

Evaluation. At meta-testing time, we resort to the ProtoNet (Snell et al., 2017) classifier for performing direct inference on each few-shot testing tasks without further adaptation. When considering task-specific fine-tuning on the support sets, we follow the same protocols in PMF (Hu et al., 2022) for all models, using **Full** fine-tuning (fine-tuning the entire model) and **LoRA** (Hu et al., 2021) as

Table 1. Few-shot testing results on the Meta-dataset benchmark and additional OOD testing datasets for methods using DINO-ViT-Small backbone. [†] and [‡] respectively indicate published results in [†](Hu et al., 2022) and [‡](Basu et al., 2023). Gray indicates our method.

Datasets	w/o fine-tuning				with gradient-based fine-tuning						
	[†]Pre	[†]PMF	SoftMerge	SMAT	[‡]Pre+full	[†]PMF+full	SoftMerge+full	SMAT+full	[‡]Pre+LoRA	PMF+LoRA	SMAT+LoRA
ImageNet	73.48	73.54	74.33	**74.94**	73.54	74.59	74.71	75.24	74.22	73.54	**75.72**
Aircraft	62.17	88.33	88.80	**89.49**	75.4	88.33	90.6	**90.78**	80.8	89.75	90.71
Omniglot	54.33	**91.79**	91.24	89.54	78.7	91.79	92.01	90.83	80.8	**92.78**	90.99
CUB	85.37	91.02	91.54	**92.48**	85.4	91.02	91.95	**92.48**	85.8	91.17	92.42
DTD	83.67	81.64	80.98	**85.86**	86.9	86.61	86.84	**88.34**	86.8	86.73	88.28
Quickdraw	60.59	**79.23**	78.98	78.83	73.6	**79.23**	79.90	78.83	72.7	**79.23**	78.83
Fungi	56.26	**74.2**	72.40	72.8	54.7	74.20	72.40	72.80	59.8	**75.44**	72.80
VGGFlower	94.45	94.12	96.89	**97.19**	94.2	94.12	97.01	97.19	94.8	96.05	**97.25**
ID Avg	71.29	84.23	84.40	**85.14**	77.81	84.99	85.56	85.81	79.47	85.59	**85.88**
TrafficSig	53.7	54.37	56.21	**58.51**	87.3	88.85	89.91	**90.83**	88.1	89.14	90.18
MSCOCO	54.58	57.04	55.75	**57.35**	61.5	62.59	62.15	63.07	62.1	61.71	**63.38**
Cifar10	**85.64**	80.82	84.58	83.95	92.48	89.61	91.84	92.08	93.33	91.53	92.46
Cifar100	76.86	69.11	70.85	74.85	86.13	82.54	85.88	85.91	**86.17**	85.06	85.88
MNIST	78.57	93.33	94.16	**94.53**	92.54	96.44	96.20	**96.73**	94.98	96.41	96.46
Sketch	47.25	41.10	43.30	**48.91**	56.39	49.65	53.85	56.55	**57.34**	47.59	55.63
Pet	91.73	91.37	89.84	**92.31**	92.03	91.73	90.48	**92.31**	92.06	92.01	92.31
Clipart	55.19	53.92	54.83	**59.87**	**67.18**	62.83	65.50	65.76	66.51	60.6	66.07
Food	62.64	61.89	63.04	**65.59**	65.08	62.97	63.36	67.43	65.06	62.71	**67.77**
Cars	34.58	**38.00**	36.21	36.79	40.98	40.07	41.62	**42.39**	39.49	42.37	40.05
OOD Avg	64.07	64.10	64.87	**67.27**	74.16	72.73	74.08	**75.31**	74.51	72.91	75.02

fine-tuning methods. Namely, for each dataset, we perform a hyperparameter search on a small set of validation tasks to obtain the optimal learning rate for task-specific fine-tuning using the Adam optimizer for 50 steps for each testing tasks from that dataset.

5.1. SMAT achieves new SOTA performance

In Tab. 1, SMAT consistently achieves the highest overall few-shot classification accuracy for both ID and OOD meta-testing with and without adaptation on task support sets. Results in Table A3 for supervised pre-trained backbone further demonstrate the superiority of our approach over baselines. More specifically, results in Tab. 1 show that:

SMAT is a better out-of-the-box few-shot learner. SMAT attains the best few-shot learning performance in 5/8 ID and 7/10 OOD datasets without adaptation, outperforming the baseline, PMF, by 0.91% and 3.17% on average in ID and OOD evaluation settings, respectively.

SMAT is a transferable initialization for few-shot fine-tuning. When considering task-specific adaptation through fine-tuning on the support set of each task, SMAT shows great compatibility with off-the-shelf fine-tuning techniques. Specifically, fine-tuning starting from SMAT's θ_i leads to the best performance among all baselines when applying the same fine-tuning technique. SMAT improves on PMF by as much as 0.82% (ID) and 2.48% (OOD) when fully fine-tuning the entire meta-tuned model as initialization.

SMAT achieves superior OOD generalization performance. While PMF exhibits relatively lower OOD perfor-

mance w.r.t. the pre-trained baselines, SMAT, in contrast, achieves improved generalization performance, outperforming the pre-trained by 3.2% and at least 0.5% (when both +LoRA) for without and with adaptation, respectively.

5.2. Roles of sparsity in meta-tuning with SMAT

ID vs. OOD tradeoff through controlled sparsity levels. We observe that adjusting the expert sparsity τ allows us to balance the trade-off between in-domain (ID) and out-of-domain (OOD) performance of our meta-tuned SMAT models. In Fig. 3, we see that the OOD performance generally improves while the ID performance decreases as the expert sparsity level τ increases. We hypothesize that this result is due to the stronger intrinsic meta-regularization effect associated with higher sparsity constraints, as well as the better preservation of the more generic pre-trained features through weight interpolation between the meta-tuned and pre-trained parameters in our formulation in sparse interpolated experts (see Eqn. (2)). Both of these factors help to mitigate meta-overfitting to the ID meta-training tasks hence improve meta-generalization.

Sparsity in experts encourages specialization. Meta-learned sparsity patterns in the experts $z_m \odot \theta^\delta$ induce sparsity in the meta-gradients which alleviates harmful task-interference, thereby encouraging specialization among the experts. In Fig. 4, we compare the average alignment of meta-gradients between tasks during training for two SMAT models with different expert sparsity levels τ, where the alignment of gradients, defined as $\mathbb{E}_{\mathcal{T}_i, \mathcal{T}_j \sim \mathcal{P}_{ID}}[\cos(\nabla_{\boldsymbol{\theta}} \mathcal{L}_i, \nabla_{\boldsymbol{\theta}} \mathcal{L}_j)]$, is respectively computed
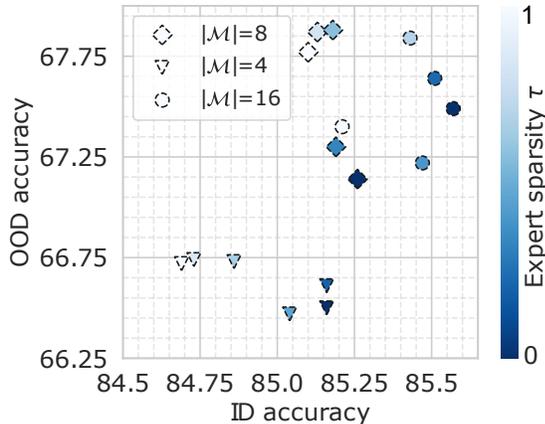
Figure 3. Average performance tradeoff on sampled ID vs OOD tasks as a function of (color) expert sparsity level τ, and (marker) number of experts.

Figure 4. Meta-gradients alignment between tasks throughout for SMAT with low and high sparsity levels. Meta-gradients are calculated w.r.t. the parameters shown in the legend.

for $\boldsymbol{\theta} \in \{\boldsymbol{\theta}^{\delta}, \boldsymbol{z}_m \odot \boldsymbol{\theta}^{\delta}, \forall m \in [|\mathcal{M}|]\}$, i.e., the overall meta-tuned parameter and each expert individually. The results show that the higher sparsity level $\tau = 0.9$ in SMAT can lead to greater alignment of meta-gradients between tasks. Moreover, the alignments in the experts' meta-gradients (which are sparse) are generally higher than that of the overall one, (i.e., w.r.t. $\boldsymbol{\theta}^{\delta}$ in black) - a sign for development of each expert into highly specialized parameters.

5.3. Ablation studies

Table 2. Ablation studies on different components of SMAT. **MLS**: meta-learned sparsity, **Meta**: Meta-training using support and query splits (otherwise no split), **DT**: dense teachers. **IE**: interpolated experts

ID	MODEL	MLS	META	DT	IE	ID	OOD	AVG
1	SMAT	✓	✓	✓	✓	85.14	**67.27**	**75.21**
2		✓	✓	✗	✓	85.07	66.44	74.74
3		✓	✓	✓	✗	84.77	67.02	74.90
4		✓	✗	✓	✓	82.35	63.64	71.95
5		✗	✓	✓	✗	**85.21**	66.21	74.75
6	PMF	✗	✗	✗	✗	84.23	64.09	73.05

Importance of the different components. In Tab. 2, we present several ablated variants of SMAT where we replace or remove certain components. We perform the study by meta-tuning on the DINO ViT-Small backbone and report the meta-testing results without adaptation. Overall, we observe that SMAT performs better than all ablated models on average, demonstrating the effectiveness of each proposed component. We notice that incorporating MoE at the MLP layers of the ViT (index 5), hence predetermining the sparsity patterns, leads to a marginally better ID performance; however, at the cost of a 1% drop in its OOD performance

compared to SMAT. The results indicate the advantages of explicitly meta-learning the sparsity patterns in the experts for generalization.

Number of experts in SMAT. In Fig. 3, we see that having more experts, hence higher model capacity given the same expert sparsity τ, generally increases both ID and OOD performance of our model. The aforementioned ID vs. OOD tradeoff still exists for different numbers of experts; however, the OOD-to-ID tradeoff ratio (defined as $\frac{\Delta_{\text{OOD acc}}}{\Delta_{\text{ID acc}}}$) varies - with $|\mathcal{M}| = 4$ experts having the worst tradeoff ratio, and increasing $|\mathcal{M}|$ from $4 \rightarrow 8$ leads to the most significant gain in the ratio while the improvement seems to saturate when further increasing $|\mathcal{M}|$ from $8 \rightarrow 16$.
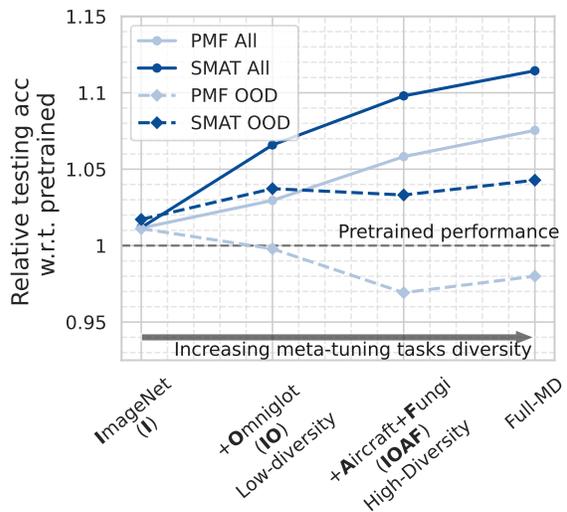
Figure 5. Relative testing performance w.r.t. the pre-trained initialization ($\frac{\text{Avg acc meta-tuned}}{\text{Avg acc pre-trained}}$) for SMAT vs PMF using meta-training tasks of increasing diversity.
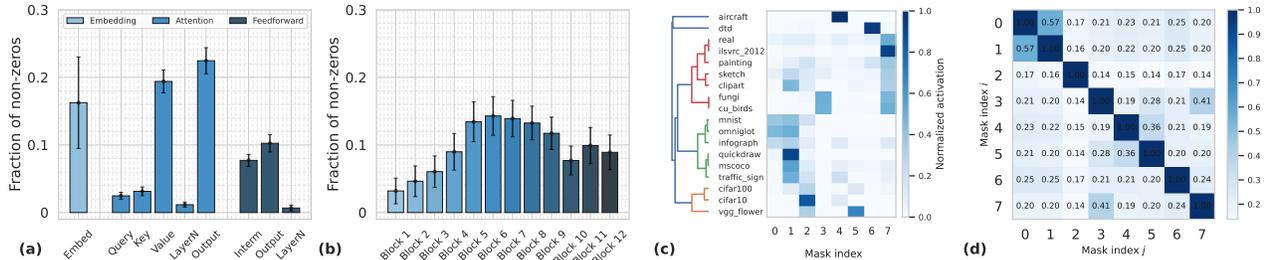
Figure 6. (a-b) Meta-learned model capacity after meta-tuning (i.e., number of non-zero parameters) grouped by **a**): layer types, and **b**): layer depth. (c-d) Expert specialisation. **c)** Dendrogram of task similarity based on expert selection scores. **d)** Overlap between masks.

Scale of the Meta-tuning Datasets. In Fig. 5, we investigate the impact of the quantity and diversity of tasks observed during meta-tuning on the generalization performance of meta-tuned models on unseen meta-testing tasks. As anticipated, the overall generalization performance (on ID and OOD testing tasks), of both PMF and ours, improves as the scale of the meta-tuning datasets increases along the axis. However, even with increased quantity and diversity in the meta-tuning tasks, the OOD performance of PMF is not always better than that of the pre-trained model, which we conjecture is due to both meta-overfitting and harmful task interference. In contrast, SMAT consistently achieves better OOD, and overall, performance compared to both PMF and pre-trained models, with a noticeable ∼4% improvement in relative OOD performance even under the low-diversity (IO) scenario. It is worth noting that this is also where the largest improvement in OOD performance for SMAT occurs. Intuitively, this is because Omniglot is very different from ImageNet - the only training source prior to its addition, resulting in a significant increase in task diversity. With only these two datasets, SMAT achieves comparable OOD performance to its full-MD version. These results highlight the effectiveness of using SMAT for meta-tuning under low data diversity settings, as well as its ability to achieve improved few-shot generalization by better leveraging task diversity during meta-tuning.

5.4. Qualitative visualization

Patterned sparsity emerges through meta-tuning. In Fig. 6, we visualize the sparsity patterns on masks identified through meta-tuning. We observe that the sparsity levels vary significantly depending on the layer types (a), and depths (b). Specifically, the intermediate layers (5-9) have lower per-layer sparsity, while the first few layers are highly sparsified, with sparsity levels as high as 95%. Among the different layer types, we find that three types of layers retain most of their modulation parameters (non-zeros): (1) the first input embedding layers, (2) values of the attention module, and (3) linear layers of attention and feedforward modules. Across different masks, we notice that the standard deviations of sparsity levels are particu-

larly larger for the first embedding layers and throughout layers at all depths. By examining the overlapping ratio (defined as $\frac{|(z_i \cap z_j) \neq 0|}{|(z_i \cup z_j) \neq 0|}$), as shown in Appendix Fig. 6(d), we find that different masks, hence experts, generally have a small overlap. This indicates that SMAT has indeed discovered a diverse set of sparse interpolated experts through meta-tuning.

Learned expert merging rule encodes task relationship. A closer look at the average expert selection scores α by datasets reveals that both specialized experts and a meaningful selection rule have been meta-learned by SMAT, as evident in the Fig. 6(c). We first note that overall, every expert has been utilized by some domains. More interestingly, the dendrogram produced by the similarity of mask selection scores clearly shows hierarchical clustering according to visual similarities between domains. Furthermore, we note sparsity and discreteness of expert selection generally inversely correlates with tasks complexity, with more sparse and discrete selection for intuitively simpler tasks (e.g., Omniglot, Dtd) than the more complex ones (e.g., Ilsvrc_2012).

6. Conclusion

We introduced a simple-yet-effective meta-tuning framework coined SMAT that accommodates to each task through an interpolation of the pre-trained model and a learned combination of sparse experts. Our experiments conclusively demonstrate SMAT's effectiveness in delivering a more generalizable pre-trained model, resulting in state-of-the-art performance on out-of-distribution datasets. Notably, SMAT seamlessly integrates with cutting-edge parameter-efficient fine-tuning methods, and analyses of sparsity patterns underscore the specialization of the learned experts.

Acknowledgement

Impact Statement

The proposed meta-tuning strategy exhibits a broad applicability, poised for extension or adaptation beyond vision pre-trained models. We envision this project as pioneering a new research trajectory for improving few-shot generalization of foundational models across diverse domains, including natural language processing, life sciences, time series, and more.

By augmenting few-shot generalization in pre-trained models, particularly in real-world applications often situated outside of the pre-training distribution, the proposed meta-tuning approach stands poised to substantially impact various downstream tasks like medical imaging analysis, self-driving, wildlife monitoring and etc.

References

Basu, S., Massiceti, D., Hu, S. X., and Feizi, S. Strong Baselines for Parameter Efficient Few-Shot Fine-tuning, April 2023. arXiv:2304.01917 [cs].

Bateni, P., Goyal, R., Masrani, V., Wood, F. D., and Sigal, L. Improved few-shot visual classification. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14481–14490, 2019.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *IEEE International Conference on Computer Vision*, 2021.

Chen, L., Lu, S., and Chen, T. Understanding benign overfitting in gradient-based meta learning. *Advances in Neural Information Processing Systems*, 35:19887–19899, 2022.

Chen, S., Huang, L.-K., Schwarz, J. R., Du, Y., and Wei, Y. Secure out-of-distribution task generalization with energy-based models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Chen, Y., Zhong, R., Zha, S., Karypis, G., and He, H. Meta-learning via language model in-context tuning. *ArXiv*, abs/2110.07814, 2021.

Dai, D., Dong, L., Ma, S., Zheng, B., Sui, Z., Chang, B., and Wei, F. StableMoE: Stable routing strategy for mixture of experts. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7085–7095, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.489.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020.

Fedus, W., Zoph, B., and Shazeer, N. M. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2021.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

Flennerhag, S., Schroecker, Y., Zahavy, T., van Hasselt, H., Silver, D., and Singh, S. Bootstrapped meta-learning. In *International Conference on Learning Representations*, 2022.

Gallego-Posada, J., Ramirez, J., Erraqabi, A., Bengio, Y., and Lacoste-Julien, S. Controlled sparsity via constrained optimization or: How i learned to stop tuning penalties and love constraints. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022.

Gao, T., Fisch, A., and Chen, D. Making pre-trained language models better few-shot learners. In *Annual Meeting of the Association for Computational Linguistics*, 2021.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. In *International Conference on Learning Representations*, 2018.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hong, D., Zhang, B., Li, X., Li, Y., Li, C., Yao, J., Yokoya, N., Li, H., Jia, X., Plaza, A., et al. Spectralgpt: Spectral foundation model. *arXiv preprint arXiv:2311.07113*, 2023.

Hospedales, T. M., Antoniou, A., Micaelli, P., and Storkey, A. J. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:5149–5169, 2020.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of

large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hu, S. X., Li, D., Stühmer, J., Kim, M., and Hospedales, T. M. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Ilharco, G., Wortsman, M., Gadre, S. Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., and Schmidt, L. Patching open-vocabulary models by interpolating weights. *ArXiv*, abs/2208.05592, 2022.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Lee, J., Tack, J., Lee, N., and Shin, J. Meta-learning sparse implicit neural representations. *Advances in Neural Information Processing Systems*, 34:11769–11780, 2021.

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P., and Finn, C. Surgical fine-tuning improves adaptation to distribution shifts. *ArXiv*, abs/2210.11466, 2022.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Lewis, M., Bhosale, S., Dettmers, T., Goyal, N., and Zettlemoyer, L. Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, 2021.

Li, W.-H., Liu, X., and Bilen, H. Cross-domain few-shot learning with task-specific adapters. In *IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-sgd: Learning to learn quickly for few shot learning. *ArXiv*, abs/1707.09835, 2017.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

Liu, J., Yang, C., Lu, Z., Chen, J., Li, Y., Zhang, M., Bai, T., Fang, Y., Sun, L., Yu, P. S., et al. Towards graph foundation models: A survey and beyond. *arXiv preprint arXiv:2310.11829*, 2023a.

Liu, T., Puigcerver, J., and Blondel, M. Sparsity-constrained optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023b.

Liu, Y., Lee, J., Zhu, L., Chen, L., Shi, H., and Yang, Y. A multi-mode modulator for multi-domain few-shot classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8433–8442, 2021. doi: 10.1109/ICCV48922.2021.00834.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*, 2018.

Matena, M. and Raffel, C. Merging models with fisher-weighted averaging. *ArXiv*, abs/2111.09832, 2021.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Min, S., Lewis, M., Zettlemoyer, L., and Hajishirzi, H. Metaicl: Learning to learn in context. *ArXiv*, abs/2110.15943, 2021.

Muqeeth, M., Liu, H., and Raffel, C. Soft merging of experts with adaptive routing. *ArXiv*, abs/2306.03745, 2023.

Mustafa, B., Ruiz, C. R., Puigcerver, J., Jenatton, R., and Houlsby, N. Multimodal contrastive learning with LIMoe: the language-image mixture of experts. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Oh, J., Yoo, H., Kim, C., and Yun, S.-Y. Boil: Towards representation change for few-shot learning. *arXiv preprint arXiv:2008.08882*, 2020.

Panigrahi, A., Saunshi, N., Zhao, H., and Arora, S. Task-specific skill localization in fine-tuned language models. In *International Conference on Machine Learning*, 2023a.

Panigrahi, A., Saunshi, N., Zhao, H., and Arora, S. Task-specific skill localization in fine-tuned language models. *arXiv preprint arXiv:2302.06600*, 2023b.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

Puigcerver, J., Riquelme, C., Mustafa, B., and Houlsby, N. From sparse to soft mixtures of experts. *ArXiv*, abs/2308.00951, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

Requeima, J., Gordon, J., Bronskill, J., Nowozin, S., and Turner, R. E. Fast and flexible multi-task classification using conditional neural adaptive processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Pinto, A. S., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. In *Neural Information Processing Systems*, 2021.

Roller, S., Sukhbaatar, S., Szlam, A., and Weston, J. Hash layers for large sparse models. In *Neural Information Processing Systems*, 2021.

Schwarz, J. R. and Teh, Y. W. Meta-learning sparse compression networks. *arXiv preprint arXiv:2205.08957*, 2022.

Schwarz, J. R., Tack, J., Teh, Y. W., Lee, J., and Shin, J. Modality-agnostic variational compression of implicit neural representations. *arXiv preprint arXiv:2301.09479*, 2023.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Shysheya, A., Bronskill, J., Patacchiola, M., Nowozin, S., and Turner, R. E. Fit: Parameter efficient few-shot transfer learning for personalized and federated image classification. *arXiv preprint arXiv:2206.08671*, 2022.

Snell, J., Swersky, K., and Zemel, R. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Tack, J., Kim, S., Yu, S., Lee, J., Shin, J., and Schwarz, J. R. Learning large-scale neural fields via context pruned meta-learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Thrun, S. and Pratt, L. *Learning to learn*. Springer Science & Business Media, 2012.

Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.-A., and Larochelle, H. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020.

Triantafillou, E., Larochelle, H., Zemel, R., and Dumoulin, V. Learning a universal template for few-shot dataset generalization. In *International Conference on Machine Learning*, pp. 10424–10433. PMLR, 2021a.

Triantafillou, E., Larochelle, H., Zemel, R. S., and Dumoulin, V. Learning a universal template for few-shot dataset generalization. *ArXiv*, abs/2105.07029, 2021b.

Von Oswald, J., Zhao, D., Kobayashi, S., Schug, S., Caccia, M., Zucchet, N., and Sacramento, J. Learning where to learn: Gradient sparsity in meta and continual learning. *Advances in Neural Information Processing Systems*, 2021.

Wang, Z., Tsvetkov, Y., Firat, O., and Cao, Y. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *ArXiv*, abs/2010.05874, 2020.

Wortsman, M., Ilharco, G., Li, M., Kim, J. W., Hajishirzi, H., Farhadi, A., Namkoong, H., and Schmidt, L. Robust fine-tuning of zero-shot models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7949–7961, 2021.

Yao, H., Wang, Y., Wei, Y., Zhao, P., Mahdavi, M., Lian, D., and Finn, C. Meta-learning with an adaptive task scheduler. *Advances in Neural Information Processing Systems*, 34:7497–7509, 2021.

Yeh, C.-C. M., Dai, X., Chen, H., Zheng, Y., Fan, Y., Der, A., Lai, V., Zhuang, Z., Wang, J., Wang, L., et al. Toward a foundation model for time series data. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 4400–4404, 2023.

Yu, L., Shi, B., Pasunuru, R., Muller, B., Golovneva, O., Wang, T., Babu, A., Tang, B., Karrer, B., Sheynin, S., et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. *ArXiv*, abs/2001.06782, 2020.

Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V. Y., Dai, A. M., Chen, Z., Le, Q. V., and Laudon, J. Mixture-of-experts with expert choice routing. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

Zintgraf, L. M., Shiarlis, K., Kurin, V., Hofmann, K., and Whiteson, S. Caml: Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, 2019.

Zuo, S., Liu, X., Jiao, J., Kim, Y. J., Hassan, H., Zhang, R., Gao, J., and Zhao, T. Taming sparsely activated transformer with stochastic experts. In *International Conference on Learning Representations*, 2022.

A. Pseudocode for SMAT

A.1. SMAT for meta-training

The pseudocode for meta-tuning using SMAT can be found in Alg. 1 below. Our implementation is publicly available at github.com/szc12153/sparse_meta_tuning.

Algorithm 1 SMAT: Meta-training

Data: Meta-training tasks $\mathbb{T} := \{\mathcal{T}_1, \mathcal{T}_2, ... \mathcal{T}_N\} \sim \mathcal{P}_{ID}$

Require: Pre-trained initialization θ^{pre}; Target expert sparsity level $\tau_m = \tau$; Number of experts M

Output: Variational distribution $\Phi = \{\phi_m\}_{m=1}^{|\mathcal{M}|}$ for the sparse masks; Hypernetwork h_ψ; Meta-tuned parameters θ^δ

Initialize ζ, Φ, θ^δ

Initialize the Lagrangian multipliers $\lambda_m = 0$ for sparsity constraint $1 - \frac{L_0(z_m)}{\dim(z_m)} \geq \tau$, $\forall m \in [|\mathcal{M}|]$.

while not converged **do**

    $\mathbb{B} \sim \mathbb{T}$ # Sample a batch of tasks

    for $i = 1, 2, ..., |\mathbb{B}|$ **do**

        $\mathcal{T}_i \to \mathcal{T}_i^s, \mathcal{T}_i^q$ # Split into support and query sets

        $\boldsymbol{\alpha}_i' \sim \texttt{GumbleSigmoid}(h_\zeta(\mathcal{T}_i^s))$ # Sample expert merging scores from the hypernetwork

        $\alpha_{i,m} = \frac{\alpha_{i,m}'}{\sum_m^M \alpha_{i,m}'}$ # Normalize the merging weights

        $z_m \sim q_{\phi_m}, \forall m \in [M]$ # Sample sparse masks from the variational distribution

        $\theta_i = \theta^{\text{pre}} + \theta^\delta \odot \sum_m^M \alpha_{i,m} z_m$ # Weighted-sum of sparse experts

        $\theta_i^{\text{tr}} \leftarrow \texttt{StopGrads}(\theta_i)$

        for $k = 1, 2, ..., K$ **do**

            # Task-specific dense teacher

            $\theta_i^{\text{tr}} \leftarrow \texttt{GradientDescent}(\nabla_{\theta_i^{\text{tr}}} \mathcal{L}_{\mathcal{T}_i^q}^{\text{ce}}(\theta_i^{\text{tr}}))$

        end for

        $\mathcal{L}_i := \beta \mathcal{L}_{\mathcal{T}_i^q}^{\text{ce}}(\theta_i) + (1 - \beta) \mathcal{L}_{\mathcal{T}_i^q}^{\text{kd}}(\theta_i, \theta_i^{\text{tr}})$ # task's meta-loss, β is a weighting coefficient $\in (0, 1)$

    end for

    $\mathcal{C}o_m := \frac{\lambda}{\dim(\phi_m)} \sum_j^{\dim(\phi_m)} Q_{\phi_m}(s_j \leq 0) - \tau, \forall m \in [M]$ # sparsity loss for each expert

    $[\zeta, \Phi, \theta^\delta] \leftarrow \texttt{GradientDescent}(\nabla_{[\zeta, \Phi, \theta^\delta]}(\frac{1}{|\mathbb{B}|} \sum_{i=1}^{|\mathbb{B}|} \mathcal{L}_i + \sum_{m=1}^M \lambda_m \mathcal{C}o_m))$ # Minimization problem in Eqn. (2)

    for $m = 1, 2, ..., M$ **do**

        if $\mathcal{C}o_m < 0$ **then**

            # expected expert sparsity is lower than the constraint τ

            $\lambda_m \leftarrow \texttt{GradientAscent}(\nabla_{\lambda_m} \mathcal{C}o_m \frac{\sum_{i \in \mathbb{B}} \alpha_{i,m}}{|\mathbb{B}|})$ # Maximization problem in Eqn. (2)

        else

            # expected expert sparsity is at least τ, the constraint is satisfied

            $\lambda_m \leftarrow 0$ # Reset the Lagrangian multiplier for the m-th constraint

        end if

    end for

end while

Return $\Phi, \zeta, \theta_\delta$

A.2. SMAT for meta-testing

We propose a heuristic in Alg. 2 for optimizing the task-specific expert selection during meta-testing time without the need for expensive gradient computation. Specifically, we restrict the normalized expert selection score estimated by our hypernetwork h_ζ in binary states i.e., $\alpha_{i,m}' \in \{0, 1\}$, $\forall m \in [M]$, and optimize $\alpha_{i,m}$ in this binary space by minimizing the loss on the meta-testing task's support set. The intuition behind this is that each expert is either needed or discarded for each few-shot learning tasks, which is supported by our empirical observations on $\alpha_{i,m}'$ being very discrete (close to 0 or 1) in most cases.

Algorithm 2 SMAT: **Meta-testing time gradient-free expert selection** (for a single task $\tilde{\mathcal{T}}_i$)

Input: Testing support set $\tilde{\mathcal{T}}_i^s$ and query inputs $\tilde{\mathbf{X}}_i^q$, Meta-trained $\zeta, \theta^\delta, \Phi$, Pre-trained $\theta^{\texttt{pre}}$
$\boldsymbol{\alpha}_i' \sim \texttt{HardGumbleSigmoid}(h_\zeta(\tilde{\mathcal{T}}_i^s))$ # Initialize expert merging scores using the hypernetwork and round to [0,1]
$l* = $ positive infinity   # use to record the lowest support loss during exploration
$\boldsymbol{z}_m \sim q_{\phi_m}, \ \forall m \in [M]$ # Sample sparse masks once at the start
for $r = 1, 2, ..., R$ **do**
     # repeat for R rounds of sampling
   for $m = 1, 2, ..., M$ **do**
      # iterate through each score in α_i
     Flip $\alpha_{i,m}'$, $0 \leftrightarrow 1$ # Generate candidate score for the m-th expert
     $\alpha_{i,m} = \frac{\alpha_{i,m}'}{\sum_m^M \alpha_{i,m}'}$ # Normalize the merging weights
     $\boldsymbol{\theta}_i = \boldsymbol{\theta}^{\texttt{pre}} + \boldsymbol{\theta}^\delta \odot \sum_m^M \alpha_{i,m} \boldsymbol{z}_m$ # Weighted-sum of sparse experts
     $\tilde{\mathcal{L}}_i := \mathcal{L}_{\tilde{\mathcal{T}}_i^s}^{\texttt{ce}}(\boldsymbol{\theta}_i)$ # Evaluate the support loss which only requires forward passes
     if $\tilde{\mathcal{L}}_i < l$ **then**
       # Rejection sampling
       Accept the candidate $\alpha_{i,m}$ with ρ and record $l* = l = \tilde{\mathcal{L}}_i$, otherwise reject
     else
       Accept the candidate $\alpha_{i,m}$ with $(1 - \rho)$ and record $l = \tilde{\mathcal{L}}_i$ otherwise reject
     end if
   end for
end for
Return: $\boldsymbol{\alpha_i}$ at the lowest support loss l^*, Which is then used for final prediction on the query $\hat{\mathbf{Y}}_i^q = f(\tilde{\mathbf{X}}_i^q; \boldsymbol{\theta}_i = \boldsymbol{\theta}^{\texttt{pre}} + \boldsymbol{\theta}^\delta \odot \sum_m^M \alpha_{i,m} \boldsymbol{z}_m$.

Algorithm 3 SMAT: **Meta-testing time full fine-tuning** using θ_i as an model initialization (for a single task $\tilde{\mathcal{T}}_i$)

Input: Testing support set $\tilde{\mathcal{T}}_i^s$ and query inputs $\tilde{\mathbf{X}}_i^q$; Meta-trained $\zeta, \theta^\delta, \Phi$, Pre-trained $\theta^{\texttt{pre}}$
$\boldsymbol{\alpha}_i' \sim \texttt{GumbleSigmoid}(h_\zeta(\tilde{\mathcal{T}}_i^s))$ # Obtain expert merging scores using the hypernetwork
$\alpha_{i,m} = \frac{\alpha_{i,m}'}{\sum_m^M \alpha_{i,m}'}$ # Normalize the merging weights
$\boldsymbol{\theta}_i = \boldsymbol{\theta}^{\texttt{pre}} + \boldsymbol{\theta}^\delta \odot \sum_m^M \alpha_{i,m} \boldsymbol{z}_m$ # Weighted-sum of sparse experts
$\boldsymbol{\theta}_{i,0} \leftarrow \texttt{StopGrads}(\boldsymbol{\theta}_i)$
for $k = 0, 2, ...(K - 1)$ **do**
   $\boldsymbol{\theta}_{i,k+1} \leftarrow \texttt{GradientDescent}(\nabla_{\boldsymbol{\theta}_{i,k}} \mathcal{L}_{\tilde{\mathcal{T}}_i^s}^{\texttt{ce}} \boldsymbol{\theta}_{i,k})$ # finetune θ_i on the support set for K steps
end for
Return: Final prediction on the query set $\hat{\mathbf{Y}}_i^q = f(\tilde{\mathbf{X}}_i^q; \boldsymbol{\theta}_{i,K})$

B. Experiment details

B.1. Implementation of baselines

PMF (Hu et al., 2022). In Tab. 1, we report published results by PMF in their paper (Hu et al., 2022) whenever they are available. For extra meta-testing datasets in Tab. 1, which were not included in their paper, we produce these results using the official meta-trained PMF model checkpoint, which is publicly available with their code on Github.

B.2. Details for meta-testing

W/O fine-tuning. To perform direct few-shot inference for a given testing task, $\tilde{\mathcal{T}}_i := \{\tilde{\mathbf{X}}_i^s, \tilde{\mathbf{Y}}_i^s, \tilde{\mathbf{X}}_i^q\}$, we consider using the prototypical network (Snell et al., 2017), which first constructs class centroids in the feature space of the model using the labeled support set, before performing the nearest centroid classification on the query input.

Denote the feature backbone (e.g., a pre-trained or meta-tuned ViT) by $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^D$ parameterized by $\boldsymbol{\theta}$, which

essentially is a mapping from the input pixel space to a vector space of dimension D. The centroid for each unique class $\mathbf{c}_k, k = [1, 2, 3...K]$ in $\tilde{\mathcal{T}}_i$ is calculated from the support set by:

$$\mathbf{c}_k = \frac{1}{|j : y^s_{i,j} = c_k|} \sum_{j:y^s_{i,j}=c_k} f_{\boldsymbol{\theta}}(\mathbf{x}^s_{i,j}).$$

To this end, the predicted label for each query input is given by

$$p(y = k|\mathbf{x}^q) = \frac{\exp(-d(f_{\boldsymbol{\theta}}(\mathbf{x}^q), \mathbf{c}_k))}{\sum_{k'=[1,2..]}^{K} \exp(-d(f_{\boldsymbol{\theta}}(\mathbf{x}^q), \mathbf{c}'_k))},$$

where $d(\cdot, \cdot)$ is some distance metric e.g., cosine distance.

C. Additional experimental results

C.1. Meta-dataset meta-testing results for meta-tuned supervised pre-trained backbones

Table 3 provides the results of the meta-tuned models on MD using PMF (Hu et al., 2022) and SMAT with Sup21K-ViT-Small backbone.

SMAT attains the best performance. Although using a different pre-trained backbone, we again observe that SMAT outperforms all baselines in both ID and OOD meta-testing. These results, together with our main results in Tab. 1, validate the efficacy of our approach.

Comparing to few-shot meta-testing results for meta-tuned DINO backbone. Meta-tuning on self-supervised and supervised pre-trained backbones produces vastly different generalization results on OOD tasks. While for both PMF and ours, meta-tuning with the Sup21K backbone generally improves few-shot testing performance over the pre-trained backbone even on unseen tasks and domains, meta-tuning with a self-supervised pre-trained backbone (e.g., DINO) requires taking more care in the design of meta-tuning strategy - noticeably, naively meta-tuning with PMF can be outperformed by simple pre-trained + fine-tuned baselines, particularly on OOD tasks; in contrast, SMAT, still maintains the high transferability in its meta-tuned feature representation which leads to better, if not better, fine-tuning performance when using ours meta-tuned model as the fine-tune initialization.

Table 3. Few-shot testing results on the Meta-dataset benchmark and additional OOD testing datasets for methods using Sup21k-ViT-Small backbone (Dosovitskiy et al., 2020). SMAT indicates our method.

	Gradient-free			Gradient-based					
Meta-dataset	Pre	PMF	Ours	Pre+full	PM+full	SMAT+full	Pre+LoRA	PMF+LoRA	SMAT+LoRA
ImageNet	68.45	79.57	**81.39**	78.96	80.37	**82.10**	76.30	80.21	81.89
Aircraft	52.57	85.55	**87.05**	83.20	88.48	**89.16**	80.73	87.40	88.40
Omniglot	37.02	**86.63**	86.14	78.25	**88.58**	88.56	73.75	88.05	87.91
CUB	84.65	94.72	**94.98**	91.20	94.72	**94.98**	90.18	94.72	**94.98**
DTD	80.75	84.84	**86.27**	87.20	88.60	**89.05**	86.59	87.85	88.50
Quickdraw	55.05	78.55	**79.35**	75.75	**80.01**	79.36	74.07	78.55	79.35
Fungi	44.20	73.02	**74.50**	56.29	73.02	**74.50**	55.88	73.02	74.50
VGGFlower	94.11	98.97	**99.09**	98.01	99.09	**99.19**	96.97	99.06	99.14
Avg ID	64.60	85.23	**86.10**	81.11	86.61	**87.11**	79.31	86.11	**86.83**
TrafficSig	48.14	55.80	**60.10**	90.02	**90.13**	**90.13**	89.04	88.89	89.52
MSCOCO	52.39	63.77	**63.91**	64.64	67.02	**68.19**	64.08	67.12	67.65
Cifar10	79.33	87.50	**91.10**	93.40	93.78	**94.87**	92.80	92.31	93.62
Cifar100	68.53	79.11	**82.02**	88.54	88.81	**90.32**	87.69	88.81	88.80
MNIST	73.53	93.90	**94.01**	95.16	96.70	**96.76**	94.82	96.51	96.67
Avg OOD	64.38	76.02	**78.23**	86.35	87.29	**88.05**	85.69	86.76	**87.25**

C.2. Different fine-tuning strategies for SMAT at meta-testing

Performance. In Tab 4 below, we evaluate various meta-testing fine-tuning strategies for SMAT and compare their performance. We first note the effectiveness of our proposed gradient-free expert selection method (see Section 4.2), as evidenced by its improved performance compared to directly using SMAT with ProtoNet (Snell et al., 2017) for meta-testing. Second, using θ_i as the initialization for full fine-tuning, which has the same capacity as the pre-trained model θ^{pre}, leads to improved performance over fine-tuning the full SMAT model jointly (i.e., **(4)**), as it is sufficiently expressive while much more parameter-efficient than the latter hence avoids potential over-fitting issues. We provide more explanation as follows.

Gradient-based fine-tuning outperforms gradient-free fine-tuning primarily because it is much more flexible. The extra flexibility of gradient-based fine-tuning (Alg. 3, Appendix) stems from the fact that the entire model θ_i is allowed to be updated, whereas gradient-free fine-tuning (Alg. 2, Appendix) only allows updates on the expert selection weights α_i - with dimensions equal to the number of experts in SMAT; while all other parameters remain frozen. That said, the hypothesis space of our gradient-free fine-tuning algorithm is much more tightly constrained around the model meta-tuned on ID tasks. As a result, the effectiveness of gradient-free fine-tuning is limited on certain OOD tasks that exhibit noticeable distribution shift, such as TrafficSign. In such cases, the meta-tuned ID model may become inadequate, requiring significant parameter updates.

Table 4. Different fine-tuning strategies for SMAT on a subset of few-shot testing tasks. **(1)**: Direct inference by ProtoNet, reported in Tab. 1 as SMAT; **(2)**: Gradient-free expert selection (Alg. 2); **(3)** Full fine-tuning using θ_i as initialization (Alg. 3), reported in Tab. 1 as SMAT+Full. **(4)**: Full fine-tuning the entire SMAT model, i.e., $\zeta, \theta^\delta, \Phi$ jointly.

	GRADIENT-FREE		GRADIENT-BASED	
DATASETS	(1)	(2)	(3)	(4)
TRAFFICSIGN	58.51	59.59	90.83	89.93
MSCOCO	57.35	58.78	63.07	62.76
CIFAR10	83.95	86.95	92.08	91.97
CIFAR100	74.85	77.20	85.91	85.95
MNIST	94.53	94.63	96.73	96.70

Computational cost. We present results below in Tab. 5 for a quantitative comparison of computational cost, in terms of time and GPU memory, between w/o fine-tuning, gradient-free fine-tuning, and gradient-based full fine-tuning for SMAT. All fine-tuning are carried out in FP16 mixed precision. In particular, our gradient-free fine-tuning method offers a +1.59% improvement on average over w/o fine-tuning at no additional memory cost, while saves $3/4$ of the total memory cost of gradient-based fine-tuning. Gradient-based fine-tuning, however, outperforms both by at least +10.31% despite requiring 4x the GPU memory of both, and 2x the time cost compared to gradient-free fine-tuning.

Table 5. Computational cost for different fine-tuning techniques with SMAT.

Method	Time (sec./task)	GPU memory (MiB)	Avg. Acc (Tab. 4)
w/o fine-tuning (ProtoNet)	0.2	4332	73.83
Alg.(2). gradient-free fine-tuning (50steps)	6.4	4332	75.43
Alg.(3). gradient-based full fine-tuning (50steps)	11.8	17264	85.74

C.3. Performance vs number of parameters.

Details on parameter counts: SMAT: We use a very naive compression scheme to remove exact zeros in our model. We use a (`value`, `position`) tuple to represent each non-zero parameter in our model after flattening all parameters in a single long vector. Thus, the total number of parameters left in the experts is equal to two-times the number of non-zero parameters remained at the end of meta-tuning. We point out that there are perhaps more memory efficient ways for representing sparse weights e.g., PyTorch sparse tensors, which could potentially result in a more significant saving in terms of number of binary *bits*.
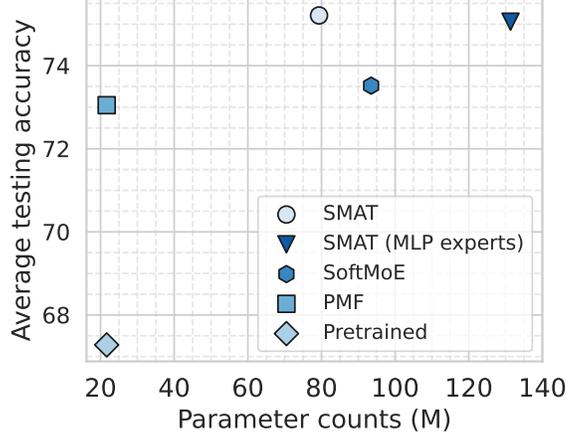
Figure 7. Average testing performance of models vs model parameter counts. We observe SMAT attains the best overall performance while requiring less number of parameters comparing to other MoE implementation variants including SoftMoE (Puigcerver et al., 2023), and incorporating experts in the MLP layers in ViT (SMAT MLP experts).

D. Sparse interpolated experts

As previously stated, *by rearranging Eqn. (1) slightly, the experts now essentially become different sparse interpolations between the same pre-trained and meta-tuned models*. Here are the details.

Starting from Eqn. (1), we have:

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}^{\text{pre}} + \sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m} \boldsymbol{\theta}_m^{\delta} \tag{3}$$

$$= \boldsymbol{\theta}^{\text{pre}} + \sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m} (\boldsymbol{z}_m \odot \boldsymbol{\theta}^{\delta}) \tag{4}$$

$$= \boldsymbol{\theta}^{\text{pre}} + \sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m} (\boldsymbol{z}_m \odot \boldsymbol{\theta}^{\delta+\text{pre}}) - \sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m} (\boldsymbol{z}_m \odot \boldsymbol{\theta}^{\text{pre}}) \tag{5}$$

$$= \sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m} (\mathbf{1} \odot \boldsymbol{\theta}^{\text{pre}}) + \sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m} (\boldsymbol{z}_m \odot \boldsymbol{\theta}^{\delta+\text{pre}}) - \sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m} (\boldsymbol{z}_m \odot \boldsymbol{\theta}^{\text{pre}}) \tag{6}$$

$$= \sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m} ((\mathbf{1} - \boldsymbol{z}_m) \odot \boldsymbol{\theta}^{\text{pre}} + \boldsymbol{z}_m \odot \boldsymbol{\theta}^{\delta+\text{pre}}). \tag{7}$$

We have assumed that $\sum_{m=1}^{|\mathcal{M}|} \alpha_{i,m} = 1$ which we have ensured through normalizing the expert activation in Alg. 1. The result shows that each task model $\boldsymbol{\theta}_i$ can now be interpreted as a weighted sum of different experts, where each expert (i.e.,$(\mathbf{1} - \boldsymbol{z}_m) \odot \boldsymbol{\theta}^{\text{pre}} + \boldsymbol{z}_m \odot \boldsymbol{\theta}^{\delta+\text{pre}}$)) is a sparse interpolation between pre-trained θ^{pre} and meta-tuned $\theta^{\text{pre}+\delta}$ models in the parameter space.