Episodic-free Task Selection for Few-shot Learning

Tao Zhang

Chengdu Techman Software Co., Ltd.

zhangtao@alu.uestc.edu.cn

Abstract

Episodic training is a mainstream training strategy for few-shot learning. In few-shot scenarios, however, this strategy is often inferior to some non-episodic training strategy, e. g., Neighbourhood Component Analysis (NCA), which challenges the principle that training conditions must match testing conditions. Thus, a question is naturally asked: How to search for episodic-free tasks for better few-shot learning? In this work, we propose a novel meta-training framework beyond episodic training. In this framework, episodic tasks are not used directly for training, but for evaluating the effectiveness of some selected episodic-free tasks from a task set that are performed for training the meta learners. The selection criterion is designed with the affinity, which measures the degree to which loss decreases when executing the target tasks after training with the selected tasks. In experiments, the training task set contains some promising types, e. g., contrastive learning and classification, and the target few-shot tasks are achieved with the nearest centroid classifiers on the miniImageNet, tiered-ImageNet and CIFAR-FS datasets. The experimental results demonstrate the effectiveness of our approach.

1. Introduction

The concept of few-shot learning (FSL) emerges as a solution to address the challenge of categorizing a limited number of samples that belong to previously unseen classes. [12, 43]. FSL scenarios frequently arise in domains where sample acquisition proves challenging, such as drug discovery [3], agriculture [44] and healthcare [26], thereby receiving increasing attention. A widely used framework for FSL is meta-learning, which follows the principle that "test and train conditions should match" [42] or "the process of improving a learning algorithm over multiple learning episodes"[17], aiming to learn transferable knowledge for inference.

Meta-learning is endeavoring to address the FSL problem through episodic training, where a batch of samples is divided into a support set and a query set to mimic
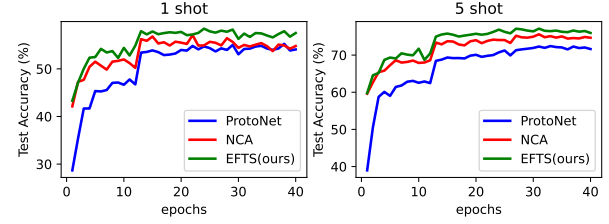

Figure 1. Test accuracies obtained by using Prototypical Network (ProtoNet), Neighbourhood Component Analysis (NCA) and our proposed Episodic-free Task Selection (EFTS) for 5-way 1-shot (left) and 5-way 5-shot (right) tasks on *mini*ImageNet, respectively. In target tasks, ProtoNet is used as the classifiers.

the test conditions [42]. Recently, numerous FSL methods have been proposed based on the episodic training [2, 14, 25, 30, 37–39, 42, 45], in which base-learners are trained in an inner loop, and meta-learners are trained in an outer loop. Furthermore, some work shows that the meta-learning models can still maintain good performance even after omitting the inner loop, e. g., NIL [40].

Here raises a question: *is the most matching task always the best training task?* Some recent experimental results shows that episodic training is often inferior to non-episodic training in few-shot learning, especially in the cross-domain scenarios [7, 8, 24, 33, 40]. To show this, we made an illustration in Fig. 1. Figure 1 shows that the non-episodic training (NCA) outperforms the episodic training (ProtoNet). The reason can be analyzed from different perspectives. Firstly, episodic training discards a large amount of available sample information that is helpful for model training [24]. Secondly, the learned embedding representation in meta-learning is still not good enough [11, 33, 40].

In this work, we address the problem of how to design a framework for selecting the better training tasks for few shot learning. Unlike episodic training, the episodic tasks (episodes in [24]) that are matching to the target tasks are not applied directly for training in our framework, but as a proxy for evaluating the benefits of the episodic-free tasks (episodes or not episodes) to the target tasks. After this evaluation, the optimal tasks are selected as the task for training
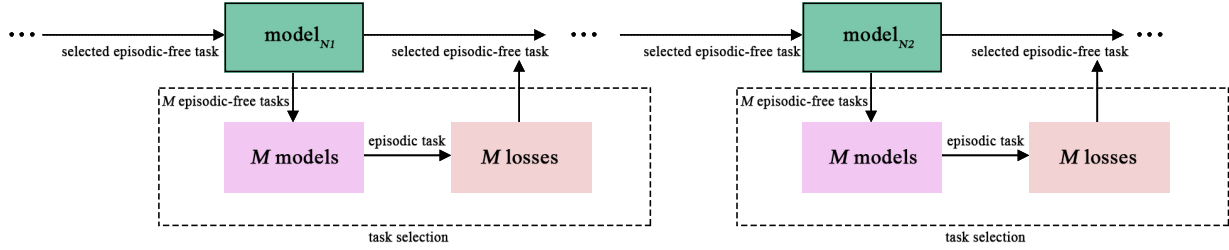
Figure 2. The framework of the proposed EFTS. In this task selection section, firstly, using M episodic-free tasks updates the model respectively; Secondly, M updated models are evaluated using the episodic task, and thus M losses are obtained; finally, M losses are compared, and the episodic-free task corresponds to the smallest loss will be selected for the next stage of training. Task selection can be done multiple times during the training process.

the meta-learners (Fig. 2).

To the best of our knowledge, the mutual influence between tasks that have been studied widely in multi-task learning [13, 36, 41] are rarely applied for few-shot learning. In this paper, a simple and effective approach for inter-task evaluation in multi-task learning, the affinity [13], will be used in our work as a task selection criterion, which is called as episodic-free task selection (EFTS). In EFTS, the tasks that have the highest affinity scores are selected to jointly train the meta-learners, where any ingenious multi-task learning strategies are not adopted. Figure 1 shows that, compared to the other two single-task fixed strategies, EFTS as a multi-task selection strategy can perform the best.

The contributions of this paper are summarized:

1. A novel framework for few-shot learning is proposed. In this framework, the training tasks can be episodic-free, and their effectiveness can be evaluated by a series of episodic tasks.

2. A criterion called affinity that is popular in multi-task learning is introduced into few-shot learning for task selection, by which the effective single tasks that have the highest affinity scores can be selected for jointly training the meta-learners.

3. A series of experiments focusing on the effectiveness of the proposed framework have been conducted, and the influence of hyper-parameters have been discussed.

2. Related Work

In this section, we review the research in multi-task learning, few-shot learning and the cross-interactions between them. The matching principle in meta-learning provides a relationship between source and target tasks, which requires that the source task should be episodic. In an episodic task, a batch of samples is divided into support and query, aiming to mimic the test conditions [42]. The episodic training as a standard method is applied in a lot of few-shot approaches [14, 30, 37, 38, 42]. Oriol Vinyals et al. proposed Matching Nets for few-shot learning , whose training procedure obeys

that test conditions must match train conditions [42]. Based on Matching Nets, Jake Snell et al. proposed prototypical networks (ProtoNet) for few-shot learning [38]. In this approach, a class is represented as a prototype, which is calculated as an average of embeddings of support samples belonging to this class. Subsequently, various improvements are constantly emerging for ProtoNet [2, 15, 28, 30, 31]. Flood Sung. et al. proposed the Relation Network to learn a deep distance metric for comparing the samples in the episodes [39]. Chelsea Finn et al. proposed the Model-Agnostic Meta-Learning (MAML), which trains the base learner in an inner loop for achieve rapid adaptation, and trains the meta learner in an outer loop [14].

These elaborate episodic training based few-shot approaches are questioned recently because in a cross-domain scenario, their performances are inferior to the simple baselines that are often used for transfer learning [8]. In this work by Steinar Laenen et al. [24], a series of systematic experiments were conducted to compare ProtoNet and Neighbourhood Component Analysis (NCA), and it was found that NCA outperformed ProtoNet because NCA was able to fully utilize the information about the similarity between samples, while this information is largely discarded in ProtoNet. In addition, Yinbo Chen et al. designed the Meta-Baseline that combines meta-learning and whole classification for few-shot learning, by which they found meta-learning model that performs better in generalization on unseen tasks from base classes may exhibit poorer performance on tasks from novel classes [9]. The essence of this problem may be related to representations. Yonglong Tian et al. proposed a baseline showing that learning a good representation is important in few-shot learning, even with just a simple linear classifier [40]. These work provides the straightforward presentation of the key problems in episodic training and a series of effective analysis, which may inspire a novel few-shot framework to improve episodic training.

Multi-task learning leverages the transfer of information between tasks to provide inductive biases for the target

task. In multi-task learning, the weighting of task loss will greatly affect the performance of the model. In [20], the weighting problem is solved by a principled approach using homoscedastic uncertainty. In addition, there are interdependencies between tasks, and thus multi-task learning problems can be considered as multi-objective optimization problems and require a Pareto optimal solution [36]. In [13], an intuitive evaluation method called inter-task affinity was proposed, which evaluates how well the model performs in a task when completing another task. This method is used for task grouping training. Recently, the multi-task meta learning (MTML) that combines MTL and bi-level meta optimization was proposed, which introduces the multi-task learning episodes architecture into MTL scenarios [41]. MTML can result in an ensemble that can achieve better performance for an unseen task in fewer steps compared to training the task individually from scratch.

Multi-task learning approaches may provide a good representation for few-shot learning [6, 11]. [11] proposed a theoretical analysis of learning a common good representation between source and target task with multi-learning, aiming to reveal the maximum extent of sample size reduction. [6] explores the framework of multi-task representation (MTR) learning, which aims to leverage source tasks to acquire a representation that reduces the sample complexity of solving a target task. The aforementioned work points out that a representation is crucial for connecting the source and target tasks, but not provides an approach to directly evaluate the benefits of a source task to a target few-shot task.

3. Problem Definition

Let us consider a set of n meta-training tasks $\mathcal{T}_t = \{\tau_1, \tau_2, ..., \tau_T\}$ that can be any types, not limited to just episodes or whole-classification. The ith meta-training task τ_i denotes the ith task set that contains a sample set $\mathcal{X}$. In addition, consider that a meta-training task is optimized using a set of the N-way K-shot meta-evaluation tasks $\mathcal{T}_e = \{\nu_1, \nu_2, ..., \nu_L\}$ that maintain consistency with the conditions of the target few-shot tasks. The lth N-way K-shot meta-evaluation task ν_l with arbitrary l contains the support set $\mathcal{S}_l$ and the query set $\mathcal{Q}_l$. The aim is achieving the target few-shot tasks using the meta-learners that are trained by $\mathcal{T}_t$.

4. Methodology

4.1. Preliminary

Episodic Training. A few-shot learning (FSL) model is commonly trained by completing a series of episodic tasks on a few-data scenario that is similar to those in test stage [42]. In each task, a few data is randomly collected for training meta-learners and base-learners in this model, which

contains support set $\mathcal{S} = \{(\boldsymbol{s}_1, y_1), (\boldsymbol{s}_2, y_2), ..., (\boldsymbol{s}_N, y_N)\}$ and query set $\mathcal{Q} = \{(\boldsymbol{q}_1, y_1), (\boldsymbol{q}_2, y_2), ..., (\boldsymbol{q}_M, y_M)\}$. In the training stage, the base-learners are commonly optimized with $\mathcal{S}$ in the inner loop, aiming to rapidly adapt to a new task, and the meta-learners are iteratively optimized in the outer loop with $\mathcal{Q}$ in classification, or $\mathcal{S}$ and $\mathcal{Q}$ in metric learning. The meta-learner training thus is the main considerations in few-shot learning, and the base-learner training in the inner loop in some cases, e. g., the metric-based meta-learning, can be omitted.

Whole-classification Training. Meta-learner training by completing the episodic tasks is only considering the connections between $\mathcal{S}$ and $\mathcal{Q}$, and massive internal connections in them are discarded. These inadequate utilization of information would degrades the performance of the FSL model. Thus, the whole-classification training strategy is proposed for addressing this issue [24]. In this strategy, a batch is not divided into support and query set, but as a whole for training the meta-learners. In metric-based meta learning, this strategy can consider all the connections in the batch, which may benefit for the model training.

Inter-Task Affinity. Hard parameter sharing is a way for multi-task learning that involves sharing the hidden layers among all tasks while maintaining separate task-specific output layers. During the training process, information is transmitted between tasks through the gradients of parameters in these shared hidden layers. The degree of impact between tasks can measured by inter task affinity [13]. Inter task affinity provides an effective way to cluster tasks for model training.

4.2. Episodic-free Task Selection (EFTS)

In general, the episodic-free tasks can be homogeneous or heterogeneous. Motivated by the work [13] that groups the tasks via inter-task affinity that measures the impact of a task's gradient updates on the objective of other tasks, we can select meta-training tasks by evaluating their gradient updates impacting the N-way K-shot meta-evaluation tasks. Suppose that a meta-training task is embodied with a subset $\mathcal{T}_{sub}$ of $\mathcal{T}_{set} = \{t_1, t_2, ..., t_S\}$ where $\mathcal{T}_{set}$ is the set of given tasks and t_i denotes a single task. Consider a batch of samples $\mathcal{X}$ inputs a network f_θ with parameters $\theta = \{\theta_s, \theta_i\}$ with θ_s the shared parameters that is shared by all the tasks, and θ_i the specific parameters that is proprietary in different tasks. The loss function with input $\mathcal{X}_t$ for the task set $\mathcal{T}_{sub}$ in a tth update by gradient descent can be expressed as

$$\theta_s^{t+1} = \theta_s^t - \lambda \nabla_{\theta_s} \mathcal{L}_{\mathcal{T}_{sub}}(\mathcal{X}^t; \theta_s, \theta_i^t), \qquad (1)$$

where λ is the learning rate that is a hyper-parameter for training. It should be noted that $\mathcal{L}_{\mathcal{T}_{sub}}$ in Eqs. 1 can be implemented with the approaches of multi-task learning. Enlightened by the work [13], we can introduce the affinity

from $\mathcal{T}_{sub}$ to the target tasks $\mathcal{T}_e$:

$$\mathcal{Z}^{t,\text{UNA}}_{\mathcal{T}_{sub}\to\mathcal{T}_e} = 1 - \frac{\sum_{l=1}^{L}\mathcal{L}_{\nu_l}(\mathcal{S}^l,\mathcal{Q}^l;\theta_s^{t+\text{UNA}},\theta_i)}{\sum_{l=1}^{L}\mathcal{L}_{\nu_l}(\mathcal{S}^l,\mathcal{Q}^l;\theta_s^t,\theta_i)}, \quad (2)$$

where UNA is the update number per affinity. In Eq. 2, as $\mathcal{Z}_{\mathcal{T}_{sub}\to\mathcal{T}_e}$ is larger, the selected task set in tth has more positive impacts on the target tasks $\mathcal{T}_e$ in the tth update. Here we expect the model to acquire a transferable knowledge, and thus the $\mathcal{S}^l$ and $\mathcal{Q}^l$ in Eq. 2 are different with $\mathcal{X}_t$ as possible. This implies that $\mathcal{T}_{sub}$ may not have a high affinity with $\mathcal{T}_e$, even they are implemented with the same classifiers (e. g., ProtoNet). In addition, [13] shows that the affinity is calculated across multiple steps is better than single step. Thus, here we propose the averaged affinity across all UNA multiple steps, i.e,

$$\hat{\mathcal{Z}}^{t,\text{UNA,M}}_{\mathcal{T}_{sub}\to\mathcal{T}_e} = \frac{1}{M}\sum_{i=1}^{M}\mathcal{Z}^{t+(i-1)\times\text{UNA},\text{UNA}}_{\mathcal{T}_{sub}\to\mathcal{T}_e}. \quad (3)$$

Then, we can use the average affinity in Eq. 3 as a criterion to select the optimal $\mathcal{T}_{sub}^*$ in an update, i. e.,

$$\mathcal{T}_{sub}^* = arg\min_{\mathcal{T}_{sub}\subset\mathcal{T}_{set}}\hat{\mathcal{Z}}^{t,\text{UNA,M}}_{\mathcal{T}_{sub}\to\mathcal{T}_e}. \quad (4)$$

Equation 4 requires solving a 0-1 integer programming problem that would be NP-hard. When the total of optional meta-training tasks is small, we can use the exponential search strategy such as exhaustive search and branch and bound to select the optimal task subset. But as it is large, it is computationally expensive. In this case, it can be solved by some other strategies, e. g., sequential algorithms and randomized algorithms, which are often used as the wrapper approaches for feature selection [1, 4]. Here we simply calculate the loss of a sub-optimal $\mathcal{T}_{sub}^*$, or called the EFTS objective function:

$$\mathcal{L}_{\mathcal{T}_{sub}^*}(\mathcal{X}^t;\theta_s,\theta_i^t) = \sum_{i=1}^{Q}\mathcal{L}_{\tau_i*}(\mathcal{X}^t;\theta_s,\theta_i^t), \quad (5)$$

where $\mathcal{L}_{\tau_i*}$ belongs to the set that contains the Q single tasks from $\mathcal{T}_t$ that have the highest Q affinity scores calculated in Eq. 3. After the stage of task selection, the few-shot models are training with

$$\theta_s^{t+1} = \theta_s^t - \lambda\nabla_{\theta_s}\mathcal{L}_{\mathcal{T}_{sub}^*}(\mathcal{X}^t;\theta_s,\theta_i^t). \quad (6)$$

It should be noted that EFTS can be applied in any update during the entire training process and can be applied multiple times. The pseudo-code for the proposed EFTS is shown in Algorithm 1.

Algorithm 1: EFTS for Few-shot Learning

Input: Training set $\mathcal{D}$, evaluation set $\mathcal{V}$, learning rate λ, maximum number of iterations $Maxitr$, and number of evaluation tasks L, time-point set of task selection Θ

Output: $\theta_i^{Maxitr}, \theta_s^{Maxitr}$

1: randomly initialize the parameters θ_s and θ_i
2: randomly initialize the meta-training task $\mathcal{T}_{sub}^*$
3: **for** t in $\{1, 2, ..., Maxitr\}$ **do**
4:     Randomly sampling $\mathcal{X}^t$ from $\mathcal{D}$
5: **end for**
6: **for** l in $\{1, 2, ..., L\}$ **do**
7:     Randomly sampling $\mathcal{S}^l$ and $\mathcal{Q}^l$ from $\mathcal{V}$
8: **end for**
9: **for** t in $\{1, 2, ..., Maxitr\}$ **do**
10:     **if** $t \in \Theta$ **then**
11:         Calculate the average affinity $\hat{\mathcal{Z}}^{t,\text{UNA,M}}_{\mathcal{T}_{sub}\to\mathcal{T}_e}$ for each single task using Eqs. 2 and 3
12:         Calculate $\mathcal{L}_{\mathcal{T}_{sub}^*}(\mathcal{X}^t;\theta_s,\theta_i^t)$ using Eq. 5
13:     **end if**
14:     $\theta_s^{t+1} = \theta_s^t - \lambda\nabla_{\theta_s}\mathcal{L}_{\mathcal{T}_{sub}^*}(\mathcal{X}^t;\theta_s,\theta_i^t)$
15: **end for**
16: **return** $\theta_i^{Maxitr}, \theta_s^{Maxitr}$

---

5. Experimental Setup

5.1. Dataset

*mini*ImageNet. The *mini*ImageNet dataset [42] consists of 84×84 pixel images sourced from ILSVRC-2012. It comprises 100 classes, with each class containing 600 images. The class division for meta-training, meta-validation, and meta-testing involves randomly splitting them into 64, 16, and 20 classes, respectively. This dataset is commonly used in the field of few-shot learning research.

tiered-ImageNet. The *tiered*-ImageNet dataset [35] also originates from ILSVRC-2012 and contains 84×84 pixel images. In comparison to *mini*ImageNet, it offers a more extensive variety, encompassing 608 classes. These classes are further divided into 34 high-level categories, with each category housing 10 to 30 classes. The *tiered*-ImageNet dataset is widely utilized for evaluating and benchmarking few-shot learning methods. All classes are divided into 351, 97 and 160 classes for meta-training, meta-validation, and meta-testing, respectively.

CIFAR-FS. The CIFAR-FS dataset [5] comprises 32×32 pixel images extracted from CIFAR-100 [23]. It consists of 100 classes, with each class containing 600 images. CIFAR-FS is frequently employed in few-shot learning experiments as a valuable resource for evaluating the performance of various approaches. All classes are randomly divided into 64, 16 and 20 for meta-training, meta-validation, and meta-testing, respectively.

5.2. Task Setting for EFTS

In EFTS, a task set need to be designed for task selection. To ensure that our task settings can well simulate the situation of unknown tasks in practice, we do not expect each single task in this task set is promising. Here ten training tasks are used for EFTS: ProtoNet [38] (task1), Neighbourhood Component Analysis (NCA) [24] (task2), Classification (task3, task4), and supervised contrastive learning (SupCon) with different augmentation strategy [21] (task5-task10). For the classification tasks, the number of classes is the same as the number of features in task4, and in task5, we set the number of classes in line with [40]. For task5-task10, there uses 6 pairs of combinations between original samples and augmented samples with cropping, color distortion, Cutout, horizontal flipping.

We mainly consider that the data sampling strategy that all tasks share the same batch of data during each update process, and this data is collected in episodes that can be divided into a support set and a query set. For episodic tasks, these episodes can be applied directly; For non-episodic tasks, we merge the support set and query set into a single batch for usage. Like [24], the batchsize = $(M + N) \times C$, where C is the way in every batch. For example, when the training way is 16, and the number support samples and query samples in each class is 5 and 3, respectively, the batchsize is 128. This data sampling strategy is not completely episodic-free in practice because data collection imitates those of target tasks, even though the classifiers are arbitrary. However, in order to have all tasks adopt the same inputs, we still consider it as our primary experimental approach. We still consider the strategy that all tasks share different data inputs: batchsize without considering {M, N, C} for non-episodic tasks, and episodes for episodic tasks.

5.3. Head Setting for Evaluation

In target tasks, the selection of the head significantly affects the performance of the model [24, 40]. Here are multiple ways to configure evaluation task for few-shot classification, e. g., KNN, nearest centroid and soft assignments [24]. In line with [24], unless otherwise specified, ProtoNet as the default classifier will be used for the evaluation in this work. In addition, we considered using logistic regression (LR) [40] as an additional head for comparison.

5.4. Implementation

In training stage, the inputs contains training and validation sets from three datasets. For ProtoNet, the embeddings of training and evaluation samples are centred and normalised for all the tasks. Specifically, the centralization is implemented by $s_i \leftarrow s_i - \overline{s}$ where $\overline{s} = 1/|\mathcal{S}| \sum_j s_j$ for support samples, and $q_i \leftarrow q_i - \overline{s}$ for query samples. The normalization for all samples x is implemented by $x \leftarrow x/||x||$.

For LR head, the embeddings of training and evaluation samples are normalised for task1,task2 and task5-task10.

All the single tasks utilize ResNet-12 as the backbone, where the SGD optimizer is applied for the training. For the input sharing strategy, the learning rate is initially set as 0.1 decayed with a factor of 0.0025, 0.00032, respectively; For the strategy of not sharing inputs, the learning rate is initially set as 0.05 then decayed with a factor of 0.1. The processes of training and test are implemented in the Py-Torch machine learning package [32].

6. Experimental Results

6.1. Hyperparameter analysis

Impacts of update number per affinity. The impacts of update number per affinity (UNA) on the affinities of 10 tasks within 500 training steps after selecting the tasks at the beginning of training process are shown in Fig. 3. 128 batchsize and 16 way are applied in the model whose architecture is ResNet-12. As UNA is 5 or 10, the sequence of affinities pertaining to the task remains unsteady when the number M of affinity for averaging is small. As M increases, the affinity curves for UNA values at 5, 10, and 50 begin to stabilize progressively. In addition, as UNA increases, affinities become more stable while the step increases. For all of the UNA, the affinity relationship between tasks is relatively consistent. For example, overall both of task 3 and task 4 always have the lower affinities. In particular, we can see that the affinity of task1 (ProtoNet) is not the highest one, even in the evaluation tasks ProtoNet is also used as the classifier.

Furthermore, to illustrate the effectiveness our methods, we report the performances of single tasks for reference, as shown in Fig. 4. We can see that task 3, task 4 and task 7 perform the worst, and they also have the smallest affinities (see Fig. 3). In addition, task5-task10 perform the best, and for 1 shot tasks, they can correspond well to their affinities (see Fig. 3). However, task3 and task4 are inferior to task5, task6 and task8-task10 for 5 shot tasks, but have the highest affinities (right part of Fig. 3), probably because they converge quickly at the beginning of training.

Impacts of data sampling strategy. In multi-task learning scenarios that involve both episodic tasks and non-episodic tasks, we may adopt different data sampling strategies: 1) sharing a common input, i.e., episodes, for both episodic and non-episodic tasks; 2) using episodes as input for episodic tasks and using batches, without considering way, shot, and query, as input for non-episodic tasks. Additionally, the parameters of way, shot, and query can be set as hyper-parameters in an episode. For comparison convenience, we have fixed the batch size and compared these two strategies, and in episodes, we have fixed the sum of shot and query, as shown in Table 1. Table 1 shows that, the
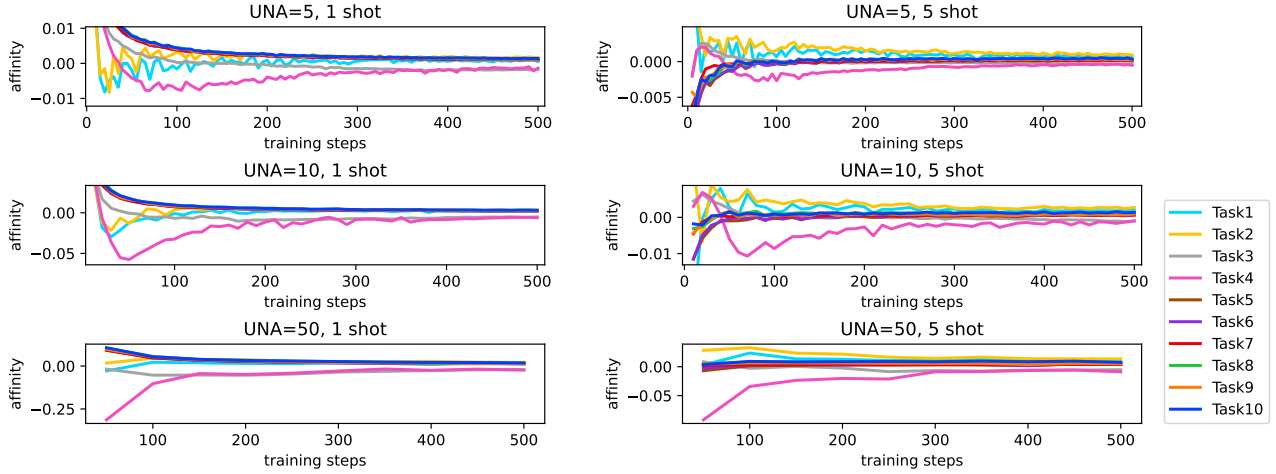
Figure 3. Average affinity against training steps with different Update Number per Affinity (UNA) on CIFAR-FS using ResNet-12. The batchsize is 128 and the training way is 16.

sum of shot and query can effect the performance of EFTS while the inputs of all tasks are episodes (inputs A). For example, when the batchsize is 256, the maximum difference in strategy inputs A can exceed 2%. As the inputs B is applied, the sum of shot and query has no significant impact on performance. Comparing two strategies, the inputs A may perform the best but depend on the setting of hyper-parameters, while the inputs B is still competitive and is almost unaffected by hyper-parameters.

Impacts of interval of task selection against Q. The interval between task selection with different number of jointly used tasks is discussed, as listed in Fig. 5. We can see that for both of 1 shot and 5 shot tasks, frequent selection of tasks during the training process does not improve the performance of the model, Selecting tasks only once at the beginning of training is enough. In addition, when the task is selected only once, the accuracy increases from Q = 1 to Q = 5, and decreases from Q = 5 to Q = 9. The reason may be that, with small Q, some effective tasks are abandoned (e. g., task 8 and task 10); With large Q, some laggard tasks are included (e. g., task 3 and task 4). These results imply that an appropriate Q ensures a small fault tolerance rate and leverage the advantages of multitasking learning.

6.2. Ablation Study

In order to demonstrate that the proposed EFTS can effectively select the training tasks, we conduct ablation experiments in this section. Firstly, we compare the performances of ProtoNet, EFTS and all task selection with different batchsize for 1 shot and 5 shot tasks, as listed in Table 2. The results show that EFTS at all cases outperforms the other two methods. The strategy for selecting all tasks may be simple, but it may introduce some tasks that have neg-
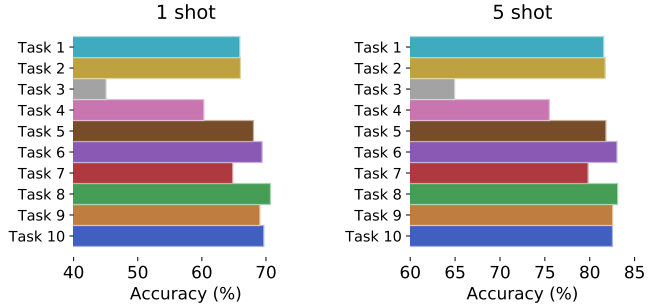

Figure 4. Average affinity against training steps with different Update Number per Affinity (UNA) on CIFAR-FS using ResNet-12. The batchsize is 128 and the training way is 16.

ative impacts on the model performance. In addition, the results supports the viewpoint that the best training task can be episodic-free, which is consistent with the results in [24].

As well as the strategy of all task selection, random task selection is also simple and would be effective. The random task selection can be repeated during the training process. Table 3 shows the performance comparison of EFTS and random task selection with different interval of task selection on *mini*ImageNet . The results in Table 3 shows that Whether the task is selected once or multiple times, selecting tasks using the EFTS criterion is better than randomly selecting tasks for 1 shot and 5 shot tasks. Although in the worst-case scenario, the accuracy obtained by random task selection is only 2% lower than that of EFTS, its performance depends on the task types in the task set. That is to say, when more tasks in the task set are unhelpful, the performance of random task selection may be worse.
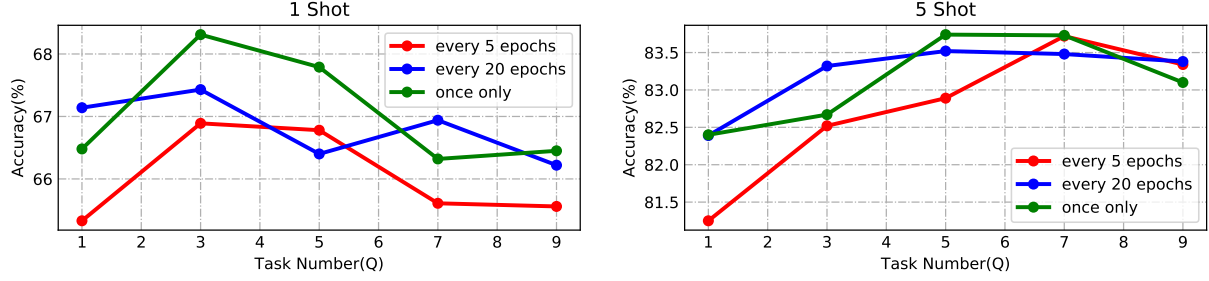
Figure 5. Accuracies (%) obtained by EFTS with different interval of task selection against different Q on CIFAR-FS using ResNet-12. The batchsize is 128 and the training way is 16. In addition, NUA = 50 and M = 4.

Table 1. Test accuracies (%) obtained by EFTS with different batchsize, training shot (ts) and training query (tq) for episodic tasks on CIFAR-FS, where Q = 5, UNA = 50, M = 4 and tasks are selected once only. A: All tasks sharing the same data inputs; B: Non-episodic tasks share different data inputs with episodic tasks, without considering ts and tq. In addition, NUA = 50 and M = 4.

batchsize	inputs	ts+tq	1 shot	5 shot
128	A	8	$67.79_{\pm0.96}$	$83.74_{\pm0.65}$
	A	16	$68.92_{\pm0.94}$	$83.86_{\pm0.68}$
	A	32	$65.80_{\pm0.97}$	$82.82_{\pm0.71}$
	B	8	$68.23_{\pm0.95}$	$83.09_{\pm0.64}$
	B	16	$68.12_{\pm0.94}$	$83.23_{\pm0.67}$
	B	32	$68.08_{\pm0.95}$	$83.04_{\pm0.64}$
256	A	8	$67.90_{\pm0.98}$	$82.48_{\pm0.71}$
	A	16	$69.18_{\pm0.94}$	$84.34_{\pm0.68}$
	A	32	$69.40_{\pm0.94}$	$84.70_{\pm0.67}$
	B	8	$69.34_{\pm0.95}$	$84.21_{\pm0.66}$
	B	16	$69.96_{\pm0.97}$	$84.14_{\pm0.63}$
	B	32	$69.55_{\pm0.94}$	$84.23_{\pm0.65}$

Table 2. Comparison of accuracies (%) obtained by three methods with different batchsize on CIFAR-FS. In addition, NUA = 50 and M = 4.

batchsize	method	1 shot	5 shot
128	ProtoNet	$65.92_{\pm0.97}$	$81.56_{\pm0.72}$
	EFTS	$68.31_{\pm0.98}$	$83.74_{\pm0.65}$
	all tasks	$67.04_{\pm0.99}$	$83.10_{\pm0.70}$
256	ProtoNet	$67.39_{\pm0.93}$	$83.20_{\pm0.69}$
	EFTS	$69.40_{\pm0.94}$	$84.70_{\pm0.67}$
	all tasks	$66.25_{\pm0.94}$	$81.75_{\pm0.74}$

6.3. Comparison with State-of-the-art

In the section, we report our results based on ResNet-12 and compare them with two kinds of recent representative methods that includes episodic training methods and non-episodic training methods on three commonly used datasets, as shown Table 4. Here we adopt ProtoNet head and LR

Table 3. Comparison of accuracies (%) obtained by EFTS and random task combinations on *mini*ImageNet, where the batchsize is 128. In addition, NUA = 50 and M = 4.

interval	method	1 shot	5 shot
every 5 epochs	Random	$57.00_{\pm0.90}$	$74.70_{\pm0.69}$
	EFTS	$57.87_{\pm0.90}$	$76.30_{\pm0.28}$
every 20 epochs	Random	$56.44_{\pm0.88}$	$75.34_{\pm0.68}$
	EFTS	$58.26_{\pm0.87}$	$76.92_{\pm0.67}$
once only	Random	$57.66_{\pm0.87}$	$74.91_{\pm0.69}$
	EFTS	$58.91_{\pm0.88}$	$76.90_{\pm0.65}$

head for comparison. For protoNet head, we employed the strategy of sharing inputs, while for LR head, we utilized a strategy of not sharing inputs. In comparison of [40], we did not compare against results after knowledge distillation because it is an embeddable method that can be applied to most methods including ours. In addition, for ProtoNet, the way is 16, and batchsize is 256; for LR head, the batchsize is 64. For all the heads, we applied data augmentation on the inputs for task1-task5. In Table 4, we can see that EFTS with LR head outperforms ProtoNet head and other methods, which is benefited from task4 (classification) that is proposed in [40]. Nevertheless, these results of the proposed EFTS with ProtoNet head is comparable to [24] that applied a greater batchsize. Overall, these results show that task selection strategies are competitive to the conventional fixed task strategies.

It should be noted that our goal is not to improve the state-of-the-art performance, but rather to demonstrate the practicality of EFTS in the context of related methods. In light of this aim, and to render our approach more comprehensible, we choice the often used methods rather than those with superior performance but not yet representative as the candidates in the task set. This choice may limit the performance of EFTS, and thus we did not employ more methods for comparison. Nonetheless, a further improved performance can be achieved by adding some stunning approaches (such as [10, 19, 46, 47]) in the task set .

Table 4. Accuracy (%) comparison with the state-of-the-art for 5-way tasks on *mini*ImageNet with 95% confidence intervals. ‡ denotes that validation set as well as training set is used for training. For EFTS, NUA = 50 and M = 4.

Model	Backbone	*mini*ImageNet 1-shot	*mini*ImageNet 5-shot	*tiered*-ImageNet 1-shot	*tiered*-ImageNet 5-shot	CIFAR-FS 1-shot	CIFAR-FS 5-shot
Episodic methods							
ProtoNet[38]	ResNet-12	$59.25_{\pm0.64}$	$75.60_{\pm0.48}$	$61.74_{\pm0.77}$	$80.00_{\pm0.55}$	$72.2_{\pm0.7}$	$83.5_{\pm0.5}$
TADAM[30]	ResNet-12	$58.50_{\pm0.30}$	$76.70_{\pm0.30}$	-	-	-	-
AdaResNet[29]	ResNet-12	$56.88_{\pm0.62}$	$71.94_{\pm0.57}$	-	-	-	-
LwoF[16]	WRN-28-10	$60.06_{\pm0.14}$	$76.39_{\pm0.11}$	-	-	-	-
DSN[37]	ResNet-12	$62.64_{\pm0.66}$	$78.83_{\pm0.45}$	$66.22_{\pm0.75}$	$82.79_{\pm0.48}$	$72.3_{\pm0.7}$	$85.1_{\pm0.5}$
CTM[27]	ResNet-18	$62.05_{\pm0.55}$	$78.63_{\pm0.06}$	$64.78_{\pm0.11}$	$81.05_{\pm0.52}$	-	-
Hyper ProtoNet[22]	ResNet-18	$59.47_{\pm0.20}$	$76.84_{\pm0.14}$	-	-	$64.02_{\pm0.24}$	$82.53_{\pm0.14}$
MetaOptNet-RR[25]	ResNet-12	$61.41_{\pm0.61}$	$77.88_{\pm0.46}$	$65.36_{\pm0.71}$	$81.34_{\pm0.52}$	$72.6_{\pm0.7}$	$84.3_{\pm0.5}$
MetaOptNet-SVM[25]	ResNet-12	$62.64_{\pm0.61}$	$78.63_{\pm0.46}$	$65.99_{\pm0.72}$	$81.56_{\pm0.53}$	$72.0_{\pm0.7}$	$84.2_{\pm0.5}$
Ravichandran et al.[34]	ResNet-12	$59.04_{\pm-}$	$77.64_{\pm-}$	$66.87_{\pm-}$	$82.64_{\pm-}$	$69.15_{\pm-}$	$84.7_{\pm-}$
CAN[18]	ResNet-12	$\mathbf{63.85_{\pm0.48}}$	$79.44_{\pm0.34}$	$69.89_{\pm0.51}$	$84.23_{\pm0.37}$	-	-
Non-episodic methods							
Baseline[8]	ResNet-18	$51.75_{\pm0.80}$	$74.27_{\pm0.63}$	-	-	$65.51_{\pm0.87}$	$82.85_{\pm0.55}$
Baseline++[8]	ResNet-18	$51.87_{\pm0.77}$	$75.68_{\pm0.63}$	-	-	$67.02_{\pm0.90}$	$83.58_{\pm0.54}$
NCA nearest centroid[24]	ResNet-12	$62.55_{\pm0.12}$	$78.27_{\pm0.09}$	$68.35_{\pm0.13}$	$83.20_{\pm0.10}$	$72.49_{\pm0.12}$	$85.15_{\pm0.09}$
RFIC-simple [40]	ResNet-12	$62.02_{\pm0.63}$	$79.64_{\pm0.44}$	$69.74_{\pm0.72}$	$84.41_{\pm0.55}$	$71.5_{\pm0.8}$	$86.0_{\pm0.5}$
Our episodic-free methods							
EFTS(ProtoNet for target tasks)	ResNet-12	$61.77_{\pm0.89}$	$79.27_{\pm0.62}$	$69.46_{\pm0.96}$	$84.30_{\pm0.66}$	$72.92_{\pm0.94}$	$85.74_{\pm0.64}$
EFTS(LR for target tasks)	ResNet-12	$63.77_{\pm0.85}$	$\mathbf{79.82_{\pm0.55}}$	$\mathbf{70.11_{\pm0.89}}$	$\mathbf{84.89_{\pm0.64}}$	$\mathbf{74.85_{\pm0.84}}$	$\mathbf{87.41_{\pm0.59}}$

7. Is validation data selection critical?

Intuitively, different tasks may be selected when using cross-domain data for validation. Moreover, if the distribution of samples used for validation is different from that of test samples, the selection of tasks deviates from the target and would have a negative impact on the model's performance in the target task. Here we will discuss whether the selection of validation sets is crucial. In Table 5, we compare the performance respectively using CIFAR-FS and *mini*ImageNet for evaluation while using CIFAR-FS for training. Counterintuitively, we see that the performance of the model is not sensitive to the selection of validation samples, regardless of whether multiple task selections were made. Similar results are hold in [13], which shows that the inter-task affinity scores computed on the validation set is highly similar to those computed on the training set. Nevertheless, we still need to emphasize that these results does not deny the importance of validating data selection in all the cases.

8. Conclusion and Future Work

In this work, we propose a framework called EFTS for few-shot learning. EFTS supports effective episodic-free tasks for training meta-learners, which thus expands the episodic training paradigm typically used for few-shot learning. In

Table 5. Train-data of CIFAR-FS for training and different data sources including train-data of CIFAR-FS, evaluation-data of CIFAR-FS and *mini*ImageNet for evaluating, where Q = 5 and batchsize is 128. In addition, NUA = 50 and M = 4.

interval	evaluation source	1 shot	5 shot
once only	*mini*ImageNet	$67.30_{\pm0.98}$	$82.18_{\pm0.71}$
	CIFAR-FS	$68.18_{\pm0.95}$	$82.87_{\pm0.72}$
every 5 epochs	*mini*ImageNet	$68.01_{\pm0.93}$	$82.88_{\pm0.71}$
	CIFAR-FS	$67.56_{\pm0.97}$	$82.60_{\pm0.70}$

EFTS, the episodic-free tasks are selected as the training tasks from a task set, according to the their benefits on a series of episodic tasks. We analyzed the impact of its hyper-parameters and demonstrated the effectiveness of EFTS through ablation experiments and comparison experiments with state-of-the-art methods.

There are two directions worth studying in future work. Firstly, the evaluation for task selection can be developed to make more accurate predictions of long-term impacts of a task subset on the performance of meta-learners. Secondly, our work makes it possible to more effectively select multiple tasks for few-shot learning. The task set including homogeneous and heterogeneous tasks used for task selection can be improved in future work through task engineering.

References

[1] Prachi Agrawal, Hattan F Abutarboush, Talari Ganesh, and Ali Wagdy Mohamed. Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *Ieee Access*, 9:26766–26791, 2021. 4

[2] Kelsey Allen, Evan Shelhamer, Hanul Shin, and Joshua Tenenbaum. Infinite mixture prototypes for few-shot learning. In *International conference on machine learning*, pages 232–241. PMLR, 2019. 1, 2

[3] Han Altae-Tran, Bharath Ramsundar, Aneesh S Pappu, and Vijay Pande. Low data drug discovery with one-shot learning. *ACS central science*, 3(4):283–293, 2017. 1

[4] Mohammed A Awadallah, Mohammed Azmi Al-Betar, Abdelaziz I Hammouri, and Osama Ahmad Alomari. Binary jaya algorithm with adaptive mutation for feature selection. *Arabian Journal for Science and Engineering*, 45(12):10875–10890, 2020. 4

[5] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018. 4

[6] Quentin Bouniot, Ievgen Redko, Romaric Audigier, Angélique Loesch, and Amaury Habrard. Improving few-shot learning through multi-task representation learning theory. In *European Conference on Computer Vision*, pages 435–452. Springer, 2022. 3

[7] Jiaxin Chen, Xiao-Ming Wu, Yanke Li, Qimai Li, Li-Ming Zhan, and Fu-lai Chung. A closer look at the training strategy for modern meta-learning. *Advances in Neural Information Processing Systems*, 33:396–406, 2020. 1

[8] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019. 1, 2, 8

[9] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9062–9071, 2021. 2

[10] Guneet S. Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations*, 2020. 7

[11] Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020. 1, 3

[12] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 1

[13] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34:27503–27516, 2021. 2, 3, 4, 8

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

[15] Stanislav Fort. Gaussian prototypical networks for few-shot learning on omniglot. *arXiv preprint arXiv:1708.02735*, 2017. 2

[16] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 8

[17] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021. 1

[18] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen. Cross attention network for few-shot classification. In *NIPS*, pages 4005–4016, 2019. 8

[19] D. Kang, H. Kwon, J. Min, and M. Cho. Relational embedding for few-shot classification. In *ICCV*, pages 8822–8833, 2021. 7

[20] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018. 3

[21] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 5

[22] V. Khrulkov, L. Mirvakhabova, E. Ustinova, and I. Oseledets. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418—-6428, 2020. 8

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[24] Steinar Laenen and Luca Bertinetto. On episodes, prototypical networks, and few-shot learning. *Advances in Neural Information Processing Systems*, 34:24581–24592, 2021. 1, 2, 3, 5, 6, 7, 8

[25] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10657–10665, 2019. 1, 8

[26] Carson K Leung, Daryl LX Fung, and Calvin SH Hoi. Health analytics on covid-19 data with few-shot learning. In *Big Data Analytics and Knowledge Discovery: 23rd International Conference, DaWaK 2021, Virtual Event, September 27–30, 2021, Proceedings 23*, pages 67–80. Springer, 2021. 1

[27] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1–10, 2019. 8

[28] Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 741–756. Springer, 2020. 2

[29] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pages 3664–3673, 2018. 8

[30] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018. 1, 2, 8

[31] Frederik Pahde, Mihai Puscas, Tassilo Klein, and Moin Nabi. Multimodal prototypical networks for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2644–2653, 2021. 2

[32] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[33] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019. 1

[34] A. Ravichandran, R. Bhotika, and S. Soatto. Few-shot learning with embedded class models and shot-free meta training. In *ICCV*, pages 331–339, 2019. 8

[35] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018. 4

[36] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018. 2, 3

[37] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4136–4145, 2020. 1, 2, 8

[38] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 2, 5, 8

[39] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018. 1, 2

[40] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 266–282. Springer, 2020. 1, 2, 5, 7, 8

[41] Richa Upadhyay, Prakash Chandra Chhipa, Ronald Phlypo, Rajkumar Saini, and Marcus Liwicki. Multi-task meta learning: learn how to adapt to unseen tasks. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2023. 2, 3

[42] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3637–3645, 2016. 1, 2, 3, 4

[43] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020. 1

[44] Jiachen Yang, Xiaolan Guo, Yang Li, Francesco Marinello, Sezai Ercisli, and Zhuo Zhang. A survey of few-shot learning in smart agriculture: developments, applications, and challenges. *Plant Methods*, 18(1):1–12, 2022. 1

[45] Zhanyuan Yang, Jinghua Wang, and Yingying Zhu. Few-shot classification with contrastive learning. In *European Conference on Computer Vision*, pages 293–309. Springer, 2022. 1

[46] C. Zhang, Y. Cai, G. Lin, and C. Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, pages 12200–12210, 2020. 7

[47] Yang Zhanyuan, Wang Jinghua, and Zhu Yingying. Few-shot classification with contrastive learning. In *ECCV(2022)*, pages 1–14, 2022. 7