Data-Efficient and Robust Task Selection for Meta-Learning

Donglin Zhan
Columbia University
New York, NY
donglin.zhan@columbia.edu

James Anderson
Columbia University
New York, NY
james.anderson@columbia.edu

Abstract

Meta-learning methods typically learn tasks under the assumption that all tasks are equally important. However, this assumption is often not valid. In real-world applications, tasks can vary both in their importance during different training stages and in whether they contain noisy labeled data or not, making a uniform approach suboptimal. To address these issues, we propose the Data-Efficient and Robust Task Selection (DERTS) algorithm, which can be incorporated into both gradient and metric-based meta-learning algorithms. DERTS selects weighted subsets of tasks from task pools by minimizing the approximation error of the full gradient of task pools in the meta-training stage. The selected tasks are efficient for rapid training and robust towards noisy label scenarios. Unlike existing algorithms, DERTS does not require any architecture modification for training and can handle noisy label data in both the support and query sets. Analysis of DERTS shows that the algorithm follows similar training dynamics as learning on the full task pools. Experiments show that DERTS outperforms existing sampling strategies for meta-learning on both gradient-based and metric-based meta-learning algorithms in limited data budget and noisy task settings.

1. Introduction

Meta-learning methods have been extensively studied and applied in computer vision, natural language processing, and robotics [9]. The key idea of meta-learning is to mimic the few-shot situations faced at test time by randomly sampling classes in meta-training data to construct tasks for episodic training. Most existing meta-learning methods randomly sample meta-training tasks with a uniform probability of learning the meta-model during meta-training [6, 33, 36]. The assumption behind such uniform sampling is that all tasks are equally contributed. In real-world scenarios, things often differ. Considering that the importance of tasks can change during the training process, the diversity of tasks, and the probability of mislabeled data within tasks, some tasks might be more informative for the meta-training process than others. In this paper, we specifically focus on two scenarios: a limited data budget and a noisy task setting.

First, making the meta-training process more efficient is an essential issue, particularly in scenarios with a limited budget of data (e.g., the number of tasks and data learned in the meta-training process). Current meta-learning algorithms require many tasks for meta-training episodes, which may contain redundant information. In this work, we raise the natural concern that not all the tasks are created equal. A coarse-grained task that contains the classification of “Dog" and “Laptop" is much easier to learn for the meta-model than the fine-grained task that includes a more difficult classification (e.g., “Dog" or “Cat") [19]. In meta-training episodes, selecting the most benignly informative subset of tasks could benefit this process by decreasing the computational load.

Second, most meta-learning methods presuppose that the few support set training samples were chosen correctly to represent their class [6, 33, 36]. Unfortunately, such assurances are not usually provided in real-world scenarios. In reality, mislabeled samples are frequently present in even highly annotated and maintained datasets [8, 17, 27, 37] as a result of automated weakly supervised annotation, ambiguity, or even human error. Suppose some tasks with noisy labels are fed to the meta-training process. In that case, the noisy-labeled data in the support set will hinder the adaptation step of the meta-training process, which can result in invalid adaptation and further make the base model incorrect on query evaluation. Furthermore, if noisy data is also in the query set, the misleading gradient would be updated for the meta-model, which hinders the generalization capacity of the meta-model during the meta-testing stage. Therefore, developing methods to avoid training meta-models on heavily label-corrupted tasks is essential for the setting with noisy data as well.

The two aspects mentioned above still need to be thoroughly studied. (i) For limited computational and data budgets, existing approaches construct information-theoretic criteria for evaluating and selecting task [10, 16, 22] or building extra module for learning the sampling probability [46] with

high computation cost. (ii) Previous work that considers meta-learning with noisy data [18, 23, 46] make the assumption that noisy data only exist in the support set of tasks. This assumption may be unrealistic for applications under the uncertainly noisy setting (e.g., tasks with different noise ratios, noisy data could be in both support sets and query sets, and identities are not exposed to models).

We propose the *Data-Efficient and Robust Task Selection (DERTS)* algorithm for meta-learning that can select appropriate tasks in the meta-training stage with efficiency for rapid training *and* robustness towards noisy data scenarios inspired by [3, 26]. Unlike existing works for task sampling, DERTS does not require any architecture modification; it only requires an iterative task pool to store tasks for the model to select. DERTS selects weighted subsets of tasks from task pools by minimizing the approximation error of the full gradient of task pools in the meta-training stage. Furthermore, by dropping tasks in the subset with potentially high estimated gradient norms, we find the proposed algorithm is robust toward the noisy task scenario.

Contributions: We propose a data-efficient and robust task selection algorithm for meta-learning in limited data budget and noisy label settings.

- We formulate a weighted subset selection objective that minimizes the approximation error of the full gradient on episodic task pools. Due to the submodularity of the approximation error, we apply a stochastic greedy approximation to the solution of the derived optimization objective. This method can be easily incorporated into both gradient-based and metric-based meta-learning schemes.
- We extend the selection algorithm to a challenging noisy task setting with mislabeled data in both the support and query sets by dropping tasks with a large estimated gradient norm.
- We provide a theoretical analysis that proves that learning on a subset of tasks produces similar training dynamics than if trained on the full task pool. We provide an upper-bound the difference between the model trained with our task selection algorithm and the model trained with all the meta-training tasks. Our result highlights a fundamental bias when applying *any* selection method.
- Extensive experiments show that DERTS outperforms the state-of-the-art sampling strategies for meta-learning with both gradient-based and metric-based meta-learning algorithms on a limited budget and noisy task settings. The performance improvement averages between 3% and 5%, while also achieving a speedup of more than three times.

2. Related Work

Meta-Learning focuses on rapidly adapting to new unseen tasks by learning prior knowledge through training on many similar tasks. Metric-based methods classify query examples based on their similarity to each class's support data, learning a transferable embedding space for evaluation and prediction [7, 18, 30, 36, 39, 47]. Other than metric-based approaches, optimization-based methods fine-tune model parameters on a few support examples [1, 6, 29, 33, 38, 42, 43, 48]. However, most previous work assumes tasks contribute equally to the meta-training stage. Recently, some work has focused on optimizing the sampling distribution strategy for meta-learning [10, 19, 22]. Arnold et al. [2] apply a uniform episodic sampler (US) to reweight tasks based on the observation that the task learning difficulty follows a normal distribution for arbitrary meta-datasets and model architectures. Although promising, the empirical assertion that US is based on may face challenges when encountering tasks with disturbances and uncertainties. Yao et al. [46] propose an adaptive task scheduler (ATS) to jointly learn the sampling probability for tasks in the candidate pool to address the noisy label issue. Specifically, ATS takes the loss value and inner product of the gradient on the support set and query set and outputs the corresponding sampling probability. The robustness of ATS is based on the assumption that the noise only exists in the support set. When breaking this condition, ATS is likely to be less robust in the noisy task setting. Unlike US and ATS, our algorithm, DERTS, focuses on both data-efficiency *and* robustness. For the data-efficient perspective, DERTS aims to approximate the performance of full training episodes via only training on a subset of tasks. As for the robustness issue, we allow for the fact that the noisy data can be present in the support set *and* query set without any side information; this poses significant challenges in the evaluation of robustness.

With respect to data selection and sampling aspects of the problem, there have been efforts to take advantage of the difference in importance among various samples to reduce the variance and improve the convergence rate of stochastic optimization methods. Those that apply to overparameterized models employ either the gradient norm [12] or the loss function [21, 35] to compute each sample's importance. Recently, a series of data selection strategies have discussed selecting subsets of the data for efficient training. CRAIG [25] finds the subsets by greedily maximizing a submodular function and provides convergence guarantees to a neighborhood of the optimal solution for both convex and non-convex models. GRADMATCH [14] proposes a variation to address the same objective using orthogonal matching pursuit. CREST [45] models the non-convex loss as a series of quadratic functions for extracting subset. However, the work mentioned above focuses on the standard supervised learning setting and it is not clear how such approaches can be ported over to task selection for episodic training in meta-learning due to their formulation as a bilevel-optimization problem.

3. Background

We consider the standard meta-learning setting, where given a set of training tasks $\mathcal{T}_1, \ldots, \mathcal{T}_n$ sampled from task distribution $p(\mathcal{T})$, we would like to learn a good parameter

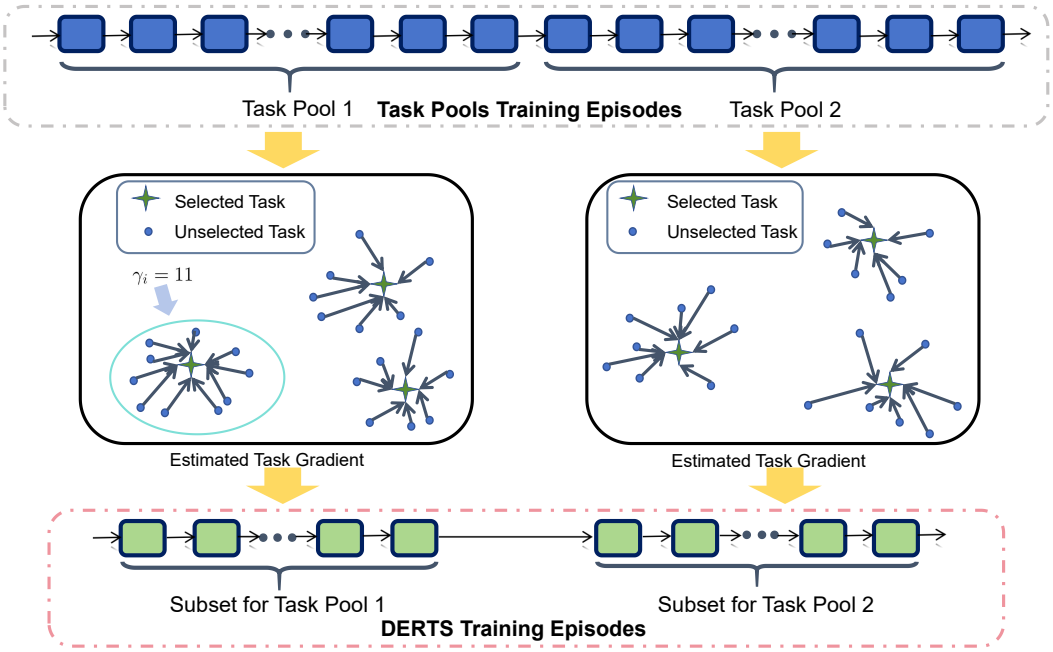


Figure 1. DERTS requires task pools to store episodic tasks sampled from task distributions. With the efficient gradient estimation in sec.4.2, the gradients of all the tasks stored in the task pool are computed. According to the approximation formulated in sec. 4.1 and optimization objective in sec. 4.2, a subset of tasks with corresponding weights is constructed to approximate the task pool gradient. The meta-model then conducts a training process on the subsets instead of task pools.

initialization θ^* for a predictive model f_θ such that it can be quickly adapted to new tasks given only a limited amount of data (i.e., few-shot regime). Each task $\mathcal{T}_i$ has a support set of labeled data $\mathcal{D}_i^s = \{\mathbf{X}_i^s, \mathbf{Y}_i^s\} = \left\{\left(\mathbf{x}_{i,j}^s, \mathbf{y}_{i,j}^s\right)\right\}_{j=1}^{N^s}$ and a query set, $\mathcal{D}_i^q = \{\mathbf{X}_i^q, \mathbf{Y}_i^q\} = \left\{\left(\mathbf{x}_{i,j}^q, \mathbf{y}_{i,j}^q\right)\right\}_{j=1}^{N^q}$ of labeled data, where N_s and N_q refer to the size of support and query sets respectively. Given the predictive model f_θ, meta-learning algorithms first train the base model on meta-training tasks. Then, during the meta-testing stage, the well-trained base model is applied to the new task $\mathcal{T}_t$ by taking a few adaptation steps on its support set $\mathcal{D}_t^s$. Finally, the performance is evaluated on the query set $\mathcal{D}_t^q$. We provide a brief introduction to gradient-based and metric-based algorithms in **Appendix A**.

4. Efficient and Robust Task Selection

The conceptual idea behind DERTS is to (i) select subsets of tasks from the overall task pools, (ii) assign a weight to each task in the subset that captures their relative importance, and (iii) use the tasks in the subset for meta-training with corresponding weights. We first formulate a full gradient approximation for the task pools in sec. 4.1. Secondly, in sec. 4.2, we establish an optimization objective for task selection and provide the solution based on submodular maximization. In addition, we provide a modified DERTS to address the scenario with noisy label tasks in sec. 4.3. We also provide a theoretical analysis of DERTS in sec. 4.4.

Figure 1 demonstrates the workflow of DERTS. We also note that DERTS can be easily incorporated into widely-used gradient-based and metric-based meta-learning schemes.

4.1. Full Gradient Approximation for Episodic Task Pools

We start by selecting a sample of candidate tasks drawn from the task distribution $p(\mathcal{T})$ in advance and store them in a task pool $\mathcal{M}$. Suppose for a task pool $\mathcal{M} := \{\mathcal{T}_j | j = 1, 2, \ldots, N\}$, we want to select a subset of tasks $\mathcal{S} := \{\mathcal{T}_i \mid i = \alpha_1, \alpha_2, \ldots, \alpha_k\}$, where $\alpha_i \in [N]$ and $k < N$, with corresponding weights $\{\gamma_i \mid i = 1, 2, \ldots, k\}$ such that the gradient for training on $\mathcal{S}$ approximates the gradient on $\mathcal{M}$. We now describe our approach for determining the weights: Let $\Gamma : \mathcal{M} \to \mathcal{S}$ be a mapping from the task pool $\mathcal{M}$ to the subset $\mathcal{S}$ that maps a task $\mathcal{T}_j$ from $\mathcal{M}$ to a task $\mathcal{T}_i$ in $\mathcal{S}$. For simplicity, we denote $\Gamma(\mathcal{T}_j) = \mathcal{T}_i$ as $\Gamma(j) = i$ and $\mathcal{T}_j \in \mathcal{M}$ as $j \in \mathcal{M}$. Let $\mathcal{S}_{\mathcal{M}}^C$ denote the complement of $\mathcal{S}$ in $\mathcal{M}$. Similar to [3], we define the weight γ_i of the selected task $\mathcal{T}_i \in \mathcal{S}$ as

$$\gamma_i := \sum_{j \in \mathcal{M}} 1_{\{\Gamma(j)=i\}} = \sum_{j \in \mathcal{S}} 1_{\{\Gamma(j)=i\}} + \sum_{j \in \mathcal{S}_{\mathcal{M}}^C} 1_{\{\Gamma(j)=i\}}, \tag{1}$$

where $1_{\mathcal{C}}$ is the indicator function for the set $\mathcal{C}$. The first term on the RHS of eq. (1) is equal to 1 since Γ is identity in $\mathcal{S}$. For the second term, by taking the summation over $e \in \mathcal{S}_M^C$, we are calculating how many elements in $\mathcal{S}_M^C$ are

mapped into task $\mathcal{T}_i$ in $\mathcal{S}$.

As we don't know $\mathcal{S}$ we cannot compute γ_i directly from eq. (1). Instead we formulate an optimization problem for establishing how to select the set $\mathcal{S}$ from $\mathcal{M}$ in such a manner that when training is performed on the tasks in $\mathcal{S}$ the training dynamics (i.e., the gradients at each iteration) approximate the gradients that would arise had we trained on the full task pool $\mathcal{M}$. As in the standard meta-training paradigm, we explore making the approximation of each task's gradient on the loss function of the task-adapted model ϕ_i (i.e., $\phi_i = \theta - \eta\nabla_\theta\mathcal{L}\left(f_\theta; \mathcal{D}_i^s\right)$) on the query set $\mathcal{D}_i^q$ that is updated by the meta-model θ. The gradient on task $\mathcal{T}_i$'s query is defined as $\nabla_\theta\mathcal{L}\left(f_{\phi_i}; \mathcal{D}_i^q\right) = \sum_{j=1}^{N^q} \nabla_\theta\mathcal{L}\left(f_{\phi_i}; \left(\mathbf{x}_{i,j}^q, \mathbf{y}_{i,j}^q\right)\right)$. Using the definition of the mapping function Γ as defined above, DERTS uses the approximate gradient

$$\sum_{j\in\mathcal{M}} \nabla_\theta\mathcal{L}\left(f_{\phi_{\Gamma(j)}}; \mathcal{D}_{\Gamma(j)}^q\right) = \sum_{j\in\mathcal{S}} \gamma_j \nabla_\theta\mathcal{L}\left(f_{\phi_j}; \mathcal{D}_j^q\right),$$

which is taken over the set $\mathcal{S}$. In contrast, standard meta-learning algorithms use the full gradient on $\mathcal{M}$.

The gradient error incurred by training on $\mathcal{S}$ instead of $\mathcal{M}$ is given by

$$\begin{aligned}&\left\|\sum_{j\in\mathcal{M}} \nabla_\theta\mathcal{L}\left(f_{\phi_j}; \mathcal{D}_j^q\right) - \sum_{j\in\mathcal{M}} \nabla_\theta\mathcal{L}\left(f_{\phi_{\Gamma(j)}}; \mathcal{D}_{\Gamma(j)}^q\right)\right\| \\ &= \left\|\sum_{j\in\mathcal{M}} \nabla_\theta\mathcal{L}\left(f_{\phi_j}; \mathcal{D}_j^q\right) - \sum_{i\in\mathcal{S}} \gamma_i \nabla_\theta\mathcal{L}\left(f_{\phi_i}; \mathcal{D}_i^q\right)\right\|.\end{aligned} \tag{2}$$

Ideally, we would like to make this error as small as possible. Unfortunately, we do not yet know Γ and hence S, and so we cannot evaluate the error, and directly optimizing over $\mathcal{S}$ is NP-hard. However, we can upper bound the RHS of (2) by a function that can be optimized over, specifically we have that

$$\begin{aligned}&\left\|\sum_{j\in\mathcal{M}} \nabla_\theta\mathcal{L}\left(f_{\phi_j}; \mathcal{D}_j^q\right) - \sum_{i\in\mathcal{S}} \gamma_i \nabla_\theta\mathcal{L}\left(f_{\phi_i}; \mathcal{D}_i^q\right)\right\| \\ &\leq \sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}} \left\|\nabla_\theta\mathcal{L}\left(f_{\phi_j}; \mathcal{D}_j^q\right) - \nabla_\theta\mathcal{L}\left(f_{\phi_i}; \mathcal{D}_i^q\right)\right\|\end{aligned} \tag{3}$$

which naturally leads to defining the approximation criteria as minimizing the RHS of (3). According to inequality (3), by assuming $\mathcal{S}$ is fixed, assigning the mapping Γ to map the task in $\mathcal{M}$ to the closest element in $\mathcal{S}$ in the gradient space will minimize and upper bound on the gradient approximation error. Thus, γ_i associated with mapping Γ is computed as

$$\gamma_i = \sum_{j\in\mathcal{M}} 1_{\left\{j=\operatorname{argmin}_{\mathcal{T}_i\in\mathcal{S}} \left\|\nabla_\theta\mathcal{L}\left(f_\phi;\mathcal{D}_i^q\right)-\nabla_\theta\mathcal{L}\left(f_\phi;\mathcal{D}_j^q\right)\right\|\right\}}.$$

4.2. Extracting Subsets Efficiently

Computing the explicit task gradient $\nabla_\theta\mathcal{L}\left(f_{\phi_j}; \mathcal{D}_j^q\right)$ that updates the meta-model is time-consuming and incurs a large computational cost. As shown in [12], the variation of the gradient norm is mainly captured by the gradient of the loss function with respect to the pre-activation outputs of the last layer. Suppose for a few-shot classification task, the above estimation only requires a forward computation on the last layer. E.g., for a softmax layer as the last, the gradients of the loss with respect to the input of the softmax layer for $\left(\mathbf{x}_{i,j}^q, \mathbf{y}_{i,j}^q\right)$ is l_i-y_i, where l_i is the logits and y_i is the encoded label. We extend the result of [45] to estimate the task-gradient $\nabla_\theta\mathcal{L}\left(f_{\phi_i}; \mathcal{D}_i^q\right)$ and denote it as $\tilde{g}_i$. This approximation indicates the computation of gradient estimation $\tilde{g}_i$ on task $\mathcal{T}_i$ is marginally more costly to compute than the value of the loss. We show the extended results and analysis in **Appendix B**.

Minimizing the RHS of (3) is mathematically equivalent to maximizing a well-known submodular function, i.e., the facility location function [4].

Definition 1 (Submodularity)**.** *A set function $F : 2^V \to \mathbb{R}^+$ is submodular if $F(e \mid S) := F(S\cup\{e\}) - F(S) \geq F(T\cup\{e\}) - F(T)$, for any $S \subseteq T \subseteq V$ and $e \in V\backslash T$. F is monotone if $F(e \mid S) \geq 0$ for any $e \in V\backslash\bar{S}$ and $S \subseteq V$.*

According to Definition 1 and eq. (3), we define a monotone submodular function $\mathcal{F}$ over $\mathcal{S}$ with estimated gradient $\tilde{g}_i$:

$$\mathcal{F}(\mathcal{S}) := C - \sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}} \|\tilde{g}_j - \tilde{g}_i\|$$

where C is a constant to upper bound $\mathcal{F}(\mathcal{S})$. To formulate our task selection objective, we follow the logic of [32]. To restrict the size of $\mathcal{S}$, we limit the number of selected tasks in $\mathcal{S}$ to be no greater than K, i.e., $|\mathcal{S}| \leq K$. Thus, the submodular maximization form of selection objective is:

$$\mathcal{S}^\star = \arg\max_{\mathcal{S}\subseteq\mathcal{M}} \mathcal{F}(\mathcal{S}), \text{ s.t. } |\mathcal{S}| \leq K. \tag{4}$$

The maximization problem of $\mathcal{F}(\mathcal{S})$ under cardinality constraint $|\mathcal{S}| \leq K$ has an approximation solution with $1 - e^{-1}$ bound can be achieved via the greedy algorithm [28, 44]. To start with, initialize $\mathcal{S}$ as an empty set $\emptyset$ and for each greedy iteration, merge an element $\mathcal{T} \in \mathcal{S}_\mathcal{M}^C$ that maximizes the marginal utility $\mathcal{F}(\mathcal{T}|\mathcal{S}) = \mathcal{F}(\mathcal{S}\cup\mathcal{T}) - \mathcal{F}(\mathcal{S})$. The update step for $\mathcal{S}$ can be described as $\mathcal{S} = \mathcal{S}\cup \arg\max_{\mathcal{T}_j\in\mathcal{S}_\mathcal{M}^C} \mathcal{F}(\mathcal{T}_j \mid \mathcal{S})$. The computational complexity of the entire greedy algorithm can be reduced to $\mathcal{O}(|\mathcal{M}|)$ using stochastic methods to choose random subset [24]. This step is correspinds to line 6 in **Algorithm 1**.

4.3. DERTS with Noisy Tasks

Making the meta-learning model robust to label noise is an essential issue, as discussed in previous work [18, 23, 46].

Previous work [13, 26] claims that subset selection based on the gradient is robust to label noise in the standard noisy label setting. Specifically, the Jacobian matrix of a neural network can be well approximated by a low-rank matrix, suggesting a further claim that error for clean labels mainly lies in the subspace corresponding to the dominant singular values while the error (corresponding to noisy labels) lies in the complementary subspace [26]. Thus, the clean data potentially form clusters and aggregate with each other, while noisy labeled data spread out in the gradient space.

We propose a heuristic based on intuition: If a task has noisy label data, some of the labels are incorrect, making the task more difficult to learn. As a result, the model may find it more difficult to find the correct parameters to minimize the loss, resulting in a larger gradient norm. On the other hand, if a task contains clean data, the labels are correct, and the model should be able to learn the task more efficiently, resulting in a smaller gradient norm. Based on the above motivation and the principle of not affecting the computation cost significantly, we dynamically set a threshold h on the gradient norm to implicitly infer the task noise ratio by truncating tasks with a gradient norm higher than the set threshold. Suppose the truncated tasks set is $\mathcal{Z} := \{\mathcal{T}_l | \tilde{g}_l \geq h\}$, we take the complement of $\mathcal{Z}$ on $\mathcal{S}$ as the finalized subset for noisy task scenario. The algorithmic details can be found in **Algorithm 1**.

4.4. Analysis for DERTS

We make three standard assumptions regarding the loss function $\mathcal{L}$:

Assumption 1. *Every loss function $\mathcal{L}(f; \mathcal{D})$ is twice continuously differentiable with respect to f, and β-smooth, i.e.,*

$$\|\nabla\mathcal{L}(f; \mathcal{D}) - \nabla\mathcal{L}(g; \mathcal{D})\| \leq \beta\|f - g\|.$$

Assumption 2 ([11])**.** *The loss function $\mathcal{L}(f; \mathcal{D})$ is $\mu - PL^\star$ (local PL condition), i.e.*

$$\frac{1}{2}\left\|\sum_{i\in\mathcal{M}} \nabla\mathcal{L}(f_i; \mathcal{D}_i)\right\|^2 \geq \mu \sum_{i\in\mathcal{M}} \mathcal{L}(f_i; \mathcal{D}_i), \quad \forall\mathcal{T}_i \in \mathcal{M}.$$

Assumption 3. *The Hessian matrix of the loss function $\mathcal{L}(f; D)$ is bounded, i.e.*

$$m \leq \|\nabla^2\mathcal{L}(f; \mathcal{D})\| \leq M.$$

Assumption 1 and **2** are common in the analysis of convergence of training dynamics [20, 31]. Due to the bilevel-optimization inherent to meta-learning, **Assumption 3** is required – a similar assumption is used in [5]. Our main analysis result concerning the error between loss function trained on the full task pools and the subsets is given below.

Algorithm 1 Efficient and Robust Task Selection (DERTS)

Require: Initialized meta-model f_{θ_0}; task distribution $p(\mathcal{T})$; outer loop and inner loop learning rate η_1, η_2; number of selected tasks K; task pool $\mathcal{M}$; noisy setting flag F; noisy setting threshold h;

Initialize the meta-model f_{θ_0}, task pool $\mathcal{M} \leftarrow \emptyset$, and task subset $\mathcal{S} \leftarrow \emptyset$
for Initialized Task Pool $\mathcal{M}$: **do**
  Randomly draw tasks $\mathcal{T}_i \sim p(\mathcal{T})$ into task pool $\mathcal{M}$
  Estimate outer loop query gradient $\tilde{g}_i$ for every task $\mathcal{T}_i$ in the task pool $\mathcal{M}$
  while $|\mathcal{S}| < K$ **do**
    $\mathcal{T}_i \in \arg\max_{\mathcal{T}_i \in \mathcal{S}_\mathcal{M}^C} \mathcal{F}(\mathcal{T}_i \mid \mathcal{S})$
    $\mathcal{S} = \mathcal{S} \cup \{\mathcal{T}_i\}$
  end while
  $\gamma_i = \sum_{j\in\mathcal{M}} 1_{\{j=\text{argmin}_{\mathcal{T}_i\in\mathcal{S}}\|\tilde{g}_j - \tilde{g}_i\|\}}$
  if noisy setting flag F **then**
    Initialize drop task set $\mathcal{Z} \leftarrow \emptyset$
    for $\mathcal{T}_j \in \mathcal{S}$ **do**
      if $\tilde{g}_j \geq h$ **then**
        $\mathcal{Z} = \mathcal{Z} \cup \{\mathcal{T}_j\}$
      end if
    end for
    $\mathcal{S} = \mathcal{Z}_\mathcal{S}^C$
  end if
  `meta-learn`$(\mathcal{S}, \gamma_1, \ldots, \gamma_{|K|})$ Run standard meta-learning algorithm on $\mathcal{S}$ with weights $\{\gamma_1, \cdots, \gamma_{|K|}\}$ multiply the corresponding computed outer loop gradient of each task.
end for

Theorem 1 (Training Dynamics)**.** *Assume that the loss function $\mathcal{L}(f, \mathcal{D})$ satisfies assumptions 1 –3 and ϵ is an upper bound for the RHS of Eq.*(2)*. Then, with the proper constant learning rate η and η' for outer and inner loop updates and a initialization point θ^0, applying DERTS has similar training dynamics to that of training on the full task pool $\mathcal{M}$. Specifically,*

$$\begin{aligned}\mathbb{E}[(\mathcal{L}(f_{\theta^t}; \mathcal{D}))] \leq (1 - \eta\eta' m\mu)^t \mathbb{E}[(\mathcal{L}(f_{\theta^0}; \mathcal{D}))] \\ + \frac{1}{\eta\eta' m\mu}(\eta\epsilon\|\mathbb{E}[\nabla_\theta\mathcal{L}(f_\phi; \mathcal{D})]\|_{L^\infty_\mathcal{M}} - \frac{\eta}{2}\epsilon^2) + r\end{aligned} \tag{5}$$

where

$$\|\mathbb{E}[\nabla_\theta\mathcal{L}(f_\phi; \mathcal{D})]\|_{L^\infty_\mathcal{M}} = \sup_\Gamma \left\|\mathbb{E}\left[\sum_{j\in\mathcal{M}} \left(\nabla_\theta\mathcal{L}\left(f_{\phi_j}; \mathcal{D}_j^q\right) - \nabla_\theta\mathcal{L}\left(f_{\phi_{\Gamma(j)}}; \mathcal{D}_{\Gamma(j)}^q\right)\right)\right]\right\|,$$

and

$$r = |\mathcal{L}(f_{\phi*}, \mathcal{D}) - \mathcal{L}(f_{\theta^0}, \mathcal{D})|.$$

This result implies training dynamics between subset learning and task pool learning is bounded. The first term on

the RHS of (5) is a contraction that decreases with each iteration. The second term shows the difference between training on subsets $\mathcal{S}$ and all the tasks. The last term r is a bias resulting from the bilevel optimization for meta-learning. We note that the bias is not an artefact of the DERTS algorithm but is inherent in the meta-learning formulation. We show the proof in **Appendix C**.

5. Experiment Results

Here we demonstrate the effectiveness of DERTS by performing extensive experiments on widely used image classification benchmarks in two settings: (i) limited data budget and (ii) noisy label tasks.

Dataset and Baseline: We conduct experiments on the standard image classification benchmarks: **Mini-ImageNet**[41] and **Tiered-ImageNet**[34]. The tasks are constructed as 5-way 1-shot and 5-way 5-shot for limited data budget setting and 5-way 5-shot for noisy task setting. To show the generality of DERTS, we implemented the proposed algorithm on both gradient-based meta-learning method ANIL [33] and metric-based meta-learning method ProtoNet (PN) [36]. For sampling strategies, we compare DERTS with the state-of-the-art sampling paradigm for meta-learning Uniform Episodic Sampling (US) [2] and Adaptive Task Scheduler (ATS) [46]. According to the sampling mechanism, we combine US with both ANIL and PN and ATS for ANIL (ATS is designed only for gradient-based meta-learning algorithms).

Implementation Details: For both ANIL and PN, We use a 4-layer Convolutional Neural Network (CNN4) as the model backbone. Additional experiments with ResNet-12 [30] could be found in **Appendix E**. We set the meta-batch size to 32. Adam [15] is selected as the optimizer. We set the learning rates for ANIL as $0.005 - 0.01$ for outer loops and 0.5 for inner loops with 3 adaptation steps. For PN, we set the learning rate as 0.005. We fix the episodic task pools with the size of 3200, and the number of selected tasks for each pool is 960 (meta-training on each selected subset would take 30 iterations based on a batch size of 32). All experiments are conducted on NVIDIA A6000 GPU.

5.1. Meta-Learning with Limited Budget

Experiment Setup: In this setting, we consider the following questions: (i) Assuming no modification on task generation, can DERTS achieve comparable performance with fewer tasks? (ii) Under the scenario where we dramatically reduce the number of meta-training classes (i.e., 64 classes reduced to 16 classes for the training set of Mini-ImageNet), we then ask how DERTS performs. We train the meta-model for 10000 iterations for 5-way 5-shot setting and 5000 iterations for 5-way 1-shot setting. We set a 500 iteration warm-up process for ANIL and a 100 iteration warm-up process for PN.

Experiment Results with Fewer Tasks: In **Table 1** we show the average accuracy with 95% confidence interval of training a CNN4 base-model on Mini-ImageNet and Tiered-ImageNet for ANIL and PN after varying number of iterations. We evaluate checkpoints after training 10% of all iterations (1000 for 5-way 5-shot and 500 for 5-way 1-shot) and 30% of all iterations (3000 for 5-way 5-shot and 1500 for 5-way 1-shot). Our key observations are as follows. (i) **DERTS is data-efficient for episodic training**. For all the evaluation after 10% and 30% of all iterations for both 5-way 5-shot and 5-way 1-shot scenarios, DERTS achieves higher performance than US, ATS, and vanilla ANIL and PN with random sampling by average gain 4.52% against US, 2.23% against ATS, and 1.92% against vanilla ANIL and PN on accuracy. 6 out of 8 of the evaluation after all iterations, DERTS outperforms all the baselines, and the rest 2 evaluation is averagely 0.46% behind the best performer. The fact that DERTS's full iteration performance indicates the diversity of selected tasks, preserves the generalization capacity for the meta-model. (ii)**DERTS gains significant speedup against other sampling strategies**. **Table 2** shows the average per iteration running time for ANIL-based algorithm on Mini-ImageNet. ANIL has the fastest running time because no extra module is added to the vanilla algorithm. Although US is the fastest sampling strategy among the three modified methods, from **Table 1** we can tell that US has the lowest convergence and performance on accuracy. As ATS has a significantly heavy computation load, the running time of ATS is above 4 times of DERTS's. The execution time of DERTS is slightly higher than vanilla ANIL and US, suggesting that DERTS's estimation of task gradient and submodular maximization with the stochastic greedy algorithm are efficient. Combining the results from **Table 1** and **Table 2**, we claim that DERTS gains general speedup on computation time against baselines for reaching a comparable performance.

Experiment Results with Fewer Class for Generating Tasks: We conduct experiments on the setting of a limited budget of training classes proposed by ATS [46]. In the few-shot classification problem, each training episode is a few-shot task that subsamples classes as well as data points. In mini-Imagenet, the original number of meta-training classes is 64, corresponding to more than 7 million 5-way combinations. Thus, we control the budgets by reducing the number of meta-training classes to 16, resulting in 4,368 combinations. Thus, in this new setting, not only did the data for constructing tasks decrease to 25% in the original setting, but also the number of classes decreased to 25% of the original setting, imposing a significant decrease in task diversity.

Dataset	Mini-ImageNet (5-way 5-shot)			Mini-ImageNet (5-way 1-shot)		
Iterations	1000	3000	10000 (All)	500	1500	5000 (All)
ANIL	56.50 ± 0.8	61.39 ± 0.6	62.83 ± 0.6	41.04 ± 0.8	46.06 ± 0.7	47.22 ± 0.7
ANIL-US	54.32 ± 0.8	58.92 ± 0.7	62.04 ± 0.6	37.01 ± 0.8	43.58 ± 0.8	45.94 ± 0.8
ANIL-ATS	52.97 ± 0.9	60.74 ± 0.7	61.82 ± 0.8	42.38 ± 0.7	46.62 ± 0.6	46.74 ± 0.7
ANIL-DERTS	$\mathbf{57.67 \pm 0.6}$	$\mathbf{61.78 \pm 0.8}$	$\mathbf{63.07 \pm 0.7}$	$\mathbf{44.06 \pm 0.8}$	$\mathbf{47.07 \pm 0.6}$	$\mathbf{47.67 \pm 0.7}$
ProtoNet(PN)	60.19 ± 0.8	62.41 ± 0.7	$\mathbf{66.29 \pm 0.8}$	41.97 ± 0.8	43.97 ± 0.6	48.32 ± 0.7
PN-US	57.21 ± 1.0	61.79 ± 0.9	64.50 ± 0.9	40.90 ± 0.6	44.00 ± 0.8	46.50 ± 0.8
PN-DERTS	$\mathbf{62.16 \pm 0.6}$	$\mathbf{65.09 \pm 0.7}$	65.51 ± 0.7	$\mathbf{46.18 \pm 0.6}$	$\mathbf{47.08 \pm 0.8}$	$\mathbf{48.52 \pm 0.6}$
Dataset	Tiered-ImageNet (5-way 5-shot)			Tiered-ImageNet (5-way 1-shot)		
Iterations	1000	3000	10000 (All)	500	1500	5000 (All)
ANIL	56.92 ± 0.8	62.48 ± 0.6	63.70 ± 0.7	40.59 ± 0.6	46.03 ± 0.7	48.28 ± 0.7
ANIL-US	56.06 ± 1.0	58.43 ± 0.8	62.20 ± 0.8	36.87 ± 0.8	43.08 ± 0.9	46.15 ± 0.8
ANIL-ATS	53.65 ± 0.9	62.92 ± 0.8	62.34 ± 0.7	42.74 ± 0.7	46.55 ± 0.7	47.81 ± 0.8
ANIL-DERTS	$\mathbf{60.49 \pm 0.7}$	$\mathbf{63.07 \pm 0.8}$	$\mathbf{64.01 \pm 0.8}$	$\mathbf{44.47 \pm 0.8}$	$\mathbf{47.78 \pm 0.8}$	$\mathbf{48.40 \pm 0.8}$
ProtoNet(PN)	64.28 ± 0.7	65.22 ± 0.8	$\mathbf{69.26 \pm 0.8}$	43.52 ± 0.7	47.07 ± 0.7	$\mathbf{49.56 \pm 0.7}$
PN-US	60.67 ± 0.8	64.05 ± 0.8	67.89 ± 0.8	40.43 ± 0.7	42.73 ± 0.8	49.02 ± 0.8
PN-DERTS	$\mathbf{65.02 \pm 0.7}$	$\mathbf{66.03 \pm 0.8}$	69.11 ± 0.7	$\mathbf{44.19 \pm 0.7}$	$\mathbf{47.96 \pm 0.7}$	49.48 ± 0.7

Table 1. Average accuracy (%) of 5-way 5-shot and 5-way 1-shot Mini-ImageNet and Tiered-ImageNet Classification with Limited Budget Setting. In the 5-way 5-shot setting, iterations of 1000 (3000) in the table denote the performance after learning on 10% (30%) tasks during the episodic training. In the 5-way 1-shot setting, iterations of 500 (1500) in the table denote the performance after learning on 10% (30%) tasks during the episodic training.

Time	ANIL	ANIL-US	ANIL-ATS	**ANIL-DERTS**
[s]	0.60	0.69	3.11	0.95

Table 2. Average single iteration run time for limited budget setting.

Dataset	Mini-ImageNet (16-Class)	
Method	5-way 5-shot	5-way 1-shot
ANIL	45.97 ± 0.65	33.61 ± 0.66
ANIL-US	46.70 ± 0.70	34.70 ± 0.50
ANIL-ATS	47.76 ± 0.68	35.15 ± 0.67
ANIL-DERTS	$\mathbf{48.92 \pm 0.54}$	$\mathbf{36.85 \pm 0.75}$
ProtoNet	50.27 ± 0.67	36.46 ± 0.67
ProtoNet-US	49.21 ± 0.77	35.48 ± 0.70
ProtoNet-DERTS	$\mathbf{52.08 \pm 0.71}$	$\mathbf{37.59 \pm 0.62}$

Table 3. 5-way 5-shot / 1-shot Mini-ImageNet Classification with 25% Class Training Set.

Table 3 shows the result of the aforementioned setting with fewer classes. By only training on 25% of the classes, we observe that DERTS outperforms all the baselines with an average of 2% on accuracy. Due to the significant decrease in task diversity, we claim our task selection method still correctly captures the decreased diversity and focuses on the more informative tasks for learning representation. Additional experiments with other ratios of decreased meta-training class can be found in **Appendix E**.

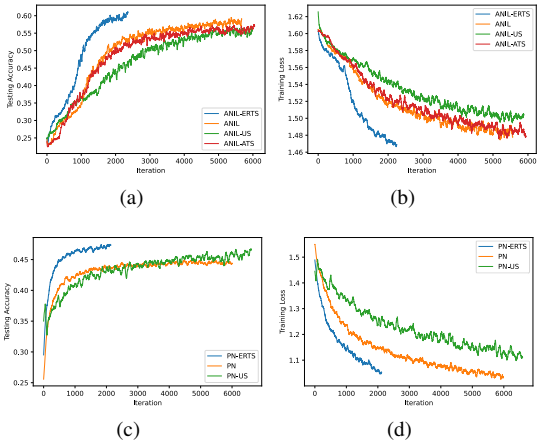
5.2. Meta-Learning with Noisy Tasks

Experiment Setup: Unlike previous work [18, 46] which only constructs noisy data within the support set, we extend the noisy data to both the support set and query set, without providing the noisy data's specific identity. The details of generating noisy tasks can be found in **Appendix D**. We consider two specific setups: the first where 25% of data points are incorrectly labeled, the second where 40% are mislabelled (25% setting means the average noise ratio for all the tasks is 25%, but there could be extremely high noise ratio tasks and roughly clean-label tasks). We test 5-way 5-shot setting with symmetric label flipping [40]. For the 25% case, we use the same warm-up model as with the limited budget setting. For the 40% case, we double the number of warm-up iterations for ANIL and PN. We set the estimated gradient norm threshold h as 1.25 times the average of the task estimated gradient norm in the task pools.

Experiment Result: **Table 4** shows the average accuracy with a 95% confidence interval of training a CNN4 base-model on Mini-ImageNet and Tiered-ImageNet for ANIL and ProtoNet with 12000 iterations. DERTS outperforms all baselines in the noisy task settings, by achieving on average 3% improvement in testing accuracy. It is worth noting that in the 40% settings, DERTS acquires a larger gain on performance than in 25% noisy setting, which empirically demonstrates the robustness of our algorithm. Another important observation is that ATS and US are empirically sensitive to the noisy task setting. The reason ATS is less robust than

Dataset	Mini-ImageNet (5-way 5-shot)		Tiered-ImageNet (5-way 5-shot)	
Noise Ratio	25%	40%	25%	40%
ANIL	61.12 ± 0.85	57.32 ± 0.79	60.84 ± 0.65	57.31 ± 0.78
ANIL-US	58.48 ± 0.69	52.39 ± 0.90	57.13 ± 0.75	51.11 ± 1.21
ANIL-ATS	59.46 ± 0.72	54.42 ± 0.73	58.35 ± 0.79	54.73 ± 0.72
ANIL-DERTS	$\mathbf{62.38 \pm 0.66}$	$\mathbf{59.14 \pm 0.72}$	$\mathbf{62.40 \pm 0.70}$	$\mathbf{58.23 \pm 0.81}$
ProtoNet	58.68 ± 0.85	49.85 ± 0.85	59.34 ± 0.79	50.02 ± 0.91
ProtoNet-US	56.09 ± 0.82	46.96 ± 0.80	55.16 ± 0.76	45.99 ± 0.98
ProtoNet-DERTS	$\mathbf{60.98 \pm 0.84}$	$\mathbf{51.40 \pm 0.78}$	$\mathbf{60.77 \pm 0.81}$	$\mathbf{52.58 \pm 0.79}$

Table 4. 5-way 5-shot Mini-ImageNet and Tiered-ImageNet Classification with Noisy Tasks. 25% and 40% denote the noise ratio (percentage of mislabeled data).



(a) (b) (c) (d)

Figure 2. Loss Residual and Accuracy for Noisy Task Settings (Early Stage). (a) Test Accuracy of 25% Noise Setting on Mini-ImageNet of ANIL. (b) Training Loss of 25% Noise Setting on Mini-ImageNet of ANIL. (c) Test Accuracy of 40% Noise Setting on Mini-ImageNet of PN. (d) Training Loss of 40% Noise Setting on Mini-ImageNet of PN.

Selecting Ratio k/N	Accuracy
10%	61.70 ± 0.72
20%	62.07 ± 0.77
30% (Default)	$\mathbf{63.07 \pm 0.71}$
30% (No Weights)	59.55 ± 0.72
50%	62.98 ± 0.70

Table 5. Ablation Study on Selection Ratio and Weights with Mini-ImageNet 5-way 5-shot Classification

vanilla ANIL and ANIL-DERTS may be caused by the inner product computation for gradient on support set and query set is ineffective because both support sets and query sets contain the noisy labeled data. **Figure 2** shows the training dynamics of DERTS and baselines. We observe that aside from the final gain in performance, DERTS converges faster than all baselines as well.

5.3. Ablation Study on DERTS

We investigate the effectiveness of selecting ratio k/N (k is the size of subsets and N is the size of task pools) and weights $\{\gamma_i \mid i = 1, 2, \ldots, k\}$ in the ablation study.

Table 5 shows the results of different selecting ratio k/N and the default ratio without adding weights. We observe that with a low selecting ratio, DERTS's performance drops 1%-2%. This phenomenon may caused by the decrease of selected tasks' diversity since the selected ratio k/N is low and the subset can not include all the informative tasks from the task pool. The performance of setting without adding weights is significantly lower than the setting with corresponding weights, showing the selected tasks are not equally important as well. This phenomenon suggests DERTS's weight selection is an essential element for capturing the full training dynamics. Additional ablation study can be found in **Appendix E**.

6. Conclusion & Discussion

We propose Data-Efficient and Robust Task Selection (DERTS), a task selection algorithm for meta-learning from the view of optimization to address the issue of data efficiency, and robustness against noisy labels. By selecting subsets with weights to approximate the gradient of the tasks from the task pools, DERTS is data-efficient for episodic training under the limited budget scenario. We then modify DERTS to address the robustness of noisy data scenarios. Our extensive experiments demonstrate that DERTS outperforms the state-of-the-art sampling and selection strategies in terms of both data efficiency and robustness.

References

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

[2] Sébastien Arnold, Guneet Dhillon, Avinash Ravichandran, and Stefano Soatto. Uniform sampling over episode difficulty. *Advances in Neural Information Processing Systems*, 34:1481–1493, 2021.

[3] Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. Diverse client selection for federated learning via submodular maximization. In *International Conference on Learning Representations*, 2022.

[4] Gerard Cornuejols, Marshall Fisher, and George L Nemhauser. On the uncapacitated location problem. In *Annals of Discrete Mathematics*, pages 163–177. Elsevier, 1977.

[5] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1082–1092. PMLR, 2020.

[6] 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.

[7] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[8] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5138–5147, 2019.

[9] 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.

[10] Jean Kaddour, Steindór Sæmundsson, et al. Probabilistic active meta-learning. *Advances in Neural Information Processing Systems*, 33:20813–20822, 2020.

[11] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pages 795–811. Springer, 2016.

[12] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

[13] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8110–8118, 2021.

[14] Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:14488–14501, 2021.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Xiaomeng Li, Lequan Yu, Yueming Jin, Chi-Wing Fu, Lei Xing, and Pheng-Ann Heng. Difficulty-aware meta-learning for rare disease diagnosis. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*, pages 357–366. Springer, 2020.

[17] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1910–1918, 2017.

[18] Kevin J Liang, Samrudhdhi B Rangrej, Vladan Petrovic, and Tal Hassner. Few-shot learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9089–9098, 2022.

[19] Chenghao Liu, Zhihao Wang, Doyen Sahoo, Yuan Fang, Kun Zhang, and Steven CH Hoi. Adaptive task sampling for meta-learning. In *European Conference on Computer Vision*, pages 752–769. Springer, 2020.

[20] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

[21] Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

[22] Ricardo Luna Gutierrez and Matteo Leonetti. Information-theoretic task selection for meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 33:20532–20542, 2020.

[23] Pratik Mazumder, Pravendra Singh, and Vinay P Namboodiri. Rnnp: A robust few-shot learning approach. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2664–2673, 2021.

[24] Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

[25] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.

[26] Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of deep neural networks against noisy labels. *Advances in Neural Information Processing Systems*, 33:11465–11477, 2020.

[27] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

[28] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1): 265–294, 1978.

[29] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[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.

[31] Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In *International Conference on Machine Learning*, pages 4951–4960. PMLR, 2019.

[32] Omead Pooladzandi, David Davini, and Baharan Mirzasoleiman. Adaptive second order coresets for data-efficient machine learning. In *International Conference on Machine Learning*, pages 17848–17869. PMLR, 2022.

[33] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*.

[34] 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.

[35] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

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

[37] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[38] Yue Sun, Adhyyan Narang, Ibrahim Gulluk, Samet Oymak, and Maryam Fazel. Towards sample-efficient overparameterized meta-learning. *Advances in Neural Information Processing Systems*, 34:28156–28168, 2021.

[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.

[40] Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. *Advances in neural information processing systems*, 28, 2015.

[41] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[42] Zhenyi Wang, Li Shen, Tiehang Duan, Donglin Zhan, Le Fang, and Mingchen Gao. Learning to learn and remember super long multi-domain task sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7982–7992, 2022.

[43] Zhenyi Wang, Li Shen, Le Fang, Qiuling Suo, Donglin Zhan, Tiehang Duan, and Mingchen Gao. Meta-learning with less forgetting on large-scale non-stationary task distributions. In *European Conference on Computer Vision*, pages 221–238. Springer, 2022.

[44] Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4): 385–393, 1982.

[45] Yu Yang, Hao Kang, and Baharan Mirzasoleiman. Towards sustainable learning: Coresets for data-efficient deep learning. *arXiv preprint arXiv:2306.01244*, 2023.

[46] Huaxiu Yao, Yu Wang, Ying Wei, Peilin Zhao, Mehrdad Mahdavi, Defu Lian, and Chelsea Finn. Meta-learning with an adaptive task scheduler. *Advances in Neural Information Processing Systems*, 34:7497–7509, 2021.

[47] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020.

[48] Runsheng Yu, Weiyu Chen, Xinrun Wang, and James Kwok. Enhancing meta learning via multi-objective soft improvement functions. In *The Eleventh International Conference on Learning Representations*, 2023.