

# Memory-Reduced Meta-Learning with Guaranteed Convergence

Honglin Yang<sup>1,2</sup>, Ji Ma<sup>1,2\*</sup>, Xiao Yu<sup>2,3</sup>

<sup>1</sup>Department of Automation, Xiamen University, Xiamen, China

<sup>2</sup>Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen, China

<sup>3</sup>Institute of Artificial Intelligence, Xiamen University, Xiamen, China  
yanghonglin@stu.xmu.edu.cn, maji08@xmu.edu.cn, xiaoyu@xmu.edu.cn

## Abstract

The optimization-based meta-learning approach is gaining increased traction because of its unique ability to quickly adapt to a new task using only small amounts of data. However, existing optimization-based meta-learning approaches, such as MAML, ANIL and their variants, generally employ back-propagation for upper-level gradient estimation, which requires using historical lower-level parameters/gradients and thus increases computational and memory overhead in each iteration. In this paper, we propose a meta-learning algorithm that can avoid using historical parameters/gradients and significantly reduce memory costs in each iteration compared to existing optimization-based meta-learning approaches. In addition to memory reduction, we prove that our proposed algorithm converges sublinearly with the iteration number of upper-level optimization, and the convergence error decays sublinearly with the batch size of sampled tasks. In the specific case in terms of deterministic meta-learning, we also prove that our proposed algorithm converges to an exact solution. Moreover, we quantify that the computational complexity of the algorithm is on the order of  $\mathcal{O}(\epsilon^{-1})$ , which matches existing convergence results on meta-learning even without using any historical parameters/gradients. Experimental results on meta-learning benchmarks confirm the efficacy of our proposed algorithm.

**Code** — <https://github.com/yangxiaolin9527/RM-MetaL>

**Extended version** —

<https://doi.org/10.48550/arXiv.2412.12030>

## 1 Introduction

Meta-learning has emerged as a promising paradigm in modern machine learning. Unlike traditional machine learning methods that require large amounts of data points for model training (He et al. 2016; Vaswani et al. 2017), meta-learning leverages prior experience from previously learned tasks to quickly adapt to new tasks with limited data. This feature makes meta-learning particularly appealing for applications where data are scarce or expensive to obtain (Denevi et al. 2019; Wang et al. 2020). To date, numerous meta-learning methods have been proposed, such as

metric-based approaches (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017), optimization-based approaches (Finn, Abbeel, and Levine 2017; Raghu et al. 2019), and model-based approaches (Santoro et al. 2016; Li et al. 2019). Recently, with the rapid progress of optimization theory in machine learning, optimization-based meta-learning approaches are gaining increased traction (Bertinetto et al. 2018; Rajeswaran et al. 2019; Lee, Yoo, and Kwak 2023; Wu et al. 2023; Chen and Wang 2024a).

Optimization-based approaches often formulate meta-learning as a bilevel programming problem, where the lower-level optimization is dedicated to adapting the model to a given task, and the upper-level optimization aims to train meta-parameters that can enable the model to generalize to new tasks (Maclaurin, Duvenaud, and Adams 2015; Zhang et al. 2024). Over the past decades, numerous optimization-based meta-learning approaches have been proposed, with typical examples including Model-Agnostic Meta-Learning (MAML) (Finn, Abbeel, and Levine 2017) and its variants (Finn et al. 2017; Zhou, Wu, and Li 2018; Nichol, Achiam, and Schulman 2018; Raghu et al. 2019; Oh et al. 2021; Chi et al. 2022; Kirsch et al. 2022), and bilevel-optimization-based approaches (Franceschi et al. 2018; Bertinetto et al. 2018; Zhou et al. 2019; Ji, Yang, and Liang 2020). However, traditional optimization-based meta-learning approaches employ iterative differentiation to estimate upper-level gradients, i.e., hypergradient estimation, which requires the use of historical lower-level parameters/gradients, leading to increased computational and memory overhead, as evidenced by our experimental results in Figure 2 and Table 2. Although recent approaches, such as FOMAML (Finn, Abbeel, and Levine 2017) and others (Nichol, Achiam, and Schulman 2018; Zhou et al. 2019; Fallah, Mokhtari, and Ozdaglar 2020), mitigate the memory costs in each iteration by avoiding the use of historical lower-level parameters/gradients, they suffer from decreased learning accuracy due to the lack of information from these lower-level parameters/gradients.

### 1.1 Contributions

Our contributions can be summarized as follows.

1. We propose an optimization-based meta-learning algorithm that eliminates the need for using any historical parameters/gradients, and thus ensures nearly invariant memory

\*Corresponding author.

costs over iterations. This is fundamentally different from traditional MAML-based and iterative-differentiation-based meta-learning approaches such as Finn, Abbeel, and Levine (2017); Raghu et al. (2019); Oh et al. (2021), where memory costs grow to infinity when the iteration number of lower-level optimization tends to infinity. In fact, our experimental results in Figure 2 show that our algorithm achieves at least a 50% reduction in memory costs compared to ANIL in Raghu et al. (2019) and the iterative-differentiation-based approach in Ji, Yang, and Liang (2020).

2. In addition to reducing memory costs, we also establish the convergence rate and computational complexity of our algorithm for both stochastic and deterministic meta-learning, which is different from the existing bilevel programming literature, e.g., Franceschi et al. (2018); Shaban et al. (2019); Ji, Yang, and Liang (2020), that focus solely on deterministic cases. For stochastic meta-learning, we prove that our algorithm converges sublinearly with the number of iterations, and the convergence error decays sublinearly with the batch size of sampled tasks, which is consistent with the nonconvex nature of the stochastic meta-learning objective functions. For deterministic meta-learning, we prove that our algorithm can converge sublinearly to an exact solution.

3. We quantify that the computational complexity of our algorithm is on the order of  $\mathcal{O}(\epsilon^{-1})$  for both stochastic and deterministic meta-learning, which matches the currently well-known results on optimization-based meta-learning (Rajeswaran et al. 2019; Ji et al. 2020).

4. Furthermore, since our algorithm estimates the hypergradient using the Hessian-inverse-vector-product without computing the full Hessian or Jacobian matrices, we improve the computational complexity of the hypergradient estimation in previous meta-learning approaches (Park and Oliva 2019; Hiller, Harandi, and Drummond 2022) from  $\mathcal{O}(q^2)$  (or  $\mathcal{O}(pq)$ ) to  $\mathcal{O}(\max\{p, q\})$  per iteration.

5. We conduct experimental evaluations using several meta-learning benchmark datasets, including the ‘‘CIFAR-FS’’, ‘‘FC100’’, ‘‘miniImageNet’’, and ‘‘tieredImageNet’’ datasets. The results confirm the efficiency of our algorithm in terms of both learning accuracy and memory reduction.

## 1.2 Related Work

The earliest optimization-based meta-learning approach is Model-Agnostic Meta-Learning (MAML), which is an initial parameter-transfer method where the objective is to learn a good ‘‘optimal initial model parameters’’ (Finn, Abbeel, and Levine 2017). Despite the widespread use of MAML and its variants such as DEML (Zhou, Wu, and Li 2018), ANIL (Raghu et al. 2019), and BOIL (Oh et al. 2021) in image classification (Chi et al. 2022; Ullah et al. 2022), reinforcement learning (Vuorio et al. 2019; Kirsch et al. 2022), and imitation learning (Finn et al. 2017; Li et al. 2021), these approaches rely on iterative differentiation for hypergradient estimation, which requires using historical lower level parameters/gradients, leading to high computational and memory overhead. To solve this issue, Finn, Abbeel, and Levine (2017) proposed a simplified MAML algorithm that avoids leveraging historical lower-level parameters/gradients and uses only the last parameters in lower-level

optimization for hypergradient updates, thereby reducing computational and memory overhead. However, the memory reduction achieved by this approach comes at the expense of degrading learning accuracy, which is undesirable in accuracy-sensitive meta-learning applications (Zhou et al. 2019; Peng 2020).

Driven by the need of reducing computational and memory costs in optimization-based meta-learning approaches, plenty of bilevel-optimization-based algorithms have recently been proposed (Bertinetto et al. 2018; Ji, Yang, and Liang 2020; Chen and Wang 2024a) (which aim to learn good embedding model parameters, rather than focusing on a good initialization as MAML-based approaches do). Initially, these algorithms addressed meta-learning problems by treating the lower-level optimality condition as constraints to the upper-level problem. However, this method is inapplicable to the scenario where the lower-level problem is large scale or the lower-level objective function has a complex structure. More recently, Rajeswaran et al. (2019); Ji, Yang, and Liang (2020) proposed an approximate-implicit-differentiation-based approach to solve meta-learning problems. However, this approach requires a common optimal solution for lower-level optimization, which is often difficult to satisfy in many meta-learning applications (note that in meta-learning, the lower-level optimization aims to adapt the model to different tasks, leading to heterogeneous optimal solutions for lower-level optimization problems). Moreover, the convergence analysis in Ji, Yang, and Liang (2020) is based on deterministic gradients, and it is not clear if the deterministic convergence analysis methods can be extended to stochastic meta-learning, where the task distribution is unknown in practice.

## 2 Preliminaries

**Notations** We use  $\mathbb{R}$  and  $\mathbb{N}^+$  to represent the sets of real numbers and positive integers, respectively. We abbreviate *with respect to* by *w.r.t.* We denote  $\nabla F(\theta) \in \mathbb{R}^p$  as the gradient of  $F(\theta)$ . We use  $\nabla_{\theta} g(\theta, \phi)$  and  $\nabla_{\phi} g(\theta, \phi)$  to represent the gradients of  $g$  w.r.t.  $\theta$  and  $\phi$ , respectively. We write  $\nabla_{\theta\phi}^2 g(\theta, \phi) \in \mathbb{R}^{p \times q}$  for the Jacobian matrix of  $g$  and  $\nabla_{\phi}^2 g(\theta, \phi) \in \mathbb{R}^{q \times q}$  for the Hessian matrix of  $g$  w.r.t.  $\phi$ . For vectors  $\phi_1, \dots, \phi_n$ , we use  $\phi = \text{col}\{\phi_1, \dots, \phi_n\}$  to represent their stacked column vector. For a set  $\mathcal{B}$ , we denote the number of its elements by  $|\mathcal{B}|$ .

### 2.1 Meta-Learning

We consider a meta-learning problem with a series of tasks  $\{\mathcal{T}_i, i \in \mathbb{N}^+\}$ . All tasks are drawn from an unknown task distribution  $\mathcal{P}(\mathcal{T})$ . Each task  $\mathcal{T}_i$  has a loss function  $\mathcal{L}(\theta, \phi; \xi_i)$  over each data point  $\xi_i$ , where  $\theta$  represents the meta-parameters of an embedding model shared by all tasks, and  $\phi$  represents the task-specific parameters. The goal of meta-learning is to find common optimal parameters  $\theta^*$  that benefit all tasks, and based on the parameters  $\theta^*$ , the model can quickly adapt its own parameters  $\phi$  to a new task  $\mathcal{T}_i$  using only a few data points and training iterations. Typically, the meta-learning problem can be formulated as the follow-

ing bilevel programming problem:

$$\begin{aligned} \min_{\theta \in \mathbb{R}^p} F(\theta) &= \mathbb{E}_{\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})} [f_i(\theta, \phi_i^*(\theta))], \\ \text{s.t. } \phi_i^*(\theta) &= \underset{\phi \in \mathbb{R}^q}{\operatorname{argmin}} g_i(\theta, \phi). \end{aligned} \quad (1)$$

In task  $\mathcal{T}_i$ , the upper-level objective function  $f_i(\theta, \phi)$  is given by  $f_i(\theta, \phi) \triangleq \frac{1}{|\mathcal{D}_{i,f}|} \sum_{j=1}^{|\mathcal{D}_{i,f}|} \mathcal{L}(\theta, \phi; \xi_{i,f}^j)$ , where data  $\xi_{i,f}^j$ ,  $j \in \{1, \dots, |\mathcal{D}_{i,f}|\}$  is sampled from the validation dataset  $\mathcal{D}_{i,f}$ , and the objective function  $g_i(\theta, \phi)$  is given by  $g_i(\theta, \phi) \triangleq \frac{1}{|\mathcal{D}_{i,g}|} \sum_{j=1}^{|\mathcal{D}_{i,g}|} \mathcal{L}(\theta, \phi; \xi_{i,g}^j) + R(\phi)$ , where data  $\xi_{i,g}^j$ ,  $j \in \{1, \dots, |\mathcal{D}_{i,g}|\}$  is sampled from training dataset  $\mathcal{D}_{i,g}$  and  $R(\phi)$  is a strongly-convex regularizer w.r.t.  $\phi$ .

**Remark 1.** Compared with existing bilevel-programming frameworks which use common lower-level optimal parameters  $\phi^*$  to enable upper-level optimization (see, e.g., Ji, Yang, and Liang (2020)), our meta-learning framework in (1) allows different tasks  $\mathcal{T}_i$  to have different lower-level optimal parameters  $\phi_i^*$ . This heterogeneity significantly complicates our convergence analysis.

**Remark 2.** The meta-learning formulation in (1) involves bilevel-programming objectives. This is fundamentally different from conventional single-level machine learning, which focuses on learning optimal parameters  $\phi^*$  for a specific task using a large number of data points (Soydaner 2020; Chen et al. 2024; Chen and Wang 2024b). However, in many applications, collecting data is costly and time consuming, and may even be infeasible due to physical system constraints (Wang et al. 2020). Therefore, meta-learning is typically regarded as a more general framework than conventional single-level machine learning (Shu, Meng, and Xu 2021). In the special case where the upper-level objective function is absent, the meta-learning problem can be reduced to a single-level machine learning problem.

The objective functions in (1) satisfy the following standard assumptions.

**Assumption 1.**  $F(\theta)$  is nonconvex and  $g_i(\theta, \phi)$  is  $\mu$ -strongly convex w.r.t.  $\phi$ .

**Assumption 2.** We denote  $f(\theta, \phi) \triangleq \mathbb{E}_{\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})} [f_i(\theta, \phi)]$  and  $g(\theta, \phi) \triangleq \mathbb{E}_{\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})} [g_i(\theta, \phi)]$ . Then, the following statements are satisfied: (i) Functions  $f$ ,  $\nabla f$ ,  $\nabla g$ ,  $\nabla_{\theta\phi}^2 g$ , and  $\nabla_{\phi\phi}^2 g$  are  $l_{f,0}$ ,  $l_f$ ,  $l_g$ ,  $l_{g,1}$ , and  $l_{g,2}$  Lipschitz continuous, respectively. (ii) Gradients  $\nabla f_i(\theta, \phi)$ ,  $\nabla_{\phi}^2 g_i(\theta, \phi)$  and  $\nabla_{\theta\phi}^2 g_i(\theta, \phi)$  are unbiased and have bounded variances  $\sigma_{f,1}^2$ ,  $\sigma_{g,1}^2$ , and  $\sigma_{g,2}^2$ , respectively.

Assumptions 1 and 2 are standard in meta-learning literature; see, for example Ji et al. (2020). Note that we do not impose that the lower-level function  $g_i$  is Lipschitz w.r.t.  $\phi$ , which is required in iMAML in Rajeswaran et al. (2019). Note that iMAML implicitly assumes the search space of parameters  $\phi_i$  to be bounded such that function  $g_i$  is bounded. Moreover, since each task  $\mathcal{T}_i$ ,  $i \in \{1, 2, \dots, |\mathcal{B}|\}$  is randomly sampled from the task distribution  $\mathcal{P}(\mathcal{T})$ , gradients  $\nabla f_i(\theta, \phi)$  and  $\nabla g_i(\theta, \phi)$  are stochastic, and thus we assume that they satisfy Assumption 2-(ii).

In practice, the task distribution  $\mathcal{P}(\mathcal{T})$  is usually unknown, making it impossible to directly solve problem (1). To circumvent this problem, a common approach is to reformulate (1) as the following empirical risk minimization problem:

$$\begin{aligned} \min_{\theta \in \mathbb{R}^p} F_{\mathcal{B}}(\theta) &= f_{\mathcal{B}}(\theta, \phi^*(\theta)) = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} f_i(\theta, \phi_i^*(\theta)), \\ \text{s.t. } \phi_i^*(\theta) &= \underset{\phi \in \mathbb{R}^q}{\operatorname{argmin}} g_i(\theta, \phi), \end{aligned} \quad (2)$$

where  $\phi^*(\theta)$  represents a stacked vector of task-specific optimal parameters  $\phi_i^*$  and  $|\mathcal{B}|$  denotes the batch size of sampled tasks.

## 2.2 Main Challenges in Meta-Learning

The main challenge in solving problem (2) lies in computing the hypergradient  $\nabla F_{\mathcal{B}}(\theta)$  of the upper-level function w.r.t.  $\theta$ . From (2), the hypergradient computation requires knowledge of  $\phi^*(\theta)$ . However, obtaining  $\phi^*(\theta)$  is often difficult in large-scale meta-learning, especially when the lower-level objective function  $g_i(\theta, \phi)$  has a complex structure.

To solve this issue, a commonly used approach is to employ iterative differentiation for hypergradient estimation; see, for example, Domke (2012); Maclaurin, Duvenaud, and Adams (2015); Franceschi et al. (2017). More specifically, the iterative-differentiation-based approach first executes a  $K$ -step gradient descent to solve the lower-level optimization problem in (2) and obtain an approximate solution  $\phi_i^K(\theta)$  for each task  $\mathcal{T}_i$ . Then, building on the solution  $\phi_i^K(\theta)$ , this approach estimates the hypergradient by using the following relation (Ji, Yang, and Liang 2020):

$$\begin{aligned} \frac{\partial f_{\mathcal{B}}(\theta, \phi^K(\theta))}{\partial \theta} &= \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \frac{\partial f_i(\theta, \phi_i^K(\theta))}{\partial \theta} \\ &= \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \left( \nabla_{\theta} f_i(\theta, \phi_i^K(\theta)) - \lambda_{\phi} \sum_{k=0}^{K-1} \nabla_{\theta\phi}^2 g_i(\theta, \phi_i^k(\theta)) \times \right. \\ &\quad \left. \prod_{q=k+1}^{K-1} (I - \lambda_{\phi} \nabla_{\phi}^2 g_i(\theta, \phi_i^q(\theta))) \nabla_{\phi} f_i(\theta, \phi_i^K(\theta)) \right). \end{aligned} \quad (3)$$

It is clear that the hypergradient estimation in (3) relies on historical lower-level parameters  $\phi_i^k(\theta)$ ,  $k = 0, \dots, K$ . As the iteration number of lower-level optimization grows, this dependency significantly increases both computational and memory overhead, as evidenced by our experimental results in Figure 2 and Table 2. Moreover, obtaining a good approximation for  $\phi_i^*(\theta)$  often requires numerous iterations in lower-level optimization, which inevitably extends the back-propagation chain and results in gradient problems such as vanishing or exploding hypergradient estimation (Ji et al. 2020; Jamal, Wang, and Gong 2021).

Motivated by these observations, we aim to propose a meta-learning algorithm that can avoid using historical lower-level parameters/gradients for hypergradient estimation while still ensuring comparable convergence compared with existing optimization-based meta-learning approaches.

### 3 Methodology

In this section, we propose a memory-reduced algorithm for meta-learning with provable convergence to a solution  $\theta^*$  to problem (2). Before presenting our algorithm, we first introduce our hypergradient-estimation approach.

#### 3.1 Hypergradient Estimation

Inspired by the recent results on stochastic bilevel programming (Ghadimi and Wang 2018; Lorraine, Vicol, and Duvenaud 2020), we estimate the hypergradient by using the following relation:

$$\begin{aligned} \nabla F_{\mathcal{B}}(\theta) &= \nabla_{\theta} f_{\mathcal{B}}(\theta, \phi^*(\theta)) - \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} (\nabla_{\theta\phi}^2 g_i(\theta, \phi_i^*(\theta)) \\ &\quad \times [\nabla_{\phi}^2 g_i(\theta, \phi_i^*(\theta))]^{-1} \nabla_{\phi} f_i(\theta, \phi_i^*(\theta))). \end{aligned} \quad (4)$$

From (4), it can be seen that obtaining  $\nabla F_{\mathcal{B}}(\theta)$  requires computing the inverse of Hessian matrix  $[\nabla_{\phi}^2 g_i(\theta, \phi_i^*(\theta))]^{-1}$  and Jacobian matrix  $\nabla_{\theta\phi}^2 g_i(\theta, \phi_i^*(\theta))$ . To avoid computing the full Hessian/Jacobian matrix, we aim to estimate the Hessian-inverse-vector product:

$$v_i^* = [\nabla_{\phi}^2 g_i(\theta, \phi_i^*(\theta))]^{-1} \nabla_{\phi} f_i(\theta, \phi_i^*(\theta)). \quad (5)$$

Based on (4), the hypergradient can be rewritten as

$$\nabla F_{\mathcal{B}}(\theta) = \nabla_{\theta} f_{\mathcal{B}}(\theta, \phi^*(\theta)) - \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \nabla_{\theta\phi}^2 g_i(\theta, \phi_i^*(\theta)) v_i^*, \quad (6)$$

where  $\nabla_{\theta\phi}^2 g_i(\theta, \phi_i^*(\theta)) v_i^*$  will be referred to as the Jacobian-vector product. It follows from (6) that if the estimation of  $\nabla_{\theta} f_{\mathcal{B}}(\theta, \phi^*(\theta))$ ,  $v_i^*$ , and  $\nabla_{\theta\phi}^2 g_i(\theta, \phi_i^*(\theta)) v_i^*$  is accurate enough, a good estimation of the hypergradient is obtained.

Note that estimating the Hessian-inverse-vector product  $v_i^*$  and further Jacobian-vector product  $\nabla_{\theta\phi}^2 g_i(\theta, \phi_i^*(\theta)) v_i^*$  circumvents the requirement on estimating the full Hessian and Jacobian matrices, and hence, reduces the computational complexity from the order of  $\mathcal{O}(q^2)$  (or  $\mathcal{O}(pq)$ ) to the order of  $\mathcal{O}(\max\{p, q\})$  per iteration. This is different from existing results on bilevel programming and meta-learning (Park and Oliva 2019; Chen et al. 2022; Hiller, Harandi, and Drummond 2022), which estimate the hypergradient by computing full matrices directly, leading to heavy computational overhead.

#### 3.2 Memory-Reduced Meta-Learning Algorithm Design

In this section, we first introduce an approach to estimate the Hessian-inverse-vector product  $v_i^*$ , which is necessary for estimating the hypergradient according to (6). Using it as a subroutine, we will then propose our memory-reduced meta-learning algorithm.

Approximating  $v_i^*$  in (5) equals to solving the equation  $\nabla_{\phi}^2 g_i(\theta, \phi_i^*(\theta)) v_i = \nabla_{\phi} f_i(\theta, \phi_i^*(\theta))$ , which is the optimality condition of the following optimization problem:

$$\min_{v \in \mathbb{R}^q} \varphi(v), \quad \varphi(v_i) = \frac{1}{2} v_i^T H_i v_i - b_i^T v_i, \quad (7)$$

---

Subroutine 1: Estimating Hessian-inverse-vector product at the  $t$ -th outer loop iteration

---

- 1: **Input:** Parameters  $\theta_t$  and  $\phi_{i,t}^K$ ; integer  $N$ .
  - 2:  $v_{i,t}^0 = v_{i,t-1}^N$  if  $t > 0$ , and  $v_{i,t}^0 = \mathbf{0}_q$  otherwise.
  - 3:  $r_{i,t}^0 = p_{i,t}^0 = \nabla_{\phi} f_i(\theta_t, \phi_{i,t}^K) - \nabla_{\phi}^2 g_i(\theta_t, \phi_{i,t}^K) v_{i,t}^0$ .
  - 4: **for**  $n = 0, 1, \dots, N - 1$  **do**
  - 5:   Get Hessian-vector product  $h_{i,t}^n = \nabla_{\phi}^2 g_i(\theta_t, \phi_{i,t}^K) p_{i,t}^n$ .
  - 6:    $\eta_{i,t}^n = \frac{r_{i,t}^{n,T} r_{i,t}^n}{p_{i,t}^{n,T} h_{i,t}^n}$ .
  - 7:    $v_{i,t}^{n+1} = v_{i,t}^n + \eta_{i,t}^n p_{i,t}^n$ .
  - 8:    $r_{i,t}^{n+1} = r_{i,t}^n - \eta_{i,t}^n h_{i,t}^n$ .
  - 9:    $\zeta_{i,t}^n = \frac{(r_{i,t}^{n+1})^T r_{i,t}^{n+1}}{r_{i,t}^{n,T} r_{i,t}^n}$ .
  - 10:    $p_{i,t}^{n+1} = r_{i,t}^{n+1} + \zeta_{i,t}^n p_{i,t}^n$ .
  - 11: **end for**
  - 12: **Output:**  $v_{i,t}^N$ .
- 

where  $H_i$  and  $b_i$  are given by  $H_i = \nabla_{\phi}^2 g_i(\theta, \phi_i^*(\theta))$  and  $b_i = \nabla_{\phi} f_i(\theta, \phi_i^*(\theta))$ , respectively. We present Subroutine 1 to find the optimal solution  $v_i^*$  to problem (7).

Building on Subroutine 1, we can estimate the hypergradient in (6) by using the following equality:

$$\nabla \widehat{F}_{\mathcal{B}}(\theta_t) = \nabla_{\theta} f_{\mathcal{B}}(\theta_t, \phi_t^K) - \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \nabla_{\theta\phi}^2 g_i(\theta, \phi_{i,t}^K) v_{i,t}^N. \quad (8)$$

With the hypergradient estimation (8), we propose a memory-reduced algorithm for solving the meta-learning problem (1) in Algorithm 1.

In Algorithm 1, we use parameters  $\phi_{i,t}^K$  at the last inner-loop iteration for hypergradient estimation. This is different from existing iterative-differentiation-based meta-learning approaches, such as Finn, Abbeel, and Levine (2017); Zhou, Wu, and Li (2018); Oh et al. (2021); Antoniou, Edwards, and Storkey (2019); Raghu et al. (2019); Baik et al. (2021); Yao, Zhang, and Finn (2022), which employ historical lower-level parameters  $\phi_i^k(\theta)$ ,  $k \in \{0, \dots, K\}$  for hypergradient estimation; see Eq. (3) for details. Our Algorithm 1 significantly reduces memory costs in each outer-loop iteration  $t$ , which is evidenced by our experimental results in Figure 2. Moreover, the avoidance of long-distance backpropagation in Algorithm 1 also reduces the risk of exploding or vanishing gradients, leading to better learning accuracy than existing iterative-differentiation-based meta-learning approaches in Finn, Abbeel, and Levine (2017); Raghu et al. (2019); Ji, Yang, and Liang (2020), as evidenced by our experimental results in Figure 1 and Table 1.

Inspired by the linear acceleration capability of conjugate gradient approaches in solving the quadratic programming problem (Grazzi et al. 2020), we leverage the conjugate gradient in our Subroutine 1. However, different from the existing conjugate gradient-based algorithm in Ji, Yang, and Liang (2020) which relies on deterministic gradients, we employ stochastic gradients, which complicates our convergence analysis.

---

**Algorithm 1: Memory-reduced meta-learning algorithm**

---

1: **Input:** The batch size of sampled tasks  $|\mathcal{B}|$ ; random initialization  $\theta_0$  and  $\phi_{i,0}$  for all  $i \in \{1, \dots, |\mathcal{B}|\}$ ; stepsizes  $\lambda_\phi$  and  $\lambda_\theta$ ; integers  $T$  and  $K$ .  
2: **for**  $t = 0, 1, 2, \dots, T - 1$  **do**  
3:   Sample a task batch  $\mathcal{B} \sim \mathcal{P}(\mathcal{T})$ .  
4:    $u_0 = \mathbf{0}_p$ .  
5:   **for**  $i = 1, \dots, |\mathcal{B}|$  **do**  
6:     Set  $\phi_{i,t}^0 = \phi_{i,t-1}^K$  if  $t > 0$ , and  $\phi_{i,0}$  otherwise.  
7:     **for**  $k = 0, 1, \dots, K - 1$  **do**  
8:        $\phi_{i,t}^{k+1} = \phi_{i,t}^k - \lambda_\phi \nabla_{\phi} g_i(\theta_t, \phi_{i,t}^k)$ .  
9:     **end for**  
10:    Run Subroutine 1 and obtain Hessian-inverse-vector product  $v_{i,t}^N$ .  
11:    Accumulate Jacobin-vector products  $u_i = u_{i-1} + \nabla_{\theta}^2 g_i(\theta_t, \phi_{i,t}^K) v_{i,t}^N$ .  
12:    **end for**  
13:    Compute  $\nabla \widehat{F}_{\mathcal{B}}(\theta_t) = \nabla_{\theta} f_{\mathcal{B}}(\theta_t, \phi_t^K) - \frac{1}{|\mathcal{B}|} u_{|\mathcal{B}|}$ .  
14:     $\theta_{t+1} = \theta_t - \lambda_\theta \nabla \widehat{F}_{\mathcal{B}}(\theta_t)$ .  
15: **end for**  
16: **Output:**  $\theta_T$

---

**Remark 3.** To deal with the bilevel programming objectives (where upper-level optimality relies on lower-level optimality), we employ nested-loop iterations in Algorithm 1. Note that nested-loop iterations are commonly used in meta-learning algorithms, such as Raghu et al. (2019); Rajeswaran et al. (2019). The iteration numbers  $K$  and  $N$  in Algorithm 1 are fixed constants, which are independent of the outer-loop iteration number  $T$ . This is different from existing bilevel programming approaches in Chen et al. (2022); Chen, Sun, and Yin (2021) which have the inner-loop iteration number increasing with the outer-loop iteration, and hence have a heavier computational overhead. In addition, different from existing meta-learning algorithms in Hiller, Harandi, and Drummond (2022); Park and Oliva (2019) which estimate the full Hessian matrix or Jacobian matrix, Algorithm 1 only estimates a vector of dimension  $\max\{p, q\}$ , and thus reduces computational complexity.

## 4 Convergence Analysis

Algorithm 1 can ensure sublinear convergence with the number  $T$  of outer-loop iteration, and the convergence error decreases sublinearly with the batch size of the sampled tasks. The results are summarized in Theorem 1, whose proof is given in Section B of the supplementary material.

**Theorem 1.** Under Assumptions 1 and 2, if the iteration numbers  $K$  and  $N$  satisfy  $K \geq K_0$  and  $N \geq N_0$  with detailed forms of  $K_0$  and  $N_0$  given in Section B.2 of the supplementary material, the iterates  $\theta_t$  generated by Algorithm 1 satisfy

$$\frac{1}{T+1} \sum_{t=0}^T \mathbb{E}[\|\nabla F(\theta_t)\|^2] \leq \mathcal{O}\left(\frac{1}{T}\right) + \mathcal{O}\left(\frac{1}{|\mathcal{B}|}\right).$$

Theorem 1 proves that Algorithm 1 converges to a stable solution to problem (1) with the optimization error de-

creasing as the batch size of sampled tasks increases. We would like to point out that the bound  $\mathcal{O}(\frac{1}{|\mathcal{B}|})$  in Theorem 1, caused by finite batch sizes of sampled tasks, inherently exists in all stochastic optimization approaches with finite samples (Gower et al. 2019). Although the variance-reduction technique (Reddi et al. 2016; Fang et al. 2018) can be used to mitigate the influence of this term in single-level stochastic optimization, the extension of this approach to meta-learning/bilevel programming is hard to implement, since it is difficult to derive unbiased estimators of hypergradient, let alone variance reduction ones; see Dagr eou et al. (2022) for details.

Moreover, to give a more intuitive description of the computational complexity, we define an  $\epsilon$ -solution to problem (1) as follows.

**Definition 1.** (Lian et al. 2017) For some positive integer  $T$ , if  $\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\theta_t)\|^2] \leq \epsilon$  holds, then we say that the sequence  $\{\theta_t\}_{t=0}^T$  can reach an  $\epsilon$ -solution to problem (1).

Building on Theorem 1, we have the following corollary.

**Corollary 1.** Under the conditions of Theorem 1, for any  $\epsilon > 0$ , Algorithm 1 requires at most  $\mathcal{O}((3 + 2K_0)|\mathcal{B}|\epsilon^{-1})$  gradient evaluations on  $\phi$  and  $\mathcal{O}((1 + K_0|\mathcal{B}|\epsilon^{-1})$  gradient evaluations on  $\theta$  to obtain an  $\epsilon$ -solution.

In Corollary 1, the inner-loop iteration number  $K_0$  in Algorithm 1 is a fixed constant, which is different from the existing bilevel programming approaches in Chen et al. (2022); Chen, Sun, and Yin (2021) which have the inner-loop iteration number increasing with the outer-loop iteration, and hence have a higher computational complexity of the order of  $\mathcal{O}(\epsilon^{-2})$ . Moreover, the computational complexity of our Algorithm 1 matches the convergence results for MAML and ANIL in Ji et al. (2020), even when we use less memory in each outer iteration.

In Theorem 1, we consider a stochastic scenario in which tasks are drawn from an unknown task distribution  $\mathcal{P}(\mathcal{T})$ . Next, we consider a deterministic scenario in which we iterate over all sample elements in the task space. In this case, Assumption 2 becomes a deterministic version as follows.

**Assumption 3.** For each task  $\mathcal{T}_i$ , functions  $f_i$ ,  $\nabla f_i$ ,  $\nabla g_i$ ,  $\nabla_{\theta}^2 g_i$ , and  $\nabla_{\phi}^2 g_i$  are  $l_{f,0}$ ,  $l_f$ ,  $l_g$ ,  $l_{g,1}$ , and  $l_{g,2}$  Lipschitz continuous, respectively.

**Theorem 2.** Under Assumptions 1 and 3, if the iteration numbers  $K$  and  $N$  satisfy  $K \geq \mathcal{O}(\kappa)$  and  $N \geq \mathcal{O}(\sqrt{\kappa})$  with  $\kappa = \frac{l_g}{\mu}$ , then the iterates  $\theta_t$  generated by Algorithm 1 with deterministic gradients satisfy

$$\frac{1}{T+1} \sum_{t=0}^T \|\nabla F(\theta_t)\|^2 \leq \mathcal{O}\left(\frac{1}{T}\right).$$

Theorem 2 demonstrates that when we consider deterministic meta-learning, Algorithm 1 converges to an exact solution to problem (1) with a sublinear convergence rate. To achieve an  $\epsilon$ -solution, Algorithm 1 with deterministic gradients requires at most  $\mathcal{O}(\kappa\epsilon^{-1})$  gradient evaluations in both  $\phi$  and  $\theta$ , which is consistent with the convergence results for the iterative-differentiation-based meta-learning algorithm in Ji, Yang, and Liang (2020).

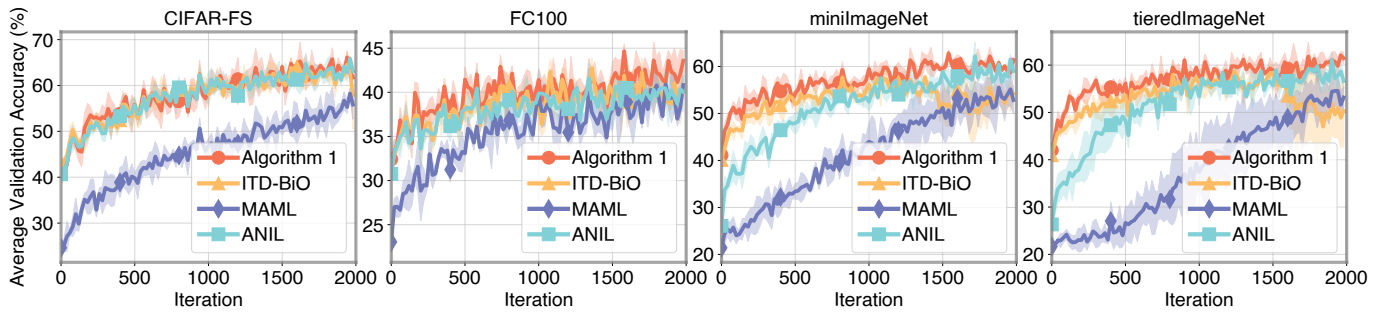


Figure 1: Validation accuracies of Algorithm 1, MAML, ANIL, and ITD-BiO in meta-learning stage using the “CIFAR-FS”, “FC100”, “miniImageNet”, “tieredImageNet” datasets, respectively.

Algorithm	Backbone <sup>a</sup>	CIFAR-FS Accuracy(%)	FC100 Accuracy(%)	Backbone	miniImageNet Accuracy(%)	tieredImageNet Accuracy(%)
MAML	64-64-64-64	54.85 ± 1.23	47.50 ± 1.06	32-32-32-32	43.33 ± 0.95	48.37 ± 1.17
ANIL	64-64-64-64	63.52 ± 1.31	47.70 ± 0.95	32-32-32-32	57.80 ± 1.14	58.16 ± 0.94
ITD-BiO	64-64-64-64	63.69 ± 1.10	47.88 ± 0.74	32-32-32-32	55.58 ± 1.31	60.78 ± 0.10
<b>Algorithm 1 (ours)</b>	64-64-64-64	<b>64.24 ± 1.41</b>	<b>48.52 ± 1.13</b>	32-32-32-32	<b>59.58 ± 1.08</b>	<b>61.97 ± 0.86</b>

<sup>a</sup> The “Backbone” represents the remaining layers in the model except for the last linear layer. The numbers in “64-64-64-64” and “32-32-32-32” represent the number of filters in each convolutional layer in the backbone.

Table 1: Test accuracies by fine-tuning the model learned by Algorithm 1, MAML, ANIL, and ITD-BiO using 20-step gradient descent on the “CIFAR-FS”, “FC100”, “miniImageNet”, “tieredImageNet” datasets, respectively.

## 5 Experiments

In this section, we evaluate the performance of our proposed Algorithm 1 by using a few-shot image classification problem on the “CIFAR-FS” dataset (Bertinetto et al. 2018), the “FC100” dataset (Oreshkin, Rodríguez López, and Lacoste 2018), the “miniImageNet” dataset (Vinyals et al. 2016), and the “tieredImageNet” dataset (Ren et al. 2018), respectively. In all experiments, we compare our Algorithm 1 with other optimization-based meta-learning approaches, including the bilevel optimization-based algorithm with iterative differentiation (ITD-BiO) in Ji, Yang, and Liang (2020), MAML in Finn, Abbeel, and Levine (2017), and ANIL in Raghu et al. (2019). Due to space limitations, we leave the experimental setup in neural network training in Appendix C.2.

### 5.1 Evaluation on Meta-Learning Accuracy

Following Finn, Abbeel, and Levine (2017), we consider a meta-learning problem with  $|\mathcal{B}|$  tasks  $\{\mathcal{T}_i, i = 1, \dots, |\mathcal{B}|\}$  in each iteration. Each task  $\mathcal{T}_i$  has a loss function  $\mathcal{L}(\theta, \phi_i; \xi)$  over each data sample  $\xi$ , where  $\theta$  represents the parameters of an embedding model shared by all tasks, and  $\phi_i$  represents the task-specific parameters in a given task  $\mathcal{T}_i$ . The goal of our meta-learning problem (1) is to find the optimal parameters  $\theta^*$  that benefit all tasks, and based on  $\theta^*$ , the model can quickly adapt its own parameters  $\phi_i$  to any new task  $\mathcal{T}_i$  using only a few data points and training iterations.

In this experiment, all comparative meta-learning algorithms are trained using a four-convolutional-layer convolutional neural network (CNN) architecture given in Ravi and Larochelle (2016). The network is trained using a cross-

entropy loss function in PyTorch. For our Algorithm 1 and ITD-BiO in Ji, Yang, and Liang (2020),  $\phi_i$  corresponds to the parameters of the last linear layer of the CNN model and  $\theta$  represents the parameters of the remaining layers. This setup ensures that the lower-level objective function  $g_i(\theta, \phi)$  is strongly convex with respect to  $\phi$ , while the upper-level objective function  $F(\theta)$  is generally nonconvex with respect to  $\theta$ . For MAML and ANIL, we employ their default initialization-based learning paradigms given in Finn, Abbeel, and Levine (2017) and Raghu et al. (2019), respectively. All algorithms are executed in  $|\mathcal{B}| = 32$  batches of tasks over 2,000 iterations with each task involving a training dataset  $\mathcal{D}_i^{\text{tra}}$  and a validation dataset  $\mathcal{D}_i^{\text{val}}$ , both designed for 5-way classification with 5 shots for each class. For the experiments conducted on the “CIFAR-FS” and “FC100” datasets, we set the step sizes to  $\lambda_\theta = 0.0001$  and  $\lambda_\phi = 0.1$ ; and for the experiments conducted on the “miniImageNet” and “tieredImageNet” datasets, we set  $\lambda_\theta = 0.001$  and  $\lambda_\phi = 0.05$ . The iteration numbers  $K$  and  $N$  are configured as  $K = N = 20$ . In our comparison, the near-optimal step-sizes and inner-loop iteration numbers are applied to ITD-BiO, MAML and ANIL, such that doubling them leads to divergent behavior. Moreover, for ITD-BiO, MAML, and ANIL, we used Adam (Kingma and Ba 2017) for upper-level optimization and gradient descent for lower-level optimization, following the guidance from Ji, Yang, and Liang (2020), Finn, Abbeel, and Levine (2017), and Raghu et al. (2019), respectively.

Note that evaluating the performance of a meta-learning algorithm involves two stages: learning-accuracy evaluation

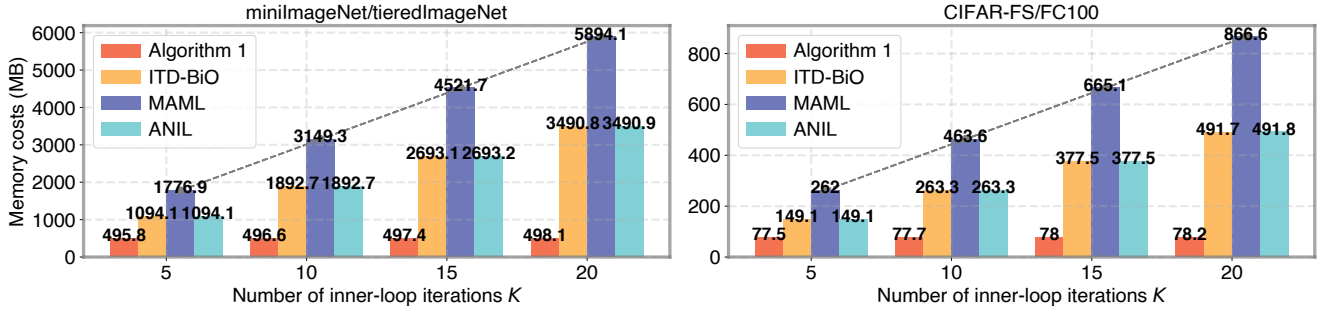


Figure 2: Memory costs of Algorithm 1, MAML, ANIL, and ITD-BiO under different numbers of inner-loop iterations  $K$  using the “CIFAR-FS”, “FC100”, “miniImageNet”, “tieredImageNet” datasets, respectively.

Algorithm	CIFAR-FS wallclock time(s) <sup>a</sup>	FC100 wallclock time(s)	miniImageNet wallclock time(s)	tieredImageNet wallclock time(s)
MAML	— <sup>b</sup>	2638.45	10065.83	9453.75
ANIL	1130.16	879.33	1639.85	1417.64
ITD-BiO	1740.76	1607.74	3031.51	2059.43
<b>Algorithm 1 (ours)</b>	<b>832.40</b>	<b>621.67</b>	<b>739.62</b>	<b>693.46</b>

<sup>a</sup> The wallclock time represents the time spent to achieve a certain validation accuracy of  $0.90 \times Acc_{max}$ , where  $Acc_{max}$  denotes the highest validation accuracy achieved among all comparison algorithms.

<sup>b</sup> Since MAML can never reach  $0.90 \times Acc_{max}$ , we have not provided its wallclock time here.

Table 2: Comparison with MAML, ANIL, and ITD-BiO in terms of wallclock time using the “CIFAR-FS”, “FC100”, “miniImageNet”, “tieredImageNet” datasets, respectively.

in the meta-learning stage and generalization-ability evaluation in the meta-test stage. The learning accuracy in the meta-learning stage is shown in Figure 1. These results confirm the advantage of the proposed algorithm over existing meta-learning algorithms on various datasets. To show that Algorithm 1’s generalization ability to a new task which is comprised of some images unseen in the meta-learning stage, we list the test accuracies by fine-tuning the model learned by Algorithm 1, MAML, ANIL, and ITD-BiO using a 20-step gradient descent. The results are summarized in Table 1, which shows that our proposed Algorithm 1 has better generalization ability than existing counterparts.

## 5.2 Evaluation on Memory Costs

In optimization-based meta-learning algorithms, a significant portion of memory costs come from storing historical lower-level parameters/gradients for hypergradient estimation (see Eq.(3)). To show that Algorithm 1 can ensure nearly time-invariant memory usage over iterations, we compare the memory needed under different numbers of inner-loop iterations. The results in Figure 2 clearly show that the memory costs of Algorithm 1 remain consistently stable regardless of the number of inner-loop iterations. In contrast, the memory costs for MAML, ANIL, and ITD-BiO increase with the number of inner-loop iterations. Moreover, even when the number of inner-loop iterations is set to 5, the memory consumption of our algorithm is more than 50% lower compared to MAML, ANIL, and ITD-BiO.

## 5.3 Evaluation on Wallclock Time

To compare the convergence rate, i.e., computational complexity, of our Algorithm 1 with existing counterparts, we summarize their wallclock time to achieve a certain  $0.90 \times Acc_{max}$  validation accuracy in Table 2, where  $Acc_{max}$  represents the highest validation accuracy achieved among all comparison algorithms. The experimental results validate that our algorithm is still faster than MAML, ANIL, and ITD-BiO in wallclock time even without using any historical lower-level parameters/gradients.

## 6 Conclusions

In this paper, we propose a meta-learning algorithm that can simultaneously reduce memory costs and ensure sub-linear convergence. More specifically, our proposed approach avoids using any historical lower-level parameters/gradients, and hence, ensures nearly invariant memory costs over iterations. This is in sharp contrast to most existing iterative-differentiation-based algorithms for meta-learning, which use historical lower-level parameters/gradients to ensure learning accuracy, implying growing memory costs when the inner-loop iteration number increases. In addition, we systematically characterize the convergence performance of our algorithm for both stochastic and deterministic meta-learning, and quantify the computational complexities for gradient evaluations on both upper-level and lower-level parameters. Experimental results on various benchmark datasets in few-shot meta-learning confirm the advantages of the proposed approach over existing counterparts.

## Acknowledgments

This work was supported in part by the National Science and Technology Major Project under Grant 2021ZD0112600, by the National Natural Science Foundation of China under Grants 62103343 and 62173283, and by the Fujian Provincial Natural Science Foundation of China under Grant 2024J01051.

## References

- Antoniou, A.; Edwards, H.; and Storkey, A. 2019. How to train your MAML. arXiv:1810.09502.
- Baik, S.; Choi, J.; Kim, H.; Cho, D.; Min, J.; and Lee, K. M. 2021. Meta-learning with task-adaptive loss function for few-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9465–9474.
- Bertinetto, L.; Henriques, J. F.; Torr, P. H. S.; and Vedaldi, A. 2018. Meta-learning with differentiable closed-form solvers. *International Conference on Learning Representations*.
- Chen, T.; Sun, Y.; Xiao, Q.; and Yin, W. 2022. A single-timescale method for stochastic bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, 2466–2488. PMLR.
- Chen, T.; Sun, Y.; and Yin, W. 2021. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Advances in Neural Information Processing Systems*, 34: 25294–25307.
- Chen, Z.; Liang, S.; Li, L.; and Cheng, S. 2024. Quantized Primal-Dual Algorithms for Network Optimization With Linear Convergence. *IEEE Transactions on Automatic Control*, 69(1): 471–478.
- Chen, Z.; and Wang, Y. 2024a. Locally Differentially Private Decentralized Stochastic Bilevel Optimization with Guaranteed Convergence Accuracy. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 7389–7439. PMLR.
- Chen, Z.; and Wang, Y. 2024b. Locally Differentially Private Gradient Tracking for Distributed Online Learning over Directed Graphs. *IEEE Transactions on Automatic Control*, 1–16.
- Chi, Z.; Gu, L.; Liu, H.; Wang, Y.; Yu, Y.; and Tang, J. 2022. MetaFscil: A meta-learning approach for few-shot class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14166–14175.
- Dagr eou, M.; Ablin, P.; Vaiter, S.; and Moreau, T. 2022. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. *Advances in Neural Information Processing Systems*, 35: 26698–26710.
- Denevi, G.; Stamos, D.; Ciliberto, C.; and Pontil, M. 2019. Online-within-online meta-learning. *Advances in Neural Information Processing Systems*, 32.
- Domke, J. 2012. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*, 318–326. PMLR.
- Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, 1082–1092. PMLR.
- Fang, C.; Li, C. J.; Lin, Z.; and Zhang, T. 2018. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in neural information processing systems*, 31.
- Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *International Conference on Machine Learning*.
- Finn, C.; Yu, T.; Zhang, T.; Abbeel, P.; and Levine, S. 2017. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, 357–368. PMLR.
- Franceschi, L.; Donini, M.; Frasconi, P.; and Pontil, M. 2017. Forward and Reverse Gradient-Based Hyperparameter Optimization. In *International Conference on Machine Learning*.
- Franceschi, L.; Frasconi, P.; Salzo, S.; Grazzi, R.; and Pontil, M. 2018. Bilevel Programming for Hyperparameter Optimization and Meta-Learning. *International Conference on Machine Learning*.
- Ghadimi, S.; and Wang, M. 2018. Approximation Methods for Bilevel Programming. arXiv:1802.02246.
- Gower, R. M.; Loizou, N.; Qian, X.; Sailanbayev, A.; Shulgin, E.; and Richt arik, P. 2019. SGD: General analysis and improved rates. In *International conference on machine learning*, 5200–5209. PMLR.
- Grazzi, R.; Franceschi, L.; Pontil, M.; and Salzo, S. 2020. On the iteration complexity of hypergradient computation. In *Proc. International Conference on Machine Learning (ICML)*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Hiller, M.; Harandi, M.; and Drummond, T. 2022. On enforcing better conditioned meta-learning for rapid few-shot adaptation. *Advances in Neural Information Processing Systems*, 35: 4059–4071.
- Jamal, M. A.; Wang, L.; and Gong, B. 2021. A lazy approach to long-horizon gradient-based meta-learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6577–6586.
- Ji, K.; Lee, J. D.; Liang, Y.; and Poor, H. V. 2020. Convergence of meta-learning with task-specific adaptation over partial parameters. *Advances in Neural Information Processing Systems*, 33: 11490–11500.
- Ji, K.; Yang, J.; and Liang, Y. 2020. Bilevel Optimization: Convergence Analysis and Enhanced Design. *International Conference on Machine Learning*.
- Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.
- Kirsch, L.; Flennerhag, S.; Van Hasselt, H.; Friesen, A.; Oh, J.; and Chen, Y. 2022. Introducing symmetries to black box meta reinforcement learning. In *Proceedings of*

- the AAAI Conference on Artificial Intelligence, volume 36, 7202–7210.
- Lee, J.; Yoo, J.; and Kwak, N. 2023. SHOT: suppressing the hessian along the optimization trajectory for gradient-based meta-learning. *Advances in Neural Information Processing Systems*, 36: 61450–61465.
- Li, H.; Dong, W.; Mei, X.; Ma, C.; Huang, F.; and Hu, B.-G. 2019. LGM-Net: Learning to generate matching networks for few-shot learning. In *International conference on machine learning*, 3825–3834. PMLR.
- Li, J.; Lu, T.; Cao, X.; Cai, Y.; and Wang, S. 2021. Meta-imitation learning by watching video demonstrations. In *International Conference on Learning Representations*.
- Lian, X.; Zhang, C.; Zhang, H.; Hsieh, C.-J.; Zhang, W.; and Liu, J. 2017. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in neural information processing systems*, 30.
- Lorraine, J.; Vicol, P.; and Duvenaud, D. 2020. Optimizing millions of hyperparameters by implicit differentiation. In *International conference on artificial intelligence and statistics*, 1540–1552. PMLR.
- Maclaurin, D.; Duvenaud, D.; and Adams, R. 2015. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, 2113–2122. PMLR.
- Nichol, A.; Achiam, J.; and Schulman, J. 2018. On First-Order Meta-Learning Algorithms. arXiv:1803.02999.
- Oh, J.; Yoo, H.; Kim, C.; and Yun, S. 2021. BOIL: Towards Representation Change for Few-shot Learning. In *International Conference on Learning Representations*.
- Oreshkin, B.; Rodríguez López, P.; and Lacoste, A. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31.
- Park, E.; and Oliva, J. B. 2019. Meta-curvature. *Advances in neural information processing systems*, 32.
- Peng, H. 2020. A Comprehensive Overview and Survey of Recent Advances in Meta-Learning. arXiv:2004.11149.
- Raghu, A.; Raghu, M.; Bengio, S.; and Vinyals, O. 2019. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. *International Conference on Learning Representations*. Freezing backbone—body and head ANIL and NIL.
- Rajeswaran, A.; Finn, C.; Kakade, S.; and Levine, S. 2019. Meta-Learning with Implicit Gradients. *Neural Information Processing Systems*.
- Ravi, S.; and Larochelle, H. 2016. Optimization as a model for few-shot learning. In *International conference on learning representations*.
- Reddi, S. J.; Hefny, A.; Sra, S.; Póczos, B.; and Smola, A. 2016. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, 314–323. PMLR.
- Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J. B.; Larochelle, H.; and Zemel, R. S. 2018. Meta-Learning for Semi-Supervised Few-Shot Classification. arXiv:1803.00676.
- Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Lillicrap, T. 2016. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, 1842–1850. PMLR.
- Shaban, A.; Cheng, C.-A.; Hatch, N.; and Boots, B. 2019. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 1723–1732. PMLR.
- Shu, J.; Meng, D.; and Xu, Z. 2021. Learning an Explicit Hyper-parameter Prediction Function Conditioned on Tasks. *Journal of machine learning research*.
- Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.
- Soydaner, D. 2020. A comparison of optimization algorithms for deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13): 2052013.
- Ullah, I.; Carrión-Ojeda, D.; Escalera, S.; Guyon, I.; Huisman, M.; Mohr, F.; van Rijn, J. N.; Sun, H.; Vanschoren, J.; and Vu, P. A. 2022. Meta-album: Multi-domain meta-dataset for few-shot image classification. *Advances in Neural Information Processing Systems*, 35: 3232–3247.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.
- Vuorio, R.; Sun, S.-H.; Hu, H.; and Lim, J. J. 2019. Multi-modal model-agnostic meta-learning via task-aware modulation. *Advances in neural information processing systems*, 32.
- Wang, Y.; Yao, Q.; Kwok, J. T.; and Ni, L. M. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3): 1–34.
- Wu, Y.; Liang, T.; Feng, S.; Jin, Y.; Lyu, G.; Fei, H.; and Wang, Y. 2023. Metazscil: A meta-learning approach for generalized zero-shot class incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 10408–10416.
- Yao, H.; Zhang, L.; and Finn, C. 2022. Meta-Learning with Fewer Tasks through Task Interpolation. arXiv:2106.02695.
- Zhang, B.; Luo, C.; Yu, D.; Li, X.; Lin, H.; Ye, Y.; and Zhang, B. 2024. Metadiff: Meta-learning with conditional diffusion for few-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16687–16695.
- Zhou, F.; Wu, B.; and Li, Z. 2018. Deep Meta-Learning: Learning to Learn in the Concept Space. arXiv:1802.03596.
- Zhou, P.; Yuan, X.; Xu, H.; Yan, S.; and Feng, J. 2019. Efficient meta learning via minibatch proximal update. *Advances in Neural Information Processing Systems*, 32.