Towards Gradient-based Bilevel Optimization with Non-convex Followers and Beyond

Anonymous Author(s)
Affiliation
Address
email

Abstract

In recent years, Bi-Level Optimization (BLO) techniques have received extensive attentions from both learning and vision communities. A variety of BLO models in complex and practical tasks are of non-convex follower structure in nature (a.k.a., without Lower-Level Convexity, LLC for short). However, this challenging class of BLOs is lack of developments on both efficient solution strategies and solid theoretical guarantees. In this work, we propose a new algorithmic framework, named Initialization Auxiliary and Pessimistic Trajectory Truncated Gradient Method (IAPTT-GM), to partially address the above issues. In particular, by introducing an auxiliary as initialization to guide the optimization dynamics and designing a pessimistic trajectory truncation operation, we construct a reliable approximate version of the original BLO in the absence of LLC hypothesis. Our theoretical investigations establish the convergence of solutions returned by IAPTT-GM towards those of the original BLO without LLC. As an additional bonus, we also theoretically justify the quality of our IAPTT-GM embedded with Nesterov's accelerated dynamics under LLC. The experimental results confirm both the convergence of our algorithm without LLC, and the theoretical findings under LLC.

1 Introduction

Bi-Level Optimization (BLO) has been widely used to formulate problems in the field of deep learning [1], especially for hyper-parameter optimization [2, 3, 4], meta learning [5, 6, 7], neural architecture search [8, 9, 10], adversarial learning [11], and reinforcement learning [12], etc. BLO aims to tackle nested optimization structures appearing in applications, which has emerged as a prevailing optimization technique for modern machine learning tasks with underlying hierarchy. In the last decade, a large number of BLO methods have been proposed to address different machine learning tasks. In fact, Gradient Methods (GMs), which can handle BLO problems of large scale effectively, thus gain popularity. According to different types of strategies for gradient calculations, existing GMs can be divided into two categories, i.e., the explicit approaches which aim to replace the LL problem with dynamic iterations and implicit schemes that apply the implicit function theorem to formulate the first-order optimality condition of the LL problem.

Explicit Gradient Methods for BLOs. In this type, solving the LL problem is regarded as the evolution path of the dynamic system starting from a given initial point of the LL variable. The gradient of the UL variable can be calculated directly by automatic differentiation based on the trajectory of LL variable. This class of methods can be further divided into three types, namely, recurrence-based EG (e.g., [2, 13, 5, 14, 8]), initialization-based EG (e.g., [15, 16]) and proxy-based EG methods (e.g., [17, 18, 19, 20]), differing from each other in the way of accessing the gradient of constructed dynamic trajectory. While most of this type of works assume the LLC and Lower Level Singleton (LLS) to simplify their optimization process and theoretical analysis, cases where the LLS

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

assumption does not hold have been tackled in the recent work [21]. In particular, to eliminate the LLS assumption which is too restrctive to be satisfied in real-world complex tasks, [21] first considers incorporating UL objective information into the dynamic iterations, but the more general cases where LLC does not hold remain unsolved. On the other hand, while most of the mentioned works focus on the asymptotic convergence, the progress on nonasymptotic convergence analysis has been recently witnessed see, e.g., [22, 23, 24].

Implicit Gradient Methods for BLOs. This type, also known as implicit differentiation [25, 6, 26], replaces the LL problem with its first-order optimality condition and uses the implicit function theorem to calculate the gradient of the UL problem by solving a linear system. This method decouples the calculation of UL gradient from the dynamic system of LL, resulting in a significant speed increase when the dynamic system iterates many times. However, because of the burden originated from computing a Hessian matrix and its inverse, IG methods are usually computationally expensive when linear systems are ill-conditioned. To alleviate this computational issue, there are mainly two kinds of techniques, i.e., IG based on Linear System (LS) [25, 6] and Neumann Series (NS) [26]. On the theoretical side, IG methods rely heavily on the strong convexity of LL problems, which is even more restrictive than the LLC and LLS together.

Initialization Optimization for Learning. In deep learning, the selection of initialization scheme has a great influence on the training speed and performance [27]. As the most representative work in recent years, Model-Agnostic Meta-Learning (MAML) [28] applies the same initialization to all tasks, and optimized by a loss function common to the task that evaluates the effect of the initial value, resulting in an initialization that achieves good generalization performance with only a few gradient steps on new tasks. Due to its simple form this method has been widely studied and applied [29, 30, 31, 32]. [33] noticed that not all network parameters are suitable for the same initialization, and therefore proposed a strategy to apply co-initialization only on a part of parameters. In theory, [34, 35, 36] give comprehensively study on the convergence and convergence rate of MAML and some MAML-type approaches based on the meta objective function. However, the convergence theory of these existing results are given based on the loss function for evaluating initial values, and the convergence analysis of such type of methods from the perspective of each task is still lacked.

1.1 Our Motivations and Contributions

As mentioned above, some theoretical progresses have long been witnessed in diversified learning areas, but for most existing BLO methods, extra restrictive assumptions (e.g., LLS, LLC and LL strong convexity) have to be enforced. Their algorithm design and associated theoretical analysis actually are only valid for optimization with simplified problem structure. Unfortunately, it has been well recognized that LL non-convexity frequently appears in a variety of applications, e.g., sparse ℓ_q regularization $(0 < q < 1)$ for avoiding over fitting, and learning parameters of coupled multi-layer neural networks, etc. Therefore, in challenging real-world scenarios, we are usually required to consider BLO problems where *these assumptions (e.g., LLS, LLC and even LL strong convexity) are naturally violated*. These fundamental theoretical issues motivate us to propose a series of new techniques to address BLO with non-convex LL problems, which have been frequently appeared in various learning applications.

In particular, by introducing an Initialization Auxiliary (IA) to the LL optimization dynamics and operating a Pessimistic Trajectory Truncation (PTT) strategy during the UL approximation, we construct a Gradient-based Method (GM), named IAPTT-GM, to address BLO in challenging optimization scenarios (i.e., with non-convex follower tasks). We analyze the convergence behaviors of IAPTT-GM on BLOs without LLC and also investigate theoretical properties of the accelerated version of our algorithm on BLOs under LLC. Extensive experiments verify our theoretical results and demonstrate the effectiveness of IAPTT-GM on different learning applications. The main contributions of our IAPTT-GM are summarized as follows:

- We propose IA and PTT, two new mechanisms to efficiently handle complex BLOs where the follower is facing with a non-convex task (i.e., without LLS and even LLC). IA actually paves the way for jointly optimizing both the UL variables and the dynamical initialization, while PTT adaptively reduces the complexity of backward recurrent propagation.

- To our best knowledge, we establish the first strict convergence guarantee for gradient-based method on BLOs with non-convex follower tasks. We also justify the quality of our IAPTT-GM embedded with Nesterov's accelerated dynamics under LLC.
- We conduct a series of experiments to verify our theoretical findings and evaluate IAPTT-GM on various challenging BLOs, in which the follower tasks are either with non-convex loss functions (few-shot learning) or coupled network structures (data hyper-cleaning).

2 The Proposed Algorithmic Framework

In this work, we consider the BLO problem in the form:

$$\min_{\mathbf{x}\in\mathcal{X},\mathbf{y}} F(\mathbf{x},\mathbf{y}), \quad s.t. \quad \mathbf{y}\in\mathcal{S}(\mathbf{x}), \tag{1}$$

where $\mathbf{x}\in\mathbb{R}^n, \mathbf{y}\in\mathbb{R}^m$ are UL and LL variables respectively, and $\mathcal{S}(\mathbf{x})$ denotes the set of solutions of the LL problem, i.e.,

$$\mathcal{S}(\mathbf{x}) := \arg\min_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x},\mathbf{y}), \tag{2}$$

where f is differentiable w.r.t. $\mathbf{y}$. To ensure the BLO model in Eq. (1) is well-defined, we assume that $\mathcal{S}(\mathbf{x})$ is nonempty for all $\mathbf{x}\in\mathcal{X}$. Observe further that the above BLO is structurally different from those in existing literature in the sense that no convexity assumption is required in the LL problem.

In the following, we describe the proposed IAPTT-GM to solve the class of BLOs defined in Eq. (1). The mechanism of a classical dynamics-embedded gradient method, approximates the LL solution via a dynamical system drawn from optimization iterations. Choosing gradient descent as the optimization dynamics for example, the approximation $\mathbf{y}_K(\mathbf{x})$ is accessed by operations performed repeatedly by $K-1$ steps parameterized with UL variable $\mathbf{x}$

$$\mathbf{y}_{k+1}(\mathbf{x}) = \mathbf{y}_k(\mathbf{x}) - s\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y}_k(\mathbf{x})),\ k=0,\cdots,K-1, \tag{3}$$

where s is a step size, and $\mathbf{y}_0$ is a fixed initial value.

Initialization Auxiliary. Embedding the dynamical iterations into the UL problem returns the approximate version $F(\mathbf{x},\mathbf{y}_K(\mathbf{x}))$. As long as $\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y}_K(\mathbf{x}))$ uniformly converges to zero w.r.t. UL variable $\mathbf{x}$ varying in $\mathcal{X}$, we call this a good approximation. To this end, usually restrictive LL strong convexity assumptions are imposed, thus the desired convergence of solutions of approximation problems towards those of the original BLO follows. By drawing inspiration from the classic dynamics-embedded gradient method which replaces the LL problem with certain optimization dynamics, hence resulting in an approximation of the bi-level problem, we propose a new gradient scheme to solve BLO in Eq. (1) without LLC restriction. Specifically, we let K be a prescribed positive integer and construct the following approximation $\mathbf{y}_K(\mathbf{x},\mathbf{z})$ of the LL solution drawn from projected gradient descent iterations

$$\begin{aligned}&\mathbf{y}_0(\mathbf{x},\mathbf{z}) = \mathbf{z},\\ &\mathbf{y}_{k+1}(\mathbf{x},\mathbf{z}) = \mathtt{Proj}_{\mathcal{Y}}(\mathbf{y}_k(\mathbf{x},\mathbf{z}) - \alpha_{\mathbf{y}}^k \nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y}_k(\mathbf{x},\mathbf{z}))),\ k=0,\cdots,K-1,\end{aligned} \tag{4}$$

where $\{\alpha_{\mathbf{y}}^k\}$ is a sequence of steps sizes. We next embed the dynamical iterations $\mathbf{y}_k(\mathbf{x},\mathbf{z})$ into $\max_{1\le k\le K}\{F(\mathbf{x},\mathbf{y}_k(\mathbf{x},\mathbf{z}))\}$, which can be regarded as a pessimistic trajectory truncation of the UL objective. The mechanism of the above scheme, in comparison, accesses the approximation parameterized with UL variable $\mathbf{x}$ and LL initial point $\mathbf{y}_0$. Our motivation for the initialization auxiliary variable $\mathbf{z}$ comes from convergence theory of non-convex first-order optimization methods. In fact, when the non-convex LL problem admits multiple solutions, the gradient descent steps with a "bad" initial point $\mathbf{y}_0$ cannot return a desired point in the LL solution set, simultaneously optimizing the UL objective. To overcome such a difficulty, instead of using a fixed initial value, we introduce an initialization auxiliary variable $\mathbf{z}$. Therefore, when it comes to solving the UL approximation problems, together with the UL variable $\mathbf{x}$, the auxiliary variable $\mathbf{z}$ is also updated and hence optimized. As a consequence, we may search for the "best" initial value, starting from which the gradient descent steps approach a solution to the BLO in Eq. (1), i.e., a point in the LL solution set, simultaneously minimizing the UL objective.

Pessimistic Trajectory Truncation. This is a striking feature of our algorithm that differs significantly from existing methods and leads to some new convergence results without LLC. The motivation for the design of pessimistic trajectory truncation comes again from convergence theory of non-convex first-order optimization methods. It is understood that when LL is non-convex, $\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y}_K(\mathbf{x}, \mathbf{z}))$ may not uniformly converge w.r.t. $\mathbf{x}$ and $\mathbf{z}$, where $\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y})$ is the proximal gradient residual mapping defined as $\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y}) := \mathbf{y} - \mathtt{Proj}_{\mathcal{Y}}(\mathbf{y} - \alpha \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}))$, [1] which can be used as a measurement of the optimality of LL problem in Eq. (2). Thus a direct embedding of $\mathbf{y}_K(\mathbf{x}, \mathbf{z})$ into UL objective $F(\mathbf{x}, \mathbf{y})$ may not necessarily provide an appropriate approximation. Fortunately, it is also understood that, for each $\mathbf{x} \in \mathcal{X}$, $\mathbf{z} \in \mathcal{Y}$ and $K > 0$, there exists at least a $\tilde{K}$ such that along the selection $\mathbf{y}_{\tilde{K}}(\mathbf{x}, \mathbf{z})$, $\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y}_{\tilde{K}}(\mathbf{x}, \mathbf{z}))$ uniformly converges to zero w.r.t. $\mathbf{x}$ and $\mathbf{z}$, as K tending infinity. However, in general, it is too ambitious to expect an explicit identification of the exact selection $\mathbf{y}_{\tilde{K}}(\mathbf{x}, \mathbf{z})$. Alternatively, we consider a pessimistic strategy, minimizing the worst case of all selections of $\{\mathbf{y}_k(\mathbf{x}, \mathbf{z})\}$, i.e., $\max_{1 \le k \le K} \{F(\mathbf{x}, \mathbf{y}_k(\mathbf{x}, \mathbf{z}))\}$. By doing so, we successfully reach a good approximation. In addition to the theoretical convergence, we also benefit from this pessimistic strategy in a numerical sense. The pessimistic $\max$ operation always results in a favorable trajectory truncation smaller than K. Consequently, this technique offers inexpensive computational cost for computing the hyper-gradient through back propagation, as shown in the numerical experiments.

To conclude this section, we state the complete IAPTT-GM in Algorithm 1.

Algorithm 1 The Proposed IAPTT-GM

Initialize $\mathbf{x}^0$ and $\mathbf{z}^0$.
for $t = 0 \to T-1$ **do**
  $\mathbf{y}_0 = \mathbf{z}^t$.
  for $k = 0 \to K-1$ **do**
    % LL Updating with $\mathbf{x}^t$ and $\mathbf{z}^t$
    $\mathbf{y}_{k+1} = \mathtt{Proj}_{\mathcal{Y}}(\mathbf{y}_k - \alpha_{\mathbf{y}}^k \nabla_{\mathbf{y}} f(\mathbf{x}^t, \mathbf{y}_k))$.
  end for
  % Pessimistic Trajectory Truncation
  $\bar{k} = \arg\max_k \{F(\mathbf{x}, \mathbf{y}_k)\}_{k=1}^K$.
  % UL Updating with $\mathbf{y}_{\bar{k}}(\mathbf{x}, \mathbf{z})$
  $\mathbf{x}^{t+1} = \mathtt{Proj}_{\mathcal{X}}(\mathbf{x}^t - \alpha_{\mathbf{x}} \nabla_{\mathbf{x}} F(\mathbf{x}^t, \mathbf{y}_{\bar{k}}))$.
  % Initialization Updating with $\mathbf{y}_{\bar{k}}(\mathbf{x}, \mathbf{z})$
  $\mathbf{z}^{t+1} = \mathtt{Proj}_{\mathcal{Y}}(\mathbf{z}^t - \alpha_{\mathbf{z}} \nabla_{\mathbf{z}} F(\mathbf{x}^t, \mathbf{y}_{\bar{k}}))$.
end for

3 Theoretical Investigations

With the purpose of studying the convergence of dynamics-embedded gradient method for BLO without LLC, we involve two signature features in our algorithmic design, i.e., initialization auxiliary and pessimistic trajectory truncated. This section is devoted to the convergence analysis of our proposed algorithm with and without LLC assumption. Please notice that all the proofs of our theoretical results are stated in the Supplemental Material.

3.1 Convergence Analysis of IAPTT-GM for BLO with Non-convex Followers

In this part, we conduct the convergence analysis of the IAPTT-GM for solving BLO in Eq. (1) without LLC. Before presenting our main convergence results, we introduce some notations related to BLO. With introduced function $\varphi(\mathbf{x}) := \inf_{\mathbf{y} \in \mathcal{S}(\mathbf{x})} F(\mathbf{x}, \mathbf{y})$, the BLO in Eq. (1) can be rewritten as

$$\min_{\mathbf{x} \in \mathcal{X}} \varphi(\mathbf{x}). \tag{5}$$

With given $K \ge 1$ and defining $\varphi_K(\mathbf{x}, \mathbf{z}) := \max_k \{F(\mathbf{x}, \mathbf{y}_k(\mathbf{x}, \mathbf{z}))\}$ with $\{\mathbf{y}_k(\mathbf{x}, \mathbf{z})\}$ defined in Eq. (4), our proposed IAPTT-GM is generating sequence $\{(\mathbf{x}^t, \mathbf{z}^t)\}$ for solving following approximation problem to BLO in Eq. (5),

$$\min_{\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Y}} \varphi_K(\mathbf{x}, \mathbf{z}). \tag{6}$$

This section is mainly devoted to the convergence of solutions of approximation problems in Eq. (6) towards those of the original BLO in Eq. (5).

Assumption 3.1 *We make following standing assumptions throughout this section.*

(1) $F, f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are continuous functions.

(2) ∇f is continuous and $\nabla_{\mathbf{y}} f$ is L_f Lipschitz continuous with respet to $\mathbf{y}$ for any $\mathbf{x} \in \mathcal{X}$.

[1] $\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y})$ reduces to $\alpha \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$ when $\mathcal{Y}$ is taken as $\mathbb{R}^m$.

(3) $\mathcal{X}$ and $\mathcal{Y}$ are convex compact sets.

(4) $\mathcal{S}(\mathbf{x})$ is nonempty for any $\mathbf{x} \in \mathcal{X}$.

(5) For any $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ minimizing $F(\mathbf{x}, \mathbf{y})$ over constraints $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}$ and $\mathbf{y} \in \hat{\mathcal{S}}(\mathbf{x})$, it holds that $\bar{\mathbf{y}} \in \mathcal{S}(\mathbf{x})$.

Note that $\hat{\mathcal{S}}(\mathbf{x})$ denotes the set of LL stationary points, i.e., $\hat{\mathcal{S}}(\mathbf{x}) = \{\mathbf{y} \in \mathcal{Y} | 0 = \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}) + \mathcal{N}_{\mathcal{Y}}(\mathbf{y})\}$. It should be noticed that $\mathbf{y} \in \hat{\mathcal{S}}(\mathbf{x})$ if and only if $\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y}) = 0$. Assumption 3.1 is standard in bi-level optimziation related literature, which will be shown to be satisfied for the numerical example given in Section 4.1.

As discussed in the preceding section, for each $\mathbf{x} \in \mathcal{X}$, $\mathbf{z} \in \mathcal{Y}$ and $K > 0$, there exists at least a $\tilde{K}$ such that along the selection $\mathbf{y}_{\tilde{K}}(\mathbf{x}, \mathbf{z})$, $\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y}_{\tilde{K}}(\mathbf{x}, \mathbf{z}))$ uniformly converges to zero w.r.t. $\mathbf{x}$ and $\mathbf{z}$, as K tending inifinity. We next specifically show the existence of such index $\tilde{K}$. To this end, $\tilde{K}$ can be chosen by optimizing $\|\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y}_k(\mathbf{x}, \mathbf{z}))\|$ among the indices $k = 0, 1, ..., K$. In particular, as stated in the following lemma, $\|\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y}_{\tilde{K}}(\mathbf{x}, \mathbf{z}))\|$ uniformly decreases with a $\frac{1}{\sqrt{K}}$ rate on $\mathcal{X} \times \mathcal{Y}$ as K increases.

Lemma 3.1 *Let $\{\mathbf{y}_k(\mathbf{x}, \mathbf{z})\}$ be the sequence defined in Eq.* (4) *with $\alpha_{\mathbf{y}}^k \in [\underline{\alpha}_{\mathbf{y}}, \overline{\alpha}_{\mathbf{y}}] \subset (0, \frac{2}{L_f})$, there exists $C_f > 0$ such that*

$$\min_{0 \le k \le K} \|\mathcal{R}_{\underline{\alpha}_{\mathbf{y}}}(\mathbf{x}, \mathbf{y}_k(\mathbf{x}, \mathbf{z}))\| \le \frac{C_f}{\sqrt{K+1}}, \quad \forall \mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Y}. \tag{7}$$

As shown in the Appendix, the proof of Lemma 3.1 for the existence cannot offer us an explicit identification of the exact selection of $\tilde{K}$. Alternatively, we construct the approximation by a pessimistic trajectory truncation strategy, minimizing the worst case of all selections of $\{\mathbf{y}_k(\mathbf{x}, \mathbf{z})\}$, i.e., $\varphi_K(\mathbf{x}, \mathbf{z})$. By further solving the approximated problems $\min \varphi_K(\mathbf{x}, \mathbf{z})$, we shall provide an lower bound estimation for the optimal value of the BLO problem in Eq. (5).

Lemma 3.2 *Let $(\mathbf{x}_K, \mathbf{z}_K) \in \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Y}} \varphi_K(\mathbf{x}, \mathbf{z})$, then*

$$\varphi_K(\mathbf{x}_K, \mathbf{z}_K) \le \inf_{\mathbf{y} \in \hat{\mathcal{S}}(\mathbf{x})} F(\mathbf{x}, \mathbf{y}), \quad \forall \mathbf{x} \in \mathcal{X}. \tag{8}$$

Upon together with the uniform convergence result in Lemma 3.1, the gap between the lower bound provided by $\min \varphi_K(\mathbf{x}, \mathbf{z})$ and the true optimal value of the BLO problem in Eq. (5) eventually vanishes. To fill in this gap and present the main convergence result of our proposed IAPTT-GM, we need the continuity of $\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y})$. Indeed, it follows from [37, Theorem 6.42] that $\texttt{Proj}_{\mathcal{Y}}$ is continuous. Combined with the assumed continuity of $\nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$, we get the desired continuity of $\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y})$ immediately.

Theorem 3.1 *Let $\{\mathbf{y}_k(\mathbf{x}, \mathbf{z})\}$ be the sequence generated by Eq.* (4) *with $\alpha_{\mathbf{y}}^k \in [\underline{\alpha}_{\mathbf{y}}, \overline{\alpha}_{\mathbf{y}}] \subset (0, \frac{2}{L_f})$, and $(\mathbf{x}_K, \mathbf{z}_K) \in \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Y}} \varphi_K(\mathbf{x}, \mathbf{z})$, then we have:*

(1) Any limit point $\bar{\mathbf{x}}$ of the sequence $\{\mathbf{x}_K\}$ is the solution to BLO in Eq. (1)*, that is $\bar{\mathbf{x}} \in \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \varphi(\mathbf{x})$.*

(2) $\inf_{\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Y}} \varphi_K(\mathbf{x}, \mathbf{z}) \to \inf_{\mathbf{x} \in \mathcal{X}} \varphi(\mathbf{x})$ as $K \to \infty$.

3.2 Theoretical Findings of IA-GM(A) for BLO with LLC

A byproduct of our study, which has its own interest, is that thanks to the involved initialization auxiliary, our theory can improve those existing results for classical gradient methods with accelerated gradient descent dynamical iterations under LLC. Nesterov's acceleration technique [38] has been used widely for solving convex optimization problem and it greatly improves the convergence rate of gradient descent. To illustrate our result, we will take the Nesterov's acceleration proximal gradient method [39] as the embedded optimization dynamics in classical gradient method for example.

Thanks to the LLC setting, $\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y}_K(\mathbf{x}, \mathbf{z}))$ may uniformly converge to zero w.r.t. UL variable x and auxiliary variable $\mathbf{z}$, thus the pessimistic trajectory truncation operation can be removed. Subsequently, we slightly simplify our algorithm that $\bar{k}$ is simply taken as K, thus the approximation objective admits a succinct form, i.e., $F(\mathbf{x}, \mathbf{y}_K(\mathbf{x}, \mathbf{z}))$.

In summary, by constructing $\mathbf{y}_k(\mathbf{x}, \mathbf{z})$ through following Nesterov's acceleration dynamics,

$$\begin{aligned}
&\mathbf{y}_0(\mathbf{x}, \mathbf{z}) = \mathbf{z}, \qquad t_0 = 1, \quad t_{k+1} = \frac{1 + \sqrt{1 + t_k^2}}{2},\ k = 0, \cdots, K-1, \\
&\mathbf{u}^{k+1}(\mathbf{x}, \mathbf{z}) = \mathbf{y}^{k+1}(\mathbf{x}, \mathbf{z}) + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{y}^{k+1}(\mathbf{x}, \mathbf{z}) - \mathbf{y}^k(\mathbf{x}, \mathbf{z})),\ k = 0, \cdots, K-1, \\
&\mathbf{y}_{k+1}(\mathbf{x}, \mathbf{z}) = \mathtt{Proj}_{\mathcal{Y}}\left(\mathbf{u}^k(\mathbf{x}, \mathbf{z}) - \alpha \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{u}^k(\mathbf{x}, \mathbf{z}))\right),\ k = 0, \cdots, K-1,
\end{aligned} \tag{9}$$

where $\alpha > 0$ is the stepsize, we propose an accelerated Gradient-based Method with Initialization Auxiliary, named IA-GM(A), via minimizing the approximation objective function,

$$\min_{\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Y}} \phi_K(\mathbf{x}, \mathbf{z}) := F(\mathbf{x}, \mathbf{y}_K(\mathbf{x}, \mathbf{z})).$$

The detailed description of the proposed IA-GM(A) are stated in the Supplemental Material.

We suppose Assumption 3.1(1)-(4) are satisfied throughout this subsection. The convergence result of our proposed IA-GM(A) in LLC case is given as below.

Theorem 3.2 *Assume that the generated sequence $\{\mathbf{y}_k(\mathbf{x}, \mathbf{z})\}$ satisfies that $\mathbf{y}_k(\mathbf{x}, \mathbf{z}) \in \mathcal{Y}$, and $\mathbf{y}_k(\mathbf{x}, \mathbf{z}) = \mathbf{z}$ for any $\mathbf{z} \in \mathcal{S}(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$, and either*

(a) for any $\epsilon > 0$, there exists $k(\epsilon) > 0$ such that whenever $K > k(\epsilon)$,

$$\sup_{\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Y}} \{f(\mathbf{x}, \mathbf{y}_K(\mathbf{x}, \mathbf{z})) - f^*(\mathbf{x})\} \leq \epsilon$$

whenever $K > k(\epsilon)$, or

(b) there exists $\alpha > 0$, for any $\epsilon > 0$, there exists $k(\epsilon) > 0$ such that,

$$\sup_{\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Y}} \|\mathcal{R}_\alpha(\mathbf{x}, \mathbf{y}_K(\mathbf{x}, \mathbf{z}))\| \leq \epsilon.$$

Let $(\mathbf{x}_K, \mathbf{z}_K) \in \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Y}} \phi_K(\mathbf{x}, \mathbf{z}) := F(\mathbf{x}, \mathbf{y}_K(\mathbf{x}, \mathbf{z}))$, then we have

(1) any limit point $\bar{x}$ of the sequence $\{\mathbf{x}_K\}$ satisfies that $\bar{\mathbf{x}} \in \arg\min_{\mathbf{x} \in \mathcal{X}} \varphi(\mathbf{x})$, i.e., $\bar{x}$ is the solution to BLO (1).

(2) $\inf_{\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Y}} \phi_K(\mathbf{x}, \mathbf{z}) \to \inf_{\mathbf{x} \in \mathcal{X}} \varphi(\mathbf{x})$ as $K \to \infty$.

Next, we show that the Nesterov's acceleration dynamics satisfy all the assumptions required in the above convergence theorem.

Theorem 3.3 *Let $\{\mathbf{y}_k(\mathbf{x}, \mathbf{z})\}$ be the sequence generated by Nesterov's acceleration dynamics in Eq.* (9) *with $\alpha = \frac{1}{L_f}$.Then $\{\mathbf{y}_k(\mathbf{x}, \mathbf{z})\}$ satisfies all the assumptions required by Theorem 3.2.*

Remark 3.1 *Our convergence result Theorem 3.2 is not only for $\mathbf{y}_k$ generated by Nesterov's acceleration dynamics. It is a general convergence result that is applicable for the case where $\mathbf{y}_k$ is generated by other dynamics. And the assumptions required in Theorem 3.2 is weak enough to be satisfied by the dynamics introduced by many first-order methods on convex LL problem in Eq. 2.*

4 Experimental Results

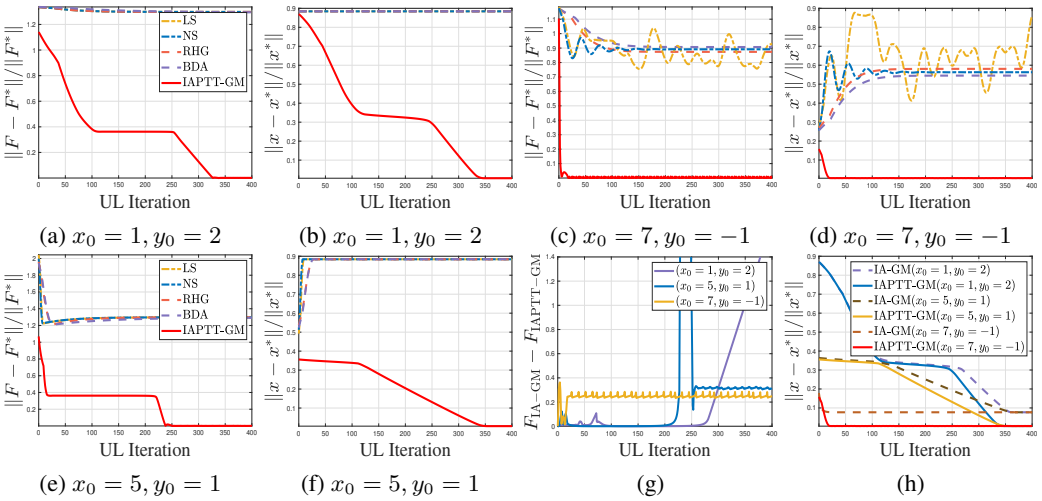
In this section, we first verify the theoretical convergence results on non-convex toy problems compared with existing EG methods and IG methods. Then we test the performance of IAPTT-GM and demonstrate its generalizability to real-world BLO problems with non-convex followers, which are caused by non-convex regularization terms and neural network structures. In addition, we further validate the performance of IA-GM (A) under LLC with toy examples and data hyper-cleaning.

4.1 Numerical Verification

To verify the convergence property under assumptions provided in Section 3, we consider the following non-convex BLO problem:

$$\min_{x\in\mathcal{X},y\in\mathbb{R}} x + xy, \quad s.t. \quad y \in \operatorname*{argmin}_{y\in\mathcal{Y}} -\sin(xy), \tag{10}$$

where $\mathcal{X} = [1, 10]$ and $\mathcal{Y} = [-2, 2]$. Given any $x \in \mathcal{X}$, it satisfies $\operatorname{argmin}_{y\in\mathcal{Y}} -\sin(xy) = \{(2k\pi + \pi/2)/x \mid k \in \mathbb{Z}\} \cap \mathcal{Y}$ and $\min_{y\in\mathcal{Y}} -\sin(xy) = -1$. The unique solution is $(x^*, y^*) = (11\pi/4, -2)$. It should be noted that the LL problem of Eq. (10) has multiple global minima, which can significantly show the advantage of initialization auxiliary technique. It can be easily verified that the above toy example satisfies Assumption 3.1.



(a) $x_0 = 1, y_0 = 2$ (b) $x_0 = 1, y_0 = 2$ (c) $x_0 = 7, y_0 = -1$ (d) $x_0 = 7, y_0 = -1$

(e) $x_0 = 5, y_0 = 1$ (f) $x_0 = 5, y_0 = 1$ (g) (h)

Figure 1: Illustrating the convergence behavior of $\|F - F^*\|/\|F^*\|$ and $\|x - x^*\|/\|x^*\|$ as the training proceeds. $F_{\mathrm{IAPTT-GM}}$ and $F_{\mathrm{IA-GM}}$ denote the UL objectives of IATT-GM and IA-GM, respectively. Three representative initialization points for UL and LL variables are $(x_0, y_0) = (1, 2)$, $(x_0, y_0) = (5, 1)$, $(x_0, y_0) = (7, -1)$.

In Figure 1, we separately compared IAPTT-GM with EG methods such as RHG [2], BDA [21], IG methods such as LS [25], NS [6] and IA-GM. From Figure 1.(a) to Figure 1.(b), we can observe that different initialization points only slightly affect the convergence speed of IAPTT-GM. With initialization points distant from (x^*, y^*), IAPTT-GM can still achieve optimal solution of UL variables and optimal objective value, while other methods fail to converge to the true solution. In Figure 1.(g) and Figure 1. (h), we compare the performances of IAPTT-GM with IA-GM, which has no convergence guarantee without LLC assumption. As shown, IA-GM fails to converge to the true solution eventually, which validates the necessity of PTT technique and the effectiveness of IAPTT-GM.

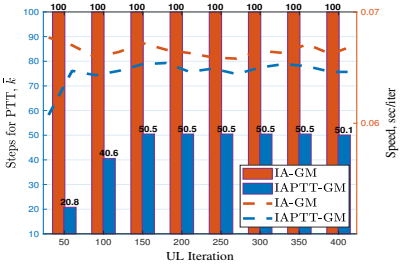


Figure 2: Illustrating mean steps $\bar{k}$ for PTT and mean speed with respect to the UL iteration. Note that we slow down the optimization iteration using larger K so as to reduce the error.

Furthermore, in Figure 2, we report the mean steps $\bar{k}$ of IAPTT-GM for PPT and iterative speed as the UL iteration increases. In comparison with IA-GM, which uses default K for LL optimization loop, changing $\bar{k}$ in IAPTT-GM lead to less iterations for backward recurrent propagation and thus faster iterative updates.

4.2 BLO with Non-convex Followers in Different Application Scenarios

To cover various real-world BLO application scenarios, we consider two categories of non-convex LL problems caused by non-convex regularization and neural network structures, which refers to few-shot classification and data hyper-cleaning, respectively.

4.2.1 Few-Shot Classification: Non-convex LL Objective

In few-shot learning, to be specific, N-way M-shot classification tasks[40], provided with M samples from each class, we train models to task advantage of prior data from similar tasks to quickly classify unseen instances from these N classes. Following the experimental protocol [41], the model parameters are separated into two parts: the hyper representation module (parameterized by $\mathbf{x}$) shared by all the tasks and the last classifier (parameterized by $\mathbf{y}^j$) for j-th task. Define the meta training dataset as $\mathcal{D} = \{\mathcal{D}^j\}$, where $\mathcal{D}^j = \mathcal{D}^j_{\mathtt{tr}} \bigcup \mathcal{D}^j_{\mathtt{val}}$ corresponds to the j-th task.

The cross-entropy loss function is widely used for UL and LL objectives. Referring to Section 3, IATT-GM covers the convergence results with non-convex LL model, thus allowing flexible design of the LL objective function. For instance, while non-convex regularization terms, e.g., ℓ_q regularization with $0 < q < 1$, has shown effectiveness to help the LL model converge and avoid overfitting, existing methods can only guarantee the convergence when $q \geq 1$, thus almost provide no support for non-convex objectives. We consider the LL subproblem with non-convex loss functions by adding ℓ_q regularization. [2]. Then the UL and LL subproblems can be written as

$$F\left(\mathbf{x}, \{\mathbf{y}^j\}\right) = \sum_j \ell\left(\mathbf{x}, \mathbf{y}^j; \mathcal{D}^j_{\mathtt{val}}\right), \quad f\left(\mathbf{x}, \{\mathbf{y}^j\}\right) = \sum_j \ell\left(\mathbf{x}, \mathbf{y}^j; \mathcal{D}^j_{\mathtt{tr}}\right) + \|\mathbf{y}^j\|_q. \tag{11}$$

Table 1: Mean test accuracy of 5-way classification on tieredImageNet and miniImageNet, and the $\pm$ represents 95% confidence intervals over tasks.

Methods	Backbone	MiniImagenet		TieredImagenet	
		5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
Proto Net [42]	ConvNet-4	49.42 ± 1.84	68.20 ± 0.66	53.31 ± 0.89	72.69 ± 0.74
Relation Net [43]	ConvNet-4	50.44 ± 0.82	65.32 ± 0.70	54.48 ± 0.93	65.32 ± 0.70
MAML [28]	ConvNet-4	48.70 ± 0.75	63.11 ± 0.11	49.06 ± 0.50	67.48 ± 0.47
RHG	ConvNet-4	48.89 ± 0.81	63.02 ± 0.70	49.63 ± 0.67	66.14 ± 0.57
T-RHG	ConvNet-4	47.67 ± 0.82	63.70 ± 0.76	50.79 ± 0.69	67.39 ± 0.60
BDA	ConvNet-4	49.08 ± 0.82	62.17 ± 0.70	51.56 ± 0.68	68.21 ± 0.58
MAML	ResNet-12	51.03 ± 0.50	68.26 ± 0.47	58.58 ± 0.49	71.24 ± 0.43
RHG	ResNet-12	50.54 ± 0.85	64.53 ± 0.68	58.19 ± 0.76	75.20 ± 0.60
IAPTT-GM	ResNet-12	$\mathbf{56.69 \pm 0.66}$	$\mathbf{70.21 \pm 0.55}$	$\mathbf{60.71 \pm 0.77}$	$\mathbf{75.85 \pm 0.59}$

Detailed information about the datasets and network architectures are stated in the Appendix. We report results of IAPTT-GM and state-of-the-art methods on miniImageNet [41] and tieredImageNet [44] datasets with two different backbones [5, 45] in Table 1. As it is shown, our proposed method outperforms state-of-the-art methods on both 5-way 1-shot and 5-way 5-shot tasks.

4.2.2 Data Hyper-Cleaning: Non-convex LL Architecture Structure

Data hyper-cleaning [2] aims to cleanup the corrupted data with noise label. According to [14], the dataset is randomly split to three disjoint subsets: $\mathcal{D}_{\mathtt{tr}}$ for training, $\mathcal{D}_{\mathtt{val}}$ for validation and $\mathcal{D}_{\mathtt{test}}$ for test, then a fixed proportion of the training samples in $\mathcal{D}_{\mathtt{tr}}$ is randomly corrupted.

Following the classical experimental protocol [2], we choose cross-entroy as the loss function ℓ ,and the UL and LL subproblem take the form of

$$F(\mathbf{x}, \mathbf{y}) = \sum_{(\mathbf{u}_i, \mathbf{v}_i) \in \mathcal{D}_{\mathtt{val}}} \ell\left(\mathbf{y}(\mathbf{x}); \mathbf{u}_i, \mathbf{v}_i\right), \quad f(\mathbf{x}, \mathbf{y}) \quad = \sum_{(\mathbf{u}_i, \mathbf{v}_i) \in \mathcal{D}_{\mathtt{tr}}} [\sigma(\mathbf{x})]_i \ell\left(\mathbf{y}; \mathbf{u}_i, \mathbf{v}_i\right), \tag{12}$$

[2]Here, we use the smoothing regularization term defined as $\|\boldsymbol{w}\|_q = (\|\boldsymbol{w}\|_2 + \|\boldsymbol{\varepsilon}\|_2)^{q/2}$, where $0 < q < 1$.

where $(\mathbf{u}_i, \mathbf{v}_i)$ denotes the data pair and $\sigma(\mathbf{x})$ represents the element-wise sigmoid function on $\mathbf{x}$. We define the hyperparameter $\mathbf{x}$ as a vector trained to label the noisy data, of which the dimension equals to the number of training samples. The LL variables parameterized by $\mathbf{y}$ contain the weights and bias of fully connected layers. Existing methods consider convex LL model by simply using single fully connected layer, while more complex neural networks structure is not applicable. Under our assumption without LLC, we employ two fully connected layers as the LL network architecture. In Table 2, we compare IAPTT-GM with EG methods and IG methods with the same non-convex network structures. As shown, IAPTT-GM gains better test performance of both accuracy and F1 score on two datasets. Our theoretical results also shows that the improvement of performance IAPTT-GM come from PTT and IA techniques to overcome nonconvex LL subproblems.

Table 2: Reporting results of existing methods for solving data hyper-cleaning tasks on MNIST [40] and FashionMNIST [46]. Acc. and F1 score denote the test accuracy and the harmonic mean of the precision and recall, respectively.

Method	MNIST		FashionMNIST	
	Acc.	F1 score	Acc.	F1 score
LS	89.19	85.96	83.15	85.13
NS	87.54	89.58	81.37	87.28
RHG	87.90	89.36	81.91	87.12
T-RHG	88.57	89.77	81.85	86.76
BDA	87.15	90.38	79.97	88.24
IAPTT-GM	**90.88**	**91.57**	**83.67**	**90.37**

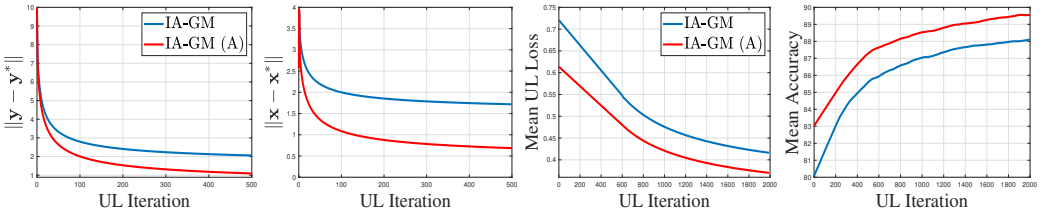
4.3 Evaluations of IA-GM (A) for BLOs under LLC

In addition to non-convex BLO problems, we also raise concerns about the acceleration strategy of our method in LLC cases. We first consider the following LLC BLO problem [21]:

$$\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}-\mathbf{y}_2\|^4 + \|\mathbf{y}_1-\mathbf{e}\|^4, \quad s.t. \quad (\mathbf{y}_1,\mathbf{y}_2) \in \arg\min_{\mathbf{y}_1\in\mathbb{R}^n,\mathbf{y}_2\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y}_1\|^2 - \mathbf{x}^\top\mathbf{y}_1, \tag{13}$$

where $n = 50$, $\mathcal{X} = [-100, 100] \times \cdots [-100, 100] \subset \mathbb{R}^n$, and $\mathbf{e}$ represents the vector whose elements are all equal to 1. The optimal solution for this problem is $\mathbf{x}^* = \mathbf{e}, \mathbf{y}_1^* = \mathbf{e}, \mathbf{y}_2^* = \mathbf{e}$.

As illustrated in Section 3.2, our method IA-GM(A) incorporates Nesterov's acceleration strategy for solving LLP. Note that with LLC assumption, IA-GM maintains the convergence property.



(a) Convex toy regression (b) Convex toy regression (c) Data hyper-cleaning (d) Data hyper-cleaning

Figure 3: The left two figures report the curves of $\|\mathbf{y}-\mathbf{y}^*\|$ and $\|\mathbf{x}-\mathbf{x}^*\|$ for IA-GM and IA-GM (A). The figures on the right illustrate the results of mean UL loss and accuracy on the convex data hyper-cleaning problems.

As illustrated in the first two figures in Figure 3, IA-GM (A) shows significant improvement of convergence speed on UL and LL variables, which verifies the convergence results of Theorem 3.3 under LLC. We further studied the convex data hyper-cleaning problem, which simply implements single fully connected layer as the network structure. From the right half of Figure 3, we can easily find that IA-GM(A) also performs better than IA-GM on real-world applications.

5 Conclusion

This paper presents a generic first-order algorithmic framework named IAPTT-GM to solve BLO problems with non-convex follower. We introduce two features, initialization auxiliary and pessimistic trajectory truncation operation to guarantee the convergence without the LLC hypothesis and achieves better performance on various applications. Meanwhile, we also validate the performance and speed of our improvement on LLC cases.

References

[1] Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *arXiv preprint arXiv:2101.11517*, 2021.

[2] L Franceschi, M Donini, P Frasconi, and M Pontil. Forward and reverse gradient-based hyperparameter optimization. In *ICML*, volume 70, pages 1165–1173. JMLR, 2017.

[3] Takayuki Okuno, Akiko Takeda, and Akihiro Kawana. Hyperparameter learning via bilevel nonsmooth optimization. *arXiv preprint arXiv:1806.01520*, 2018.

[4] Matthew Mackay, Paul Vicol, Jonathan Lorraine, David Duvenaud, and Roger Grosse. Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions. In *ICLR*, 2018.

[5] L Franceschi, P Frasconi, S Salzo, R Grazzi, and M Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *ICML*, volume 80, pages 1563–1572, 2018.

[6] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *NeurIPS*, pages 113–124, 2019.

[7] Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *ICLR*, 2018.

[8] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *ICLR*, 2018.

[9] Hanwen Liang, Shifeng Zhang, Jiacheng Sun, Xingqiu He, Weiran Huang, Kechen Zhuang, and Zhenguo Li. Darts+: Improved differentiable architecture search with early stopping. *arXiv preprint arXiv:1909.06035*, 2019.

[10] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *ICCV*, pages 1294–1303, 2019.

[11] David Pfau and Oriol Vinyals. Connecting generative adversarial networks and actor-critic methods. *arXiv preprint arXiv:1610.01945*, 2016.

[12] Zhuoran Yang, Yongxin Chen, Mingyi Hong, and Zhaoran Wang. Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. In *NeurIPS*, pages 8353–8365, 2019.

[13] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR, 2015.

[14] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732. PMLR, 2019.

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

[16] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(2):1, 2018.

[17] Ke Li and Jitendra Malik. Learning to optimize. *arXiv preprint arXiv:1606.01885*, 2016.

[18] Yoonho Lee and Seungjin Choi. Meta-learning with adaptive layerwise metric and subspace. 2018.

[19] Eunbyung Park and Junier B Oliva. Meta-curvature. *arXiv preprint arXiv:1902.03356*, 2019.

[20] Sebastian Flennerhag, Andrei A Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. *arXiv preprint arXiv:1909.00025*, 2019.

[21] Risheng Liu, Pan Mu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. In *ICML*, pages 6305–6315. PMLR, 2020.

[22] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Nonasymptotic analysis and faster algorithms. *arXiv preprint arXiv:2010.07962v2*, 2020.

[23] Riccardo Grazzi, Saverio Salzo, Massimiliano Pontil, and Luca Franceschi. On the iteration complexity of hypergradient computations. In *ICML 2020: 37th International Conference on Machine Learning*, volume 1, pages 3748–3758, 2020.

[24] Kaiyi Ji and Yingbin Liang. Lower bounds and accelerated algorithms for bilevel optimization. *arXiv preprint arXiv:2102.03926*, 2021.

[25] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *ICML*, pages 737–746, 2016.

[26] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1552. PMLR, 2020.

[27] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[28] 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.

[29] Pan Zhou, Xiaotong Yuan, Huan Xu, Shuicheng Yan, and Jiashi Feng. Efficient meta learning via minibatch proximal update. *Advances in Neural Information Processing Systems*, 32: 1534–1544, 2019.

[30] Liam Collins, Aryan Mokhtari, and Sanjay Shakkottai. Task-robust model-agnostic meta-learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[31] Ren Wang, Kaidi Xu, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Chuang Gan, and Meng Wang. On fast adversarial robustness adaptation in model-agnostic meta-learning. *arXiv preprint arXiv:2102.10454*, 2021.

[32] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Generalization of model-agnostic meta-learning algorithms: Recurring and unseen tasks. *arXiv preprint arXiv:2102.03832*, 2021.

[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*, 2019.

[34] 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.

[35] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Theoretical convergence of multi-step model-agnostic meta-learning. *arXiv e-prints*, pages arXiv–2002, 2020.

[36] Kaiyi Ji, Jason D Lee, Yingbin Liang, and H. Vincent Poor. Convergence of meta-learning with task-specific adaptation over partial parameters. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11490–11500. Curran Associates, Inc., 2020.

[37] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[38] Yurii E. Nesterov. A method for solving the convex programming problem with convergence rate O(1/kˆ2). *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.

[39] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *Siam Journal on Imaging Sciences*, 2(1):183–202, 2009.

[40] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[41] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *NeurIPS*, 29:3630–3638, 2016.

[42] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[43] 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.

[44] 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.

[45] Boris N Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *arXiv preprint arXiv:1805.10123*, 2018.

[46] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Please refer to Section 1.1 for detailed description corresponding to the abstract.
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Please refer to Section 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] We put most of the detailed calculation process in the supplementary materials.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Instruction for the code, datasets are provided in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Information about the dataset and hyperparameters for numerical experiments can be found in the supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We specify the detailed configuration of computing devices, GPUs in the supplementary materials.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We conduct all the experiments based on public data.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]