Multi-Objective Bilevel Learning

Zhiyao Zhang[1], Zhuqing Liu[2], Xin Zhang[3], Wen-Yen Chen[3], Jiyan Yang[3], Jia Liu[1]

[1]Dept. of Electrical and Computer Engineering, The Ohio State University
[2]Dept. of Computer Science, University of North Texas
[3]Meta Platforms, Inc.

Abstract

As machine learning (ML) applications grow increasingly complex in recent years, modern ML frameworks often need to address multiple potentially conflicting objectives with coupled decision variables across different layers. This creates a compelling need for multi-objective bilevel learning (MOBL). So far, however, the field of MOBL remains in its infancy and many important problems remain under-explored. This motivates us to fill this gap and systematically investigate the theoretical and algorithmic foundation of MOBL. Specifically, we consider MOBL problems with multiple conflicting objectives guided by preferences at the upper-level subproblem, where part of the inputs depend on the optimal solution of the lower-level subproblem. Our goal is to develop efficient MOBL optimization algorithms to (1) identify a preference-guided Pareto-stationary solution with low oracle complexity; and (2) enable systematic Pareto front exploration. To this end, we propose a unifying algorithmic framework called weighted-Chebyshev multi-hyper-gradient-descent (WC-MHGD) for both deterministic and stochastic settings with finite-time Pareto-stationarity convergence rate guarantees, which not only implies low oracle complexity but also induces systematic Pareto front exploration. We further conduct extensive experiments to confirm our theoretical results.

1 Introduction

Background and motivation: As machine learning (ML) applications grow increasingly complex, modern ML training frameworks often need to address *multiple objectives* that are potentially conflicting. As an example, in the design of online shopping recommender systems, the objective of a training loss function could aim to prioritize lower prices, more popular brand names, or faster delivery speeds, all of which may be in conflict with each other. Moreover, such objective functions could further depend on an optimal solution from a separate and coupled optimization task, thus making the overall training task *bilevel* in nature. For instance, the multi-objective trainings in the aforementioned online shopping recommender system could all rely on a joint pre-training of a set of shared model parameters that define all objectives. Similar multi-objective bilevel problem structures also arise from multi-agent reinforcement learning with an actor-critic framework (Ji, Yang, and Liang 2021; Yang, Ji, and Liang 2021; Zhang et al. 2024; Qiu et al. 2023) and the pretraining-finetuning pipeline of multi-task adaptation of a foundation model (Chakraborty et al. 2024; Shen, Yang, and Chen 2024). These complex problem structures create a compelling need for multi-objective bilevel learning (MOBL).

Mathematically, an MOBL problem can be formulated in the following general form:

$$\begin{aligned} &\min_{x\in\mathbb{R}^p} \Phi(x) := [\phi_1(x), \dots, \phi_S(x)] \\ &\text{s.t. } y^*(x) \in \arg\min_{y\in\mathbb{R}^q} g(x,y), \end{aligned} \tag{1}$$

where $\phi_s(x) := f^{(s)}(x, y^*(x))$, $s \in [S]$, represents an upper-level (UL) objective function with an implicit UL decision variable $y^*(x)$ being an optimizer of a lower-level (LL) subproblem. Clearly, MOBL combines the problem structures of multi-objective optimization (MOO) and bilevel optimization (BLO), both of which are challenging ML optimization paradigms in their own right and have attracted a significant amount of attention in the research community in recent years. Although numerous algorithms have been proposed for solving MOO and BLO problems (e.g., (Konak, Coit, and Smith 2006; Sener and Koltun 2018; Yang et al. 2024a) for MOO and (Ji, Yang, and Liang 2021; Yang, Ji, and Liang 2021; Zhang et al. 2024; Qiu et al. 2023) for BLO, respectively; see Section 2 for in-depth discussions), the field of MOBL remains in its infancy, with many important open problems yet to be addressed. This gap between current MOBL research state and practical needs of MOBL arises primarily from the following technical challenges.

Technical challenges: Solving MOBL problems is, surprisingly, *significantly more challenging* than a straightforward combination of applying existing MOO and BLO solution techniques, which is due to the following two key factors: **First**, being both multi-objective and bilevel, an MOBL problem not only inherits all the challenges of MOO and BLO, but also sees many unique and new complications unseen in either MOO or BLO. Specifically, due to the finite-time constraint in solving the LL problem under the bilevel structure of MOBL problems, one typically has an inexact y_k at each iteration instead of the true solution $y^*(x_k)$. This inaccuracy in y_k subsequently results in a *biased estimate* $\hat{\nabla}\phi_s(x_k)$, which could *deviate* from a common descent direction computed according to $\nabla\phi_s(x_k)$ for all objectives.

Beyond the cascaded error in $\nabla \phi_s(x_k)$, the inaccuracy can accumulate over iterations, causing the solution trajectory to *drift significantly* from the desired target. **Second**, preference-guided Pareto front (i.e., the set of all Pareto-optimal solutions) exploration in MOBL is even more challenging (see solution philosophies in Section 3 for more details), since the Pareto stationarity (necessary condition of Pareto-optimality) and preference alignment should be balanced in the algorithmic design, which is highly susceptible to the accumulative cascaded errors inherent in the bilevel structure. Moreover, although existing works have incorporated scalarization methods to handle preference alignment, conventional rescaling strategies no longer apply due to the bilevel nature of MOBL problems, thereby necessitating novel technical designs.

Main contributions: In this paper, we propose a series of new MOBL algorithms to overcome the aforementioned challenges, all of which enjoy a fast finite-time convergence rate guarantees and hence low oracle complexities. Collectively, our results contribute toward establishing a theoretical foundation for MOBL. Our main contributions and key results are summarized as follows:

- We propose a unifying MOBL algorithmic framework called weighted-Chebyshev multi-hyper-gradient-descent (WC-MHGD) for both deterministic and stochastic settings. Our WC-MHGD framework offers provable fast finite-time convergence rates and low oracle complexities of $\mathcal{O}(\epsilon^{-1})$ and $\mathcal{O}(\epsilon^{-2})$ for achieving an ϵ-Pareto stationary solution in deterministic and stochastic settings, respectively. To our knowledge, our finite-time convergence and oracle complexity results are the first of their kind in the MOBL literature involving preference.
- It is worth pointing out that the algorithmic design and theoretical convergence analysis of our proposed WC-MHGD framework are highly non-trivial and far from straightforward extensions of traditional MOO and BLO techniques. Notably, we rigorously show that our *new* dynamic weighting optimization subproblem (cf. Eq. (5)) guarantees Pareto stationarity in MOBL problems. Moreover, we establish a general result that the preference-guided dynamic weighted hypergradient direction in WC-MHGD satisfies a key minimum-norm condition (cf. Lemma 5.4), which serves as a foundation for our theoretical analysis and could be of independent interest for other MOBL research problems.
- In addition to WC-MHGD algorithmic design and theoretical convergence analysis, we also offer the rationale behind our key algorithmic approaches and the geometric interpretation of our WC-MHGD approach in this paper, which advances and deepens our fundamental understandings of MOBL optimization theory and algorithms. Moreover, in Section 6, we conduct extensive numerical experiments to verify our WC-MHGD algorithms. All of our empirical results confirm the efficacy and effectiveness of our WC-MHGD algorithms.

2 Related Work

In this section, we provide an overview on recent works in MOBL to put our work in comparative perspectives. Due to space limitation, we relegate the overview of two closely related research areas, namely multi-objective optimization and bilevel optimization to Appendix B.

As mentioned in Section 1, the field of MOBL remains in a nascent stage and results in this area are rather limited. To our knowledge, existing works on MOBL include (Wang, Singh, and Ray 2024; Li et al. 2024; Yang et al. 2024b; Ye et al. 2024, 2021; Gu et al. 2023). Among these, the MOBL algorithms proposed in (Wang, Singh, and Ray 2024; Li et al. 2024) only provided empirical results without theoretical convergence and oracle complexity analysis. In contrast, it was shown that algorithms in (Yang et al. 2024b; Ye et al. 2021) converge to a Pareto stationary point. However, no finite-time convergence rate result was provided.

The most related works to our paper are (Ye et al. 2024; Fernando et al. 2022), which have been shown to offer a finite-time convergence rate for the deterministic and stochastic setting, respectively. However, our work differs from (Fernando et al. 2022; Ye et al. 2024) in the following key aspects: **(1)** Due to the use of less efficient value function algorithmic approach, the convergence rates in (Fernando et al. 2022; Ye et al. 2024) are $\mathcal{O}(SK^{-\frac{1}{2}})$ and $\mathcal{O}(1/K^{\frac{1}{4}})$, respectively. In contrast, with a judicious algorithmic design, our WC-MHGD approach achieves a significantly faster $\mathcal{O}(1/K)$ finite-time convergence rate, and covers both the deterministic setting and the stochastic setting simultaneously; and **(2)** Our WC-MHGD approach enables a systematic *Pareto stationarity front exploration* guided by preferences (see solution philosophies in Section 3 for more details). Due to space limitation, we provide a detailed comparison in Appendix B.

3 Multi-Objective Bilevel Learning: A Primer

To facilitate our MOBL deterministic and stochastic algorithm designs and theoretical convergence analysis, we first provide a primer on MOBL in this section.

Deterministic and Stochastic Versions of MOBL: Note that the deterministic version of the MOBL problem is stated Eq. (1) in Section 1. For the stochastic version of the MOBL problem, we assume the UL and LL objective functions can be expressed in the following forms, respectively:

$$\phi_s(x) = f^{(s)}(x, y^*(x)) := \mathbb{E}_{\xi \sim \pi_s}[F^{(s)}(x, y^*(x); \xi)],$$
$$g(x, y) := \mathbb{E}_{\zeta \sim \pi_g}[G(x, y; \zeta)],$$

where π_s and π_g represent the data distributions underlying the s-th UL and the LL objective functions, respectively. As will be shown later, the deterministic and stochastic versions of MOBL will have key differences in their algorithmic designs and theoretical convergence analysis.

Optimality Criterion of MOBL: As mentioned in Section 1, it is in general impossible to find a common x-solution to minimize all UL functions simultaneously due to the presence of potential conflicts between them. As a result, it is more appropriate to adopt the following notion of Pareto optimality as the performance metric for MOBL:

Definition 3.1 (Pareto Optimality). A solution x_1 is said to dominate another solution x_2 if and only if 1) $\phi_s(x_1) \leq \phi_s(x_2)$, $\forall s \in [S]$ and 2) there exists at least one $s \in [S]$

such that the inequality holds strictly. A solution x is Pareto optimal if no other x' dominates x.

Similar to solving most non-convex MOO or BLO problems, finding a Pareto-optimal solution in non-convex MOBL is NP-Hard in general. As a result, it is often more practical to search for a Pareto-stationary solution defined as follows (a necessary condition for Pareto optimality):

Definition 3.2 (Pareto Stationarity)**.** A solution x is Pareto stationary if there does not exist a common descent direction $d \in \mathbb{R}^p$ such that $\nabla \phi_s(x)^\top d < 0, \forall s \in [S]$.

Although Definition 3.2 rigorously defines the notion of Pareto stationary solution, it is inconvenient to work with in MOBL algorithm design. Toward this end, we will adopt the notion of ϵ-Pareto stationary solution defined as follows:

Definition 3.3 (ϵ-Pareto Stationarity)**.** Let $\nabla\Phi(x) := [\nabla\phi_1(x), \ldots, \nabla\phi_S(x)]^\top$. A solution x is ϵ-Pareto stationary if, for $\epsilon > 0$, there exists a vector $\lambda \in \mathbb{R}^S$ in the S-simplex (i.e., λ is nonnegative and $\mathbf{1}^\top \lambda = 1$) such that the λ-weighted gradient direction $d := \nabla\Phi(x)\lambda \in \mathbb{R}^p$ satisfies $\|d\|^2 \leq \epsilon$.

The above ϵ-Pareto stationarity is derived from solving the Lagrangian dual problem of minimizing the worst descent among all UL objectives, which is logically equivalent to Definition 3.1 (cf. e.g., (Sener and Koltun 2018; Yang et al. 2024a; Zhou et al. 2024)). We note that Definition 3.3 not only provides a useful metric in convergence analysis, but also motivates our WC-MHGD algorithm design later.

MOBL Solution Philosophies: Due to the flexible multi-objective nature of MOBL problems, there are four basic solution philosophies commonly seen in practice, depending on whether or not a preference weight vector $r \in \Delta_S^+$ (S-simplex) associated with the UL objectives is used:

P1) Given preference r, identify a Pareto stationary point that aligns most closely with the preference;

P2) Explore (partially) the Pareto front first, and then allow the decision-maker to select one from the exploration results based on some given preference r;

P3) Allow the decision-maker to select iteratively by combining 1) and 2) with changing preferences;

P4) Identify a solution based on certain global criterion rather than relying on preference information.

In this paper, we focus on Philosophy **P2**, since (i) the preference weight r could be difficult to pre-determine in many MOBL applications; and (ii) the preference vector r may not intersect with the Pareto front, rendering the P1 problem ill-defined. Also, we will show in Section 5 that the P4 can be treated as a *special case* of our proposed solution.

Under Philosophy P2, our goal in solving a non-convex MOBL problem is two-fold: (1) achieving a ϵ-Pareto stationary solution; and (2) systematically exploring the Pareto-stationarity front (i.e., the collection of all Pareto-stationary solutions) guided by preference information.

4 The Proposed WC-MHGD Algorithms

In this section, we will first provide necessary preliminaries of our algorithmic designs in Section 4.1. Then, we will present our WC-MHGD algorithms for deterministic and stochastic settings in Sections 4.2 and 4.3, respectively.

4.1 Preliminaries

The basic idea of our WC-MHGD algorithms: To achieve our first goal in finding a Pareto stationary solution in MOBL, our idea is to generalize the multi-gradient descent algorithm (MGDA) (Désidéri 2012) for solving MOO problems (Coello 2007; Elsken, Metzen, and Hutter 2018; Xu et al. 2024; Sener and Koltun 2018) by replacing gradient information in MGDA with the *hypergradient* used in BLO approaches, hence the name multi-hypergradient-descent (MHGD). Specifically, MHGD can be viewed as an extension of the (stochastic) gradient descent method to the multi-objective bilevel setting by first identifying a *common UL descent* direction that reduces the values of all S UL objectives, and then move along this direction with an appropriately chosen step-size. If no common descent direction exists, then the current solution is a Pareto stationary point of the UL subproblem.

Next, to achieve the second goal in systematic Pareto-stationarity front exploration, we propose to integrate the weighted Chebyshev (WC) scalarization technique with our MHGD design. The WC-scalarization transforms a vector-valued objective function into a scalar-valued objective function, which is more amenable for algorithm design. Specifically, let Δ_S^+ represent the S-dimensional simplex. In an MOBL problem, the WC-scalarization with a preference vector $r \in \Delta_S^+$ is defined in the following **min-max** form: $\mathsf{WC}_r(\Phi(x)) := \min_x \max_i \{r_i \phi_i(x)\}_{i=1}^S = \min_x \|r \odot \Phi(x)\|_\infty$, where $\odot$ denotes the Hadamard product. The use of WC-scalarization in our WC-MHGD algorithmic design is motivated by the following key fact in MOO (Golovin and Zhang 2020; Qiu et al. 2024):

Lemma 4.1 (Pareto Optimality Equivalence)**.** *A solution x^* is weakly Pareto-optimal to an MOO problem $\min_x \Phi(x)$ if and only if $x^* \in \arg\min_x \mathsf{WC}_r(\Phi(x))$ for some $r \in \Delta_S^+$.*

Lemma 4.1 suggests that, by appropriately integrating WC-scalarization in MOBL algorithm design, we can systematically obtain all weakly Pareto-optimal solutions (i.e., exploring the weak Pareto front) by enumerating the WC-scalarization preference weight vector r if the WC-scalarization problem can be solved optimally. Indeed, combining the WC scalarization and MHGD yields our WC-MHGD framework.

The hypergradient of the MOBL problem: In our WC-MHGD algorithms, we need to evaluate the set of all hypergradients denoted by the matrix $\nabla\Phi(x)$, or equivalently, to find the hypergradient $\nabla\phi_s(x)$ of each UL objective $s \in [S]$. Following the implicit function theorem (Zhang et al. 2024), it can be shown that the each UL hypergradient can be computed as follows (Ghadimi and Wang 2018; Ji, Yang, and Liang 2021):

$$\nabla\phi_s(x) = \nabla_x f^{(s)}(x, y^*(x)) - \nabla_{xy}^2 g(x, y^*(x)) v^*, \quad (2)$$

where v^* is the solution of the following linear equation system $\nabla_y^2 g(x, y^*(x)) v = \nabla_y f^{(s)}(x, y^*(x))$. It is clear from Eq. (2) and the definition of v^* that the Hessian $\nabla_y^2 g(x, y^*(x))$ being invertible is a *necessary* condition for the hypergradient $\phi_s(x)$ to be well-defined. Similar to most existing works on bilevel optimization in the ML literature, in this paper, we assume that the LL function $g(x, y)$ is strongly convex in y (cf. Assumption 5.1), so that $\nabla_y^2 g(x, y^*(x))$ is positive definite and hence invertible. Thus, the LL solution $y^*(x)$ can be uniquely determined.

In practice, however, since the LL subproblem is typically solved numerically, one usually does *not* have the exact $y^*(x)$-solution and can only use some estimation of $y^*(x)$ in Eq. (2) to compute the hypergradient. Consequently, the inexact $y^*(x)$-information results in systematic bias in hypergradient information. Indeed, most of our subsequent MOBL algorithm designs in deterministic and stochastic settings are centered around addressing this key challenge.

Moreover, note that the hypergradient evaluation in Eq. (2) involves numerous first-order and second-order oracle evaluations (i.e., the gradient information, the Jacobian information, and the Hessian information), all of which are computationally expensive. Therefore, to evaluate the efficiency of our proposed algorithms, we adopt the *oracle complexity* metric (Zhang et al. 2024). Specifically, we let $\mathsf{Gc}(f^{(s)}, \epsilon)$ denote the number of the first-order oracle evaluations (i.e., partial gradients $\nabla_x f^{(s)}$ and $\nabla_y f^{(s)}$) for $f^{(s)}(x, y)$, $s \in [S]$. Also, for a vector v and a positive ϵ, we let $\mathsf{JV}(g, \epsilon)$ and $\mathsf{HV}(g, \epsilon)$ represent the numbers of second-order oracle evaluations in Jacobian-vector product $\nabla_{xy}^2 g(x, y)v$ and Hessian-vector product $\nabla_y^2 g(x, y)v$, respectively. Then, the oracle complexity is defined as follows:

Definition 4.2 (Oracle Complexity of MOBL Algorithms)**.** The oracle complexity of an MOBL algorithm is defined as the total required numbers of first-order and second-order oracle evaluations for the algorithm to converge to an ϵ-Pareto-stationary solution.

With all these preliminaries, we are now in a position to present our WC-MHGD algorithms.

4.2 The Deterministic WC-MHGD Algorithm

As illustrated in Algorithm 1, our deterministic WC-MHGD algorithm adopts a "double-loop" structure. In the inner-loop, we perform gradient-descent-style updates to approximately find the optimal $y^*(x_k)$-solution in the k-th outer iteration. To accelerate convergence, we apply a "warm start" technique inspired by (Ji, Yang, and Liang 2021): the output from the end of the previous round of inner iterations serves as the initial point for the current round of inner iterations. After D inner iterations, y_k^D will be used as an estimation of $y^*(x_k)$. In the k-th iteration of the outer-loop, we offer two options to compute the hypergradient for each UL objective $s \in [S]$. Option 1 performs N steps of the conjugate-gradient method (CG) to solve the linear equation system $\nabla_y^2 g(x_k, y_k^D)v = \nabla_y f^{(s)}(x_k, y_k^D)$ to obtain $v_k^{(s),N}$, which is used to compute the hypergradient as:

$$\hat{\nabla}\phi_s(x_k) = \nabla_x f^{(s)}(x_k, y_k^D) - \nabla_{xy}^2 g(x_k, y_k^D) v_k^{(s),N}. \quad (3)$$

In comparison, Option 2 approximates the Hessian inverse with the Neumann series $[\nabla_y^2 g(x_k, y_k^D)]^{-1} \approx \prod_{j=t+1}^{D-1}(I - \nabla_y^2 g(x_k, y_k^j))$ and compute the hypergradient as:

$$\hat{\nabla}\phi_s(x_k) = \nabla_x f^{(s)}(x_k, y_k^D) - \alpha \sum_{t=0}^{D} \nabla_{xy}^2 g(x_k, y_k^t) \prod_{j=t+1}^{D-1} (I - \nabla_y^2 g(x_k, y_k^j)) \nabla_y f^{(s)}(x_k, y_k^D). \quad (4)$$

Algorithm 1: Deterministic WC-MHGD.

Input: The numbers of iteration K, D, N, initialization values x_0, y_0, v_0, preference r, and step-sizes α, β.
for $k = 0, 1, \ldots, K-1$ **do**
  Set $y_k^0 = y_{k-1}^D$ if $k > 0$ and y_0 otherwise.
  for $t = 1, 2, \ldots, D$ **do**
    Update $y_k^t = y_k^{t-1} - \alpha \nabla_y g(x_k, y_k^{t-1})$.
  end for
  for $s \in [S]$ **do**
    Option 1: Conjugate-Gradient(CG)
    Set $v_k^{(s),0} = v_{k-1}^{(s),N}$ if $k > 0$ and v_0 otherwise.
    Solve $\nabla_y^2 g(x_k, y_k^D)v = \nabla_y f^{(s)}(x_k, y_k^D)$ for N steps from $v_k^{(s),0}$ to get $v_k^{(s),N}$.
    Compute $\hat{\nabla}\phi_s(x_k)$ according to Eq. (3).
    Option 2: Neumann-Series(NS)
    Compute $\hat{\nabla}\phi_s(x_k)$ according to Eq. (4).
  end for
  Compute λ_k according to Equation (5).
  Update $x_{k+1} = x_k - \beta \hat{\nabla}\Phi(x_k)(r \odot \lambda_k)$.
end for

With the hypergradients obtained from Eqs. (3) or (4), the heart of our WC-MHGD algorithm is to solve the following convex quadratic programming problem to find the optimal λ-weights to linearly combine hypergradients:

$$\min_{\lambda \geq \mathbf{0}} \underbrace{\|K(r \odot \lambda)\|^2}_{\text{MHGD}} - u \underbrace{\lambda^\top (r \odot \Phi(x_k))}_{\text{WC-Scalarization}} \quad \text{s.t. } \mathbf{1}^\top \lambda = 1, \quad (5)$$

where $u > 0$ is a tunable parameter, and $K := (\hat{\nabla}\Phi(x_k)^\top \hat{\nabla}\Phi(x_k))^{\frac{1}{2}} \in \mathbb{R}^{S \times S}$. Upon obtaining λ_k^u by solving Problem (5), we compute the update direction $d_k(u)$ by taking the gradient of the Lagrangian of Problem (5) to yield $d_k(u) = \hat{\nabla}\Phi(x_k)(r \odot \lambda_k^u)$. For brevity, we will simply denote λ_k^u and $d_k(u)$ as λ_k and d_k, respectively, when the context is clear. Four important remarks are in order on Problem (5), which is a **new design** *unseen* in the literature:

1) The first term in Problem (5) plays the role of searching for a Pareto-stationary solution. To see this, it is insightful to recognize that the first term $\|K(r \odot \lambda)\|^2$ in Problem (5) is equivalent to $\|\hat{\nabla}\Phi(x_k)(r \odot \lambda)\|^2$ in the k-th iteration (it is advantageous to use K instead of $\hat{\nabla}\Phi(x_k)$ since the dimension of K is typically much smaller ($S \ll p$)). Consequently, $\|K(r \odot \lambda)\|^2 = \|\hat{\nabla}\Phi(x_k)\text{diag}(r)\lambda\|^2$, where $\text{diag}(r)$ denotes the diagonal matrix with elements in r on the main diagonal. $\|K(r \odot \lambda)\|^2$ can be viewed as an r-scaled version

Algorithm 2: Stochastic WC-MHGD.

Input: The numbers of iteration K, D, Q, initialization values x_0, y_0, preference r, and step-sizes α, β.
for $k = 0, 1, \ldots, K$ **do**
  Set $y_k^0 = y_{k-1}^D$ if $k > 0$ and y_0 otherwise.
  for $t = 1, 2, \ldots, D$ **do**
    Draw a sample batch $\mathcal{T}_{t-1}$, and update $y_k^t = y_k^{t-1} - \alpha \nabla_y G(x_k, y_k^{t-1}; \mathcal{T}_{t-1})$.
  end for
  Draw sample batches $\mathcal{D}_G, \mathcal{D}_H$.
  for $s \in [S]$ **do**
    Draw sample batch $\mathcal{D}_F^s$.
    Set $v_k^{(s),0} = \nabla_y F^{(s)}(x_k, y_k^D; \mathcal{D}_F^s)$.
    Compute $v_k^{(s),Q}$ via algorithm 3, and $\hat{\nabla}\phi_s(x_k) = \nabla_x F^{(s)}(x_k, y_k^D; \mathcal{D}_F^s) - \nabla_{xy}^2 G(x_k, y_k^D; \mathcal{D}_G) v_k^{(s),Q}$.
  end for
  Compute λ_k according to Eq. (5).
  Update $x_{k+1} = x_k - \beta \hat{\nabla}\Phi(x_k)(r \odot \lambda_k)$.
end for

Algorithm 3: Hessian-vector Product $v_k^{(s),Q}$ Computation

Input: The number of iteration Q, initialization values $v_k^{(s),Q}$, samples $\mathcal{D}_H = \{\mathcal{B}_j\}_{j=1}^Q$ and step-sizes η.
for $j = 1, 2, \ldots, Q$ **do**
  Use $\mathcal{B}_j$ to compute $G_j(y) = y - \eta \nabla_y G(x, y; \mathcal{B}_j)$.
end for
Set $\nu^{(s),Q} = v_k^{(s),0}$.
for $i = Q, Q-1, \ldots, 1$ **do**
  Compute $\nu^{(s),i-1} = \nu^{(s),i} - \eta \nabla_y^2 G(x, y; \mathcal{B}_i) \nu^{(s),i}$.
end for
Output: $v_k^{(s),Q} = \eta \sum_{i=0}^{Q} \nu^{(s),i-1}$.

of $\|\hat{\nabla}\Phi(x_k)\lambda\|^2$. Since r is strictly positive, the λ-vector that minimizes $\|\hat{\nabla}\Phi(x_k)\lambda\|^2$ provides a **common descent direction** $d = \hat{\nabla}\Phi(x_k)(r \odot \lambda)$ that improves all objectives. It is worth pointing out that the use of $\|K(r \odot \lambda)\|^2$ instead of the quadratic term $\|\Phi(x_k)\lambda\|^2$ in the original MGDA (Désidéri 2012) is **a major novelty** in this paper. Indeed, this above Pareto-stationarity intuition can be made rigorous and stated as follows (proved in Appendix D):

Lemma 4.3 (Pareto Stationarity)**.** *For any preference $r \in \mathbb{R}_{++}^S$, if the first term in Eq.* (5) *satisfies $K(r \odot \lambda) = 0$, then Pareto stationarity is achieved.*

2) The second term $-\lambda^\top(r \odot \Phi(x_k))$ in Problem (5) is used to facilitate the WC-scalarization for systematic Pareto stationarity front exploration. To see why this term is corresponding to the WC scalarization, note that the primal WC problem can be written as minimizing a scalar γ subject to $r \odot \Phi(x_k) \leq \gamma \mathbf{1}$, which effectively minimizes the weighted loss in ℓ_∞-norm sense (Momma, Dong, and Liu 2022). Then, by taking the Wolfe dual to incorporate Pareto-stationarity conditions, this WC-related objective transforms into maximizing $u\lambda^\top(r \odot \Phi(x_k))$, or equivalently minimizing $-u\lambda^\top(r \odot \Phi(x_k))$.

3) By choosing an appropriate u-value, the objective function in Problem (5) strikes a balance between minimizing Pareto-stationarity gap under preference r and systematically explore Pareto-stationarity front following the r-weighted WC scalarization. Specifically, if Pareto-stationarity front exploration is of higher priority, one can choose a larger u-value. Otherwise, a small u-value favors minimizing the Pareto-stationarity gap of the obtained solution. Geometric insight of Problem (5) is provided in Appendix C due to space limitation.

4) In addition to the new algorithmic design of Problem (5) (as shown in Lemma 5.4), we also emphasize that the overall framework design of WC-MHGD approach is a *consequence of the special problem structure in MOBL.* Specifically, we note that a family of BLO algorithms (namely SOBA-based approaches update UL and LL variables x and y simultaneously in each iteration, (Dagréou et al. 2022; Liu et al. 2023a)) achieves efficient convergence rate in the context of BLO. However, such SOBA-based approaches are not applicable in MOBL when multiple objectives are taken into account, since the v-variables lose their dual variable meaning as their counterpart in the single-objective setting. In contrast, our WC-MHGD method handles LL and UL variables in an alternating manner to avoid the explicit use of dual variables. This overcomes the limitation of the SOBA-based approaches and is specifically designed for MOBL problems.

4.3 The Stochastic WC-MHGD Algorithm

The stochastic WC-MHGD is illustrated in Algorithm 2, which shares a similar structure and is based on the same intuition as in Algorithm 1. The key differences of the stochastic WC-MHGD algorithm include: (1) In the t-th inner loop iteration, the y-updates are conducted in a stochastic gradient descent fashion with a sample batch $\mathcal{T}_{t-1}$ and similar warm start technique as in deterministic WC-MHGD. (2) In the k-th outer loop iteration, the Hessian-vector product $v_k^{(s),Q}$ required for hypergradient computation is inspired by (Ji, Yang, and Liang 2021), which also uses the Neumann series technique for Hessian inverse approximation as shown in Algorithm 3. However, the computation of $v_k^{(s),Q}$ is significantly more complex compared to the deterministic case and care must be taken to tame oracle complexity. Similar to (Ji, Yang, and Liang 2021), we select mutually independent datasets $\mathcal{B}_j$ with exponentially shrinking sizes in Algorithm 3.

5 Pareto-Stationarity Convergence Analysis

In this section, we start by introducing several assumptions required for our analysis, which are followed by the main finite-time convergence results of our WC-MHGD algorithms.

Assumption 5.1 (LL Objective Function)**.** For any $x \in \mathbb{R}^p$, $g(x, \cdot)$ in the deterministic setting and $G(x, \cdot; \zeta)$ in the stochastic setting are μ_g-strongly-convex with respect to y for some constant $\mu_g > 0$.

We note that Assumption 5.1 is standard and has been widely adopted in the literature (Ji, Yang, and Liang 2021; Yang, Ji, and Liang 2021; Liu et al. 2023b; Hu et al. 2023; Yang, Xiao, and Ji 2023; Gong, Hao, and Liu 2024). On one hand, strong convexity ensures that the Hessian inverse $\nabla_y^2 g(x,y)^{-1}$ well-defined and that the solution to the LL subproblem can be uniquely determined; on the other hand, it enables the quantification of the error between the approximated and ground-truth hypergradients.

Assumption 5.2 (Smoothness)**.** There exist positive constants M, L, τ, and ρ, such that for any $s \in [S]$ and any $z, z' \in \mathbb{R}^{p+q}$, the following Lipchitz continuity conditions hold under the deterministic setting:

$$
\begin{aligned}
&|f^{(s)}(z) - f^{(s)}(z')| \leq M\|z - z'\|, \\
&\|\nabla f^{(s)}(z) - \nabla f^{(s)}(z')\| \leq L\|z - z'\|, \\
&\|\nabla g(z) - \nabla g(z')\| \leq L\|z - z'\|, \\
&\|\nabla_{xy}^2 g(z) - \nabla_{xy}^2 g(z')\| \leq \tau\|z - z'\|, \\
&\|\nabla_y^2 g(z) - \nabla_y^2 g(z')\| \leq \rho\|z - z'\|,
\end{aligned}
$$

where $\|\cdot\|$ denotes the ℓ_2-induced norm (abbreviated as $\|\cdot\|$ in the sequel for simplicity). In addition, the stochastic functions $F^{(s)}(z;\xi)$ and $G(z;\zeta)$ also satisfy their corresponding smoothness assumptions for any sample pairs ξ and ζ.

We note that the above smoothness assumptions are also standard and have been widely adopted in the literature for convergence analysis (Ghadimi and Wang 2018; Ji, Yang, and Liang 2021; Qiu et al. 2023; Yang, Ji, and Liang 2021; Lin et al. 2024). Lastly, we make the following assumption on the variance of the LL objective function as follows:

Assumption 5.3 (Bounded LL Variance)**.** The LL stochastic gradient $\nabla G(z;\zeta)$ has a variance bounded by a constant $\sigma > 0$, i.e., $\mathbb{E}_\zeta\|\nabla G(z;\zeta) - \nabla g(z)\|^2 \leq \sigma^2$.

Convergence Rate of the Deterministic WC-MHGD**:** We let $\bar{d}_k = \nabla\Phi(x_k)\mathrm{diag}(r)\lambda_k$ for any $k \in \{0, \ldots, K\}$. Also, we let $\kappa = \frac{L}{\mu_g}$ denote the condition number and define $L_\phi := L + \frac{2L^2+\tau M^2}{\mu_g} + \frac{\rho LM + L^3 + \tau LM}{\mu_g^2} + \frac{\rho L^2 M}{\mu_g^3} = \Theta(\kappa^3)$. We first state a key result to establish the convergence of our WC-MHGD method, which is new for MOBL problems and not in the MOO and BLO literature.

Lemma 5.4 (Preference-Guided Minimum-Norm Lemma)**.** *Let λ_k^u be obtained from solving Problem* (5) *for some $u > 0$ and thus the update direction is $d_k(u) := \hat{\nabla}\Phi(x_k)(r \odot \lambda_k^u)$. Let r_{max} be the maximal element of r. For any $s \in [S]$ and $r \in \Delta_S^+$, there exists a sufficiently small u such that the following holds: $\|d_k(u)\|^2 \leq 2r_{max}\langle d_k(u), \hat{\nabla}\phi_s(x_k)\rangle$.*

Lemma 5.4 plays an important role in the subsequent analysis. While the relationship between the derived direction d_k and the gradient $\hat{\nabla}\phi_s(x_k)$ in MGDA is well understood, it only holds in conventional MOO settings. Therefore, adopting the newly designed Problem (5) is essential for MOBL problems. However, this new design also implies a *new* common descent direction $d_k(u)$, for which *no existing works* have characterized its properties. To this end, we establish Lemma 5.4 to reveal how $d_k(u)$ correlates with r and $\hat{\nabla}\phi_s$. The result proved in Lemma 5.4 further leads to our finite-time convergence rate result, which is stated as follows:

Theorem 5.5 (Finite-Time Convergence Rate of Deterministic WC-MHGD)**.** *Under Assumptions 5.1 and 5.2, for any preference $r \in \Delta_S^{++}$ and $\epsilon > 0$, by choosing $\alpha \leq \frac{1}{L}$, $\beta = \min\{\frac{1}{2(1+L_\phi)r_{max}}, \frac{1}{3L_\phi}\}$, $D \geq \Theta(\kappa \log\frac{1}{\epsilon})$, $N \geq \Theta(\sqrt{\kappa}\log\frac{1}{\epsilon})$, and $K \geq \Theta(\kappa^3\epsilon^{-1})$, the following results of Algorithm 1 hold:*

- *The iterates under the NS option satisfy:*

$$
\begin{aligned}
\frac{1}{K}\sum_{k=0}^{K-1}\|\bar{d}_k\|^2 &\leq \frac{24r_{max}L_\phi(\phi_s(x_0) - \phi_s(x_K))}{K} \\
&+ (12 + 24r_{max}L_\phi)\cdot\Bigg[\Bigg(\frac{4M^2(\tau\mu_g + L\rho)^2}{\mu_g^4}(1-\alpha\mu_g)^{D-1} \\
&+ \frac{2L^2(L+\mu_g)^2(1-\alpha\mu_g)^D}{\mu_g^2}\Bigg)\chi + \frac{L^2M^2(1-\alpha\mu_g)^{2D}}{\mu_g^2}\Bigg],
\end{aligned}
$$

- *The iterates under the CG option satisfy:*

$$
\begin{aligned}
\frac{1}{K}\sum_{k=0}^{K-1}\|\bar{d}_k\|^2 &\leq \frac{12r_{max}L_\phi\Big(\phi_s(x_0) - \phi_s(x_K) + \delta_{D,N}\chi_0\Big)}{K} \\
&+ 24(r_{max}L_\phi + 1)M^2\delta_{D,N}\Omega^{\frac{3}{2}} + 6\delta_{D,N}\chi_0,
\end{aligned}
$$

where $\delta_{D,N} := \Gamma(1-\alpha\mu_g)^D + 6L^2\kappa\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2N}$, $\Omega := 4\beta^2\left(\kappa^2 + \frac{ML}{\mu_g^2} + \frac{ML\kappa}{\mu_g^2}\right)^2$, $\chi_0 := \|y_0 - y^*(x_0)\|^2 + \|v_0 - v_0^*\|^2$, and Γ and χ are positive constants.

The following oracle complexity results immediately follow from Theorem 5.5:

Corollary 5.6 (Oracle Complexity of Deterministic WC-MHGD)**.** *To achieve an ϵ-Pareto stationary point, by choosing D and N as Appendix D, Algorithm 1 achieves the following oracle complexities:*

- *NS:* $\mathsf{Gc}(f,\epsilon) = \mathcal{O}(\kappa^3\epsilon^{-1}S)$, $\mathsf{Gc}(g,\epsilon) = \widetilde{\mathcal{O}}(\kappa^4\epsilon^{-1})$, $\mathsf{JV}(g,\epsilon) = \widetilde{\mathcal{O}}(\kappa^4\epsilon^{-1}S)$, $\mathsf{HV}(g,\epsilon) = \widetilde{\mathcal{O}}(\kappa^4\epsilon^{-1}S)$.
- *CG:* $\mathsf{Gc}(f,\epsilon) = \mathcal{O}(\kappa^3\epsilon^{-1}S)$, $\mathsf{Gc}(g,\epsilon) = \widetilde{\mathcal{O}}(\kappa^4\epsilon^{-1})$, $\mathsf{JV}(g,\epsilon) = \mathcal{O}(\kappa^3\epsilon^{-1}S)$, $\mathsf{HV}(g,\epsilon) = \widetilde{\mathcal{O}}(\kappa^{3.5}\epsilon^{-1}S)$.

Compared to single-objective BLO approaches (e.g., (Ji, Yang, and Liang 2021)), both options in Algorithm 1 achieve the *same* convergence rate of $\mathcal{O}(1/K)$. Also, all oracle complexity bounds increase by a factor of S, which is due to the fact that oracle samples are needed for each of the S UL objectives. Notably, Theorem 5.5 indicates that, even with more complex preference incorporation, our WC-MHGD approach still converges to a neighborhood of a preference-guided Pareto stationary point. This highly non-trivial preference-guided Pareto-stationarity convergence, which is similar to that of the conventional MGDA method, critically depends on our new algorithmic design in Problem (5) – a main contribution of this work as pointed out previously.

Convergence rate of the stochastic WC-MHGD**:** With the same notation as in the deterministic setting, we state the result of the stochastic WC-MHGD algorithm as follows:

Theorem 5.7 (Finite-Time Convergence Rate of Stochastic WC-MHGD)**.** *Under Assumptions 5.1 to 5.3, for any preference $r \in \Delta_S^{++}$ and $\epsilon > 0$, by choosing $\alpha = \frac{2}{L+\mu_g}$, $\beta = \min\{\frac{1}{2r_{max}(1+L_\phi)}, \frac{1}{3L_\phi}\}$, $\eta \leq \frac{1}{L}$, $D = \Theta(\kappa \log(\epsilon^{-1}))$, and $K = \Theta(\frac{\kappa^3}{\epsilon})$, the following results of Algorithm 2 hold:*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\bar{d}_k\|^2 \leq \gamma \cdot \frac{3M^2(\kappa^2+1)}{1-\gamma}\Big(1+2L_\phi r_{max}\Big) + \gamma \cdot \Big(6\nu + \frac{12L_\phi r_{max}\nu}{1-\gamma}\Big)\mathbb{E}\|y_0 - y^*(x_0)\|^2 + \mathcal{O}\Big(\frac{L_\phi}{K} + \frac{\kappa^8\sigma^2}{T} + \frac{\kappa^5}{D_g} + \frac{\kappa^5}{D_f} + \frac{\kappa^5}{B} + \kappa^5(1-\eta\mu_g)^{2Q}\Big),$$

where $\nu = \Theta(\kappa^4)$, $\gamma = 16\left(\frac{L-\mu_g}{L+\mu_g}\right)^{2D}\frac{\beta^2 L^2 \nu}{\mu_g^2}\frac{r_{max}}{1-r_{max}}$; and the batch-sizes satisfy $|\mathcal{B}_{Q+1-j}| = BQ(1-\eta\mu_g)^{j-1}$ for $j \in [Q]$, $BQ(1-\eta\mu_g)^{Q-1} \geq 1$, $Q = \Theta(\kappa \log(\kappa^2\epsilon^{-1}))$.

The following oracle complexity results immediately follow from Theorem 5.7:

Corollary 5.8 (Oracle Complexity of Stochastic WC-MHGD)**.** *To achieve an ϵ-Pareto stationary point, by choosing parameters as Appendix D, Algorithm 2 achieves the following oracle complexities:* $\mathsf{Gc}(f,\epsilon) = \mathcal{O}(\kappa^8\epsilon^{-2}S)$, $\mathsf{Gc}(g,\epsilon) = \widetilde{\mathcal{O}}(\kappa^{12}\epsilon^{-2})$, $\mathsf{JV}(g,\epsilon) = \mathcal{O}(\kappa^8\epsilon^{-2}S)$, $\mathsf{HV}(g,\epsilon) = \widetilde{\mathcal{O}}(\kappa^9\epsilon^{-2}S)$.

Theorem 5.7 shows that, as the numbers of iterations and batch sizes increase, Algorithm 2 converges to a neighborhood of a Pareto stationary solution at rate $\mathcal{O}(1/K)$. Compared to the single-objective BLO approaches (e.g., (Ji, Yang, and Liang 2021)), the oracle complexity results in Theorem 5.7 increase by a factor κ^3. This is because stochastic MOBL methods require more data to address the bilevel error propagation issue of each UL objective and mitigate the discrepancy between approximate and true hypergradients.

Lastly, we note that preference information may not always be available in certain MOBL applications. For such MOBL problems, we can still establish their Pareto-stationarity convergence rate results in terms of $\bar{d}_k$ under the special case without r. In fact, we show that the Pareto-stationarity convergence rate in this special case can be further tightened thanks to the simplified structure of the problem. Due to space limitations, we present this result in Appendix D.

6 Numerical Experiments

Experiment settings: *a) The Deterministic Setting Experiments:* We use a 5-way 5-shot meta-learning task to test Algorithm 1 on the `FC-100` dataset (Oreshkin, Rodríguez López, and Lacoste 2018), thus we have $S = 5$ objectives. For these tasks, we consider multiple models comprising shared parameters and task-specific parameters. The corresponding MOBL problem involves minimizing the training loss by first determining the task-specific LL parameters and then finding the optimal shared LL parameters based on the LL solutions to minimize the validation loss. We set $(K, D) = (500, 32)$ and $u = 10$ by default. We set the preference vector r as:

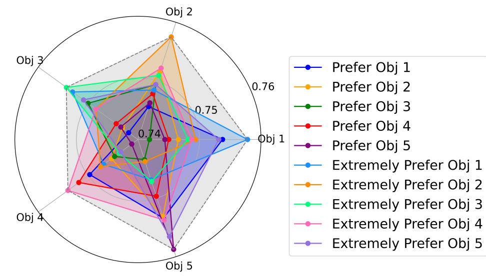

Figure 1: Pareto front exploration of Algorithm 1.

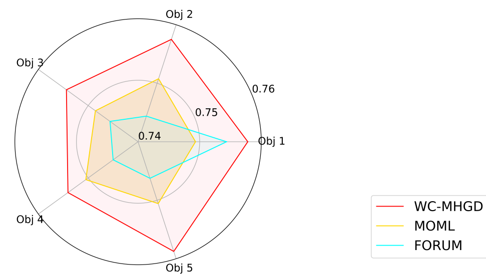

Figure 2: Comparison between algorithms.

i) $r_s = 0.8$ for some $s \in [S]$ and $r_{s'} = 0.05, \forall s' \neq s$; and ii) $r_s = 0.96$ for some $s \in [S]$ and $r_{s'} = 0.01, \forall s' \neq s$. Those objectives with $r_s = 0.8$ and $r_s = 0.96$ are referred to as *"preferred"* and *"extremely preferred,"* respectively. Each experiment is averaged over 4 trials.

b) The Stochastic Setting Experiments: We use a data hyper-cleaning task to test Algorithm 2 on the `MNIST` dataset (LeCun et al. 1998). Under 5 different corruption rates p (hence $S = 5$ objectives), multiple models share the same UL regularization parameter based on the updates of their p-specific LL parameters. We set $(K, D) = (150, 200)$ and $u = 10$ by default. Each experiment is averaged over 5 trials. The detailed setups can be found in Appendix E.

Experiment Results: Fig. 1 shows how Algorithm 1 converges to preference-guided Pareto stationary points. We can see that the accuracy of the more preferred objectives are consistently higher than those of the less preferred objectives. Moreover, for those extremely preferred objectives, this trend becomes more pronounced. The outer gray area demonstrates that our WC-MHGD algorithm effectively explores a large Pareto stationarity footprint. Fig. 2 highlights that Pareto exploration allows Algorithm 1 to cover a larger portion of Pareto front, compared to MOML (Ye et al. 2021) and FORUM (Ye et al. 2024). Due to space limitation, we relegate further experimental details and results in Appendix E.

7 Conclusion

In this paper, we investigated MOBL problems and proposed a unifying WC-MHGD algorithmic framework, which offers finite-time Pareto-stationarity convergence guarantees, low oracle complexities, and systematic Pareto-stationarity front exploration capability. Our numerical results further verified the effectiveness of our proposed algorithms.

Acknowledgements

This work is supported in part by Meta Research Award INB3267726, NSF grants CAREER CNS-2110259, CNS-2112471, and ECCS-233104, DARPA grants YFA D24AP00265 and HR0011-25-2-0019, and ONR grant N00014-24-1-2729.

References

Chakraborty, S.; Bedi, A.; Koppel, A.; Wang, H.; Manocha, D.; Wang, M.; and Huang, F. 2024. Parl: A unified framework for policy alignment in reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*.

Coello, C. A. C. 2007. *Evolutionary algorithms for solving multi-objective problems*. Springer.

Dagréou, 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.

Désidéri, J.-A. 2012. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6): 313–318.

Elsken, T.; Metzen, J. H.; and Hutter, F. 2018. Efficient multi-objective neural architecture search via lamarckian evolution. *arXiv preprint arXiv:1804.09081*.

Fernando, H.; Shen, H.; Liu, M.; Chaudhury, S.; Murugesan, K.; and Chen, T. 2022. Mitigating gradient bias in multi-objective learning: A provably convergent stochastic approach. *arXiv preprint arXiv:2210.12624*.

Ghadimi, S.; and Wang, M. 2018. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*.

Golovin, D.; and Zhang, Q. 2020. Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization. *ArXiv*, abs/2006.04655.

Gong, X.; Hao, J.; and Liu, M. 2024. An accelerated algorithm for stochastic bilevel optimization under unbounded smoothness. *arXiv preprint arXiv:2409.19212*.

Gu, A.; Lu, S.; Ram, P.; and Weng, T.-W. 2023. Min-max multi-objective bilevel optimization with applications in robust machine learning. In *The Eleventh International Conference on Learning Representations*.

Hu, Y.; Wang, J.; Xie, Y.; Krause, A.; and Kuhn, D. 2023. Contextual stochastic bilevel optimization. *Advances in Neural Information Processing Systems*, 36: 78412–78434.

Ji, K.; Yang, J.; and Liang, Y. 2021. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, 4882–4892. PMLR.

Konak, A.; Coit, D. W.; and Smith, A. E. 2006. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability engineering & system safety*, 91(9): 992–1007.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, F.; Yang, Y.; Liu, Y.; Liu, Y.; and Qian, M. 2024. Bi-Level Model Management Strategy for Solving Expensive Multi-Objective Optimization Problems. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Lin, S.; Sow, D.; Ji, K.; Liang, Y.; and Shroff, N. 2024. Non-convex bilevel optimization with time-varying objective functions. *Advances in Neural Information Processing Systems*, 36.

Liu, R.; Liu, Y.; Yao, W.; Zeng, S.; and Zhang, J. 2023a. Averaged method of multipliers for bi-level optimization without lower-level strong convexity. In *International Conference on Machine Learning*, 21839–21866. PMLR.

Liu, Z.; Zhang, X.; Khanduri, P.; Lu, S.; and Liu, J. 2023b. Prometheus: taming sample and communication complexities in constrained decentralized stochastic bilevel learning. In *International Conference on Machine Learning*, 22420–22453. PMLR.

Momma, M.; Dong, C.; and Liu, J. 2022. A multi-objective/multi-task learning framework induced by pareto stationarity. In *International Conference on Machine Learning*, 15895–15907. PMLR.

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.

Qiu, P.; Li, Y.; Liu, Z.; Khanduri, P.; Liu, J.; Shroff, N. B.; Bentley, E. S.; and Turck, K. 2023. DIAMOND: Taming Sample and Communication Complexities in Decentralized Bilevel Optimization. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, 1–10. IEEE.

Qiu, S.; Zhang, D.; Yang, R.; Lyu, B.; and Zhang, T. 2024. Traversing Pareto Optimal Policies: Provably Efficient Multi-Objective Reinforcement Learning. arXiv:2407.17466.

Sener, O.; and Koltun, V. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.

Shen, H.; Yang, Z.; and Chen, T. 2024. Principled penalty-based methods for bilevel reinforcement learning and rlhf. *arXiv preprint arXiv:2402.06886*.

Wang, B.; Singh, H. K.; and Ray, T. 2024. Pareto Set Prediction Assisted Bilevel Multi-objective Optimization. *arXiv preprint arXiv:2409.03328*.

Xu, M.; Ju, P.; Liu, J.; and Yang, H. 2024. PSMGD: Periodic Stochastic Multi-Gradient Descent for Fast Multi-Objective Optimization. *arXiv preprint arXiv:2412.10961*.

Yang, H.; Liu, Z.; Liu, J.; Dong, C.; and Momma, M. 2024a. Federated multi-objective learning. *Advances in Neural Information Processing Systems*, 36.

Yang, J.; Ji, K.; and Liang, Y. 2021. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34: 13670–13682.

Yang, X.; Yao, W.; Yin, H.; Zeng, S.; and Zhang, J. 2024b. Gradient-based algorithms for multi-objective bi-level optimization. *Science China Mathematics*, 1–20.

Yang, Y.; Xiao, P.; and Ji, K. 2023. Achieving $O(\epsilon^{-1.5})$ Complexity in Hessian/Jacobian-free Stochastic Bilevel Optimization. arXiv:2312.03807.

Ye, F.; Lin, B.; Cao, X.; Zhang, Y.; and Tsang, I. W. 2024. A first-order multi-gradient algorithm for multi-objective bilevel optimization. In *ECAI 2024*, 2621–2628. IOS Press.

Ye, F.; Lin, B.; Yue, Z.; Guo, P.; Xiao, Q.; and Zhang, Y. 2021. Multi-objective meta learning. *Advances in Neural Information Processing Systems*, 34: 21338–21351.

Zhang, Y.; Khanduri, P.; Tsaknakis, I.; Yao, Y.; Hong, M.; and Liu, S. 2024. An Introduction to Bilevel Optimization: Foundations and applications in signal processing and machine learning. *IEEE Signal Processing Magazine*, 41(1): 38–59.

Zhou, T.; Hairi, F.; Yang, H.; Liu, J.; Tong, T.; Yang, F.; Momma, M.; and Gao, Y. 2024. Finite-time convergence and sample complexity of actor-critic multi-objective reinforcement learning. *arXiv preprint arXiv:2405.03082*.