

ONLINE MULTI-OBJECTIVE CONVEX OPTIMIZATION: A UNIFIED FRAMEWORK AND JOINT GRADIENT DE- SCENT

Anonymous authors

Paper under double-blind review

ABSTRACT

Online Convex Optimization (OCO) usually addresses the learning task with a single objective; however, in real-world applications, multiple conflicting objectives often need to be optimized simultaneously. In this paper, we present an Online Multi-objective Convex Optimization (OMCO) framework with a novel multi-objective regret. We prove that, when the number of objectives in OMCO decreases to one, the regret is equal to the regret in OCO, thus unifying the OCO and OMCO frameworks. To facilitate the analysis of the proposed novel regret, we derive its equivalent form using the strong duality theory of convex optimization. Moreover, we propose an Online Joint Gradient Descent algorithm and prove that it achieves a sublinear multi-objective regret according to the equivalent regret form. Experimental results on several real-world datasets validate the effectiveness of our proposed algorithm.

1 INTRODUCTION

In recent years, with the rapid advancement of the Internet industry and Internet of Things technologies, online learning has become an essential paradigm for efficiently processing streaming data (Hoi et al., 2021). In online learning, the paramount framework is Online Convex Optimization (OCO) (Hazan, 2022), which has been widely studied. OCO can be regarded as a repeated game between a learner and an adversary: at each t -th round, the learner makes a decision \mathbf{x}_t from a convex set $X \in \mathbb{R}^n$ for future unforeseeable data based on previous data, and then the adversary reveals a convex loss $f_t(\mathbf{x}_t)$, which is unknown in advance and may change with time. The goal of the learner is to minimize *regret*, defined as the difference between the cumulative loss of the learner and that of a comparator sequence in hindsight over the time horizon T , i.e.,

$$regret_T = \sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in X} \sum_{t=1}^T f_t(\mathbf{x}). \quad (1)$$

The regret in the OCO framework implies that: if an algorithm incurs regret that grows sub-linearly with time, its performance loss tends to zero when averaged over an infinite time horizon T .

Under the OCO framework, many OCO algorithms have been developed (Zinkevich, 2003; Hazan et al., 2007; Zhang et al., 2019; Wang et al., 2020), which are designed based on a scalar loss f_t revealed by an adversary, i.e., single-objective cases. However, we often need to consider multiple objectives simultaneously in real-world applications. For example, in natural language processing (Li et al., 2021; Chen et al., 2021), multiple objectives such as sequence tagging, classification, and text generation, are required to be optimized simultaneously. In autonomous driving (Chen et al., 2018; Ravindran et al., 2020), the vehicle needs to solve object detection and distance prediction at the same time. In these cases, when the revealed losses are vectors and convex, these problems are called Online Multi-objective Convex Optimization (OMCO) problems.

To address OMCO problems, most existing work transforms the multi-objective optimization problem into a single-objective form by using scalarization techniques for further analysis. The most common scalarization method is to combine multiple objectives linearly (Cesa-Bianchi et al., 2022; Achddou et al., 2024; Busa-Fekete et al., 2017; Liu et al., 2024). Besides, methodologies based on

054 ϵ -constraint (Mahdavi et al., 2013) and reference point techniques have been investigated (Lyu et al.,
 055 2019; 2024). However, for the above algorithms, the single-objective functions obtained by different
 056 scalarization techniques and the counterpart regret forms are different. These regrets have different
 057 meanings for an OMCO problem and the obtained regret bounds cannot be compared. In addition,
 058 these regret cannot adequately reflect the trade-off relationship and the interaction among multiple
 059 objectives.

060 In fact, unlike a single-objective optimization problem, a multi-objective optimization problem has
 061 no unique optimal solution, but a set of efficient solutions that do not dominate each other (Ehrgott,
 062 2005; Caramia et al., 2020). The image of these solutions in the objective space is called the
 063 non-dominated set. There are a few works that have considered OMCO problems from a multi-
 064 objective perspective (Blackwell, 1956; Mannor et al., 2014; Tiedemann et al., 2015; Turgay et al.,
 065 2018). Recently, Jiang et al. (2023) proposed a OMCO framework and a multi-objective regret
 066 called Sequence-wise Pareto Suboptimality Gap (S-PSG). S-PSG measures the discrepancy from
 067 the cumulative loss of the online decisions to the non-dominated set. However, S-PSG restricted to
 068 be non-negative. It needs to be noted that the regret in the OCO framework may be less than zero
 069 if a decision maker happens to make an optimal decision at each round. Considering the unifor-
 070 mity of the online convex optimization framework for any number of objectives, the non-negativity
 071 of S-PSG is inconsistent with the regret (which can be negative) defined in the OCO framework.
 072 This raises a natural question: *Can we develop a generalized OMCO framework with a designed*
 073 *multi-objective regret such that it has the following properties?*

- 074 • The designed multi-objective regret can reflect the relationship between the accumulated
- 075 vector loss and the non-dominated set of an OMCO problem in hindsight.
- 076
- 077 • The OMCO framework is a generalized version of OCO, i.e., when the number of objec-
- 078 tives decreases to one, the multi-objective regret degenerates into the regret of OCO.

079 In this paper, we give a positive answer to the above question. We define a novel multi-objective
 080 regret, which is the difference between the cumulative loss vector and the (weakly) non-dominated
 081 set along the direction of e vector, where e is a vector in which each entry is 1. This difference
 082 can be obtained by solving a Translative Scalarization (TS) problem (Pascoletti & Serafini, 1984).
 083 We prove that this multi-objective regret degenerates to the regret in the OCO framework when the
 084 number of objectives is 1, thus unifying the online convex optimization framework for any number
 085 of objectives. To further support the analysis of OMCO algorithms, we derive an equivalent form of
 086 the regret in the OMCO framework using the strong duality theory of convex optimization. Further-
 087 more, we consider minimizing the worst-case objective value and propose an Online Joint Gradient
 088 Descent (OJGD) algorithm. In the algorithm, we introduce weight variables for instantaneous loss
 089 functions to construct a joint gradient descent direction, and the weights are updated by an ascent
 090 step. Compared to the methods based on minimizing the (regularized) norm of the joint gradient
 091 (Jiang et al., 2023; Désidéri, 2012; Sener & Koltun, 2018), our proposed algorithm is easier to im-
 092 plement and updates the weights in linear time. We prove that the OJGD algorithm guarantees a
 093 sublinear multi-objective regret.

094 To validate the effectiveness of our proposed algorithm, we provide experimental evaluations on
 095 two classical machine learning convex problem models (i.e., linear regression and logistic regression
 096 problem models). Furthermore, to verify OJGD’s generalizability, we conduct experiments on multi-
 097 task learning (MTL) networks with non-convex objective functions, which essentially represent non-
 098 convex multi-objective optimization problems (Sener & Koltun, 2018; Liu et al., 2019). We use
 099 numerical results to show the superior performance of our proposed algorithm on several real-world
 100 datasets. Our main contributions are as follows.

- 101 • **Unified optimization framework.** We establish an OMCO framework with a novel multi-
- 102 objective regret and prove that the proposed novel regret degenerates to the regret in the
- 103 OCO framework when the number of objectives is equal to one, thereby unifying the online
- 104 convex optimization framework for any number of objectives.
- 105
- 106 • **OMCO algorithm with robustness.** Considering minimizing the worst-case objective
- 107 value, we propose an OMCO algorithm called Online Joint Gradient Descent, which is
- easy to implement and is proved to guarantee a sublinear multi-objective regret.

- **Experiments on practical applications.** We provide empirical evaluations of our proposed algorithm on several real-world datasets. The results show that our algorithm consistently outperforms other OMCO algorithms.

2 RELATED WORKS

Online multi-objective convex optimization. Among those methods that transform the OMCO problem into single-objective forms, the most straightforward method is to sum multiple objectives (Cesa-Bianchi et al., 2022; Achddou et al., 2024). Besides, weighted sum methods have been studied to convexly combine multiple objectives into a single objective (Busa-Fekete et al., 2017; Liu et al., 2024). Mahdavi et al. (2013) used the ϵ -constraint method (Ehrgott, 2005) to choose a primary objective as the objective function with other objectives bounded by thresholds. Based on an expected point in the objective space, Lyu et al. (2019; 2024) established a quadratic programming problem to minimize the distance between the average cumulative loss and the expected point. Furthermore, a few works have studied OMCO problems from a multi-objective perspective. Blackwell (1956) and Mannor et al. (2014) considered that the learner’s goal is to make the average vectorial loss function converge to a closed target set known in advance. However, for OMCO problems, the target set is a set of the (weakly) non-dominated points, which is unknown in advance. Tiedemann et al. (2015) introduced a framework for the competitive analysis of OMCO algorithms. Turgay et al. (2018) proposed a metric called Pareto Suboptimality Gap (PSG) for multi-objective contextual bandit problems, where the PSG is restricted to be non-negative. Jiang et al. (2023) generalized PSG to OMCO problems and defined the S-PSG multi-objective regret of OMCO. The S-PSG is also restricted to a non-negative domain, which exposes a fundamental inconsistency with the regret in the OCO framework, allowing the sign to be unrestricted. In this paper, we propose a unified online convex optimization framework for any number of objectives.

Multi-objective optimization. There are two classes of multi-objective optimization (MO) methods that are relevant to our work, i.e., translative scalarization methods (Pascoletti & Serafini, 1984; Benson & Sayin, 1997) and gradient-based methods (Fliege & Svaiter, 2000; Povalej, 2014). In a translative scalarization method, a single-objective optimization problem is constructed based on a given reference point and a projection direction in the objective space. By solving the problem, the projection of this reference point on the (weakly) non-dominated set is obtained along this projection direction. Many MO methods have been designed based on this technique (Das & Dennis, 1998; Ismail-Yahaya & Messac, 2002; Shao & Ehrgott, 2016). In this paper, we design a novel multi-objective regret based on this technique. Gradient-based methods were designed to solve gradient conflict problems in MO (Désidéri, 2012; Sener & Koltun, 2018; Yu et al., 2020; Liu et al., 2021). The min-norm method (Désidéri, 2012; Sener & Koltun, 2018), which obtains the gradient weights by minimizing the norm of the joint gradient, is a foundational algorithm in this regard. However, most of the above MO methods are designed for offline settings and non-adversarial MO problems. In recent years, Jiang et al. (2023) extended such a min-norm method to OMCO problems and proposed the Doubly Regularized Online Mirror Multiple Descent (DR-OMMD) algorithm. The DR-OMMD algorithm obtains the weights of the joint gradient by solving a quadratic optimization problem with a regularization term, which incurs substantial computational overhead in practice. In this paper, our proposed OJGD uses an ascent step to update the weights of the joint gradient, which can be implemented in linear time.

3 MULTI-OBJECTIVE CONVEX OPTIMIZATION

Throughout the paper vectors are column vectors written in boldface. For a vector $\mathbf{x} \in \mathbb{R}^p$, we write x^i to denote the i -th component of the vector \mathbf{x} . We denote by $\mathbf{x}^T \mathbf{y}$ the inner product of two vectors \mathbf{x} and \mathbf{y} . We use $\|\mathbf{x}\|$ to denote the standard Euclidean norm, i.e., $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$. For $p \in \mathbb{N}$, we denote by $[p]$ as the set $\{1, 2, \dots, p\}$. We denote by $\mathbf{e} \in \mathbb{R}^p$ as a vector in which each entry is 1. Furthermore, \mathbf{e}_i denotes a vector whose i -th entry is 1 and others are 0. We use $\Pi_X(\mathbf{y})$ to denote the Euclidean projection of a vector \mathbf{y} onto X , i.e., $\Pi_X(\mathbf{y}) = \operatorname{argmin}_{\mathbf{x} \in X} \|\mathbf{x} - \mathbf{y}\|$. Let $\Lambda := \{\boldsymbol{\lambda} \in \mathbb{R}^p \mid \forall i \in [p], \lambda^i \geq 0, \mathbf{e}^T \boldsymbol{\lambda} = 1\}$ and $\mathbb{R}_{\geq}^p := \{\mathbf{x} \in \mathbb{R}^p \mid \forall i \in [p], x^i \geq 0\}$. For a set $S \subseteq \mathbb{R}^p$, we denote the interior of S by $\operatorname{int} S$.

Definition 1. For a set $\mathcal{S} \subseteq \mathbb{R}^p$, we define its non-dominated set \mathcal{S}_N and its weakly non-dominated set \mathcal{S}_{WN} as follows.

$$\mathcal{S}_N := \left\{ \mathbf{y} \in \mathcal{S} : (\{\mathbf{y}\} - \mathbb{R}_{\geq}^p) \setminus \{0\} \cap \mathcal{S} = \emptyset \right\}, \quad \mathcal{S}_{WN} := \left\{ \mathbf{y} \in \mathcal{S} : (\{\mathbf{y}\} - \text{int}\mathbb{R}_{\geq}^p) \cap \mathcal{S} = \emptyset \right\}.$$

Definition 2. For two sets $\mathcal{A} \subseteq \mathbb{R}^p$ and $\mathcal{B} \subseteq \mathbb{R}^p$, their Minkowski sum is defined by $\mathcal{A} \oplus \mathcal{B} = \{\mathbf{x} + \mathbf{y} \mid \mathbf{x} \in \mathcal{A}, \mathbf{y} \in \mathcal{B}\}$. For a set $\mathcal{S} \subseteq \mathbb{R}^p$, the \mathbb{R}_{\geq}^p cone expansion of \mathcal{S} is defined by $\mathcal{S}' := \mathcal{S} \oplus \mathbb{R}_{\geq}^p$.

According to Definitions 1 and 2, we have $\mathcal{S}_N = \mathcal{S}'_N$ and $\mathcal{S}'_N \subseteq \mathcal{S}'_{WN}$. Then, we consider a multi-objective convex optimization (MCO) problem:

$$\min_{\mathbf{x} \in X} \mathbf{F}(\mathbf{x}) = (F^1(\mathbf{x}), F^2(\mathbf{x}), \dots, F^p(\mathbf{x}))^T, \quad (2)$$

where X is a non-empty convex compact set in decision (or variable) space \mathbb{R}^n . $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ is a vector loss composed of p scalar losses, i.e., $\mathbf{F} = (F^1, \dots, F^p)^T$, where $\forall i \in [p], F^i$ is convex and differentiable. The image of the feasible set X under the vector loss \mathbf{F} is denoted as $Y := \mathbf{F}(X) \subset \mathbb{R}^p$, it is called the feasible set in objective space.

The set of optimal solutions for an MCO problem (2) is actually a set of efficient solutions. Specifically, if $\bar{\mathbf{x}} \in X$ is an efficient solution of an MCO problem (2), then for the image $\mathbf{F}(\bar{\mathbf{x}})$ of $\bar{\mathbf{x}}$ in the objective space, there are no other feasible points that can simultaneously improve all objective values of $\mathbf{F}(\bar{\mathbf{x}})$. The mathematical definition of an (weakly) effective solution is as follows.

Definition 3 (Ehrgott, 2005). For an MCO problem (2) and a feasible set X in the decision space, a solution $\mathbf{x} \in X$ is called a (weakly) efficient solution if $(\mathbf{F}(\mathbf{x}) \in Y_{WN}) \mathbf{F}(\mathbf{x}) \in Y_N$. Correspondingly, point $\mathbf{y} = \mathbf{F}(\mathbf{x})$ is called a (weakly) non-dominated point in the objective space, where the set of non-dominated points is called the Pareto front.

The **translative scalarization** (Pascoletti & Serafini, 1984) problem of an MCO problem (2) is

$$\max \delta, \text{ s.t. } \mathbf{q} - \delta \mathbf{e} \geq \mathbf{F}(\mathbf{x}), \text{ s.t. } \mathbf{x} \in X, \delta \in \mathbb{R}, \quad (3)$$

where $\mathbf{q} \in \mathbb{R}^p$ is a given reference point.

Lemma 1 (Pascoletti & Serafini, 1984). Given a reference point $\mathbf{q} \in \mathbb{R}^p$, suppose $\bar{\mathbf{x}} \in X$ is an optimal solution of the translative scalarization problem (3), then $\bar{\mathbf{x}}$ is a weakly efficient solution of an MCO problem (2) and its image in the objective space is a weakly non-dominated point.

In fact, the optimal value δ^* of problem (3) is the difference between \mathbf{q} and Y' along the direction \mathbf{e} and this difference could be either non-negative or negative depending on whether $\mathbf{q} \in Y'$ or not. In the next section, we will use this translative scalarization to establish an OMCO framework.

4 ONLINE MULTI-OBJECTIVE CONVEX OPTIMIZATION

We now gradually formulate the proposed OMCO framework. Similar to the OCO framework for single-objective cases, OMCO can be described as a game process between a learner and an adversary: at each round $t \in [T]$, the learner first makes a decision $\mathbf{x}_t \in X$ for arriving data. Then the adversary reveals a vector $\mathbf{F}_t : X \rightarrow \mathbb{R}^p$ of loss functions, and the learner suffers the vector-valued loss $\mathbf{F}_t(\mathbf{x}_t)$. The learner's goal is to minimize a certain kind of regret.

For an OMCO problem, we are interested in obtaining a (weak) effective solution to the following problem.

$$\min \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) = \left(\sum_{t=1}^T \mathbf{F}_t^1(\mathbf{x}), \dots, \sum_{t=1}^T \mathbf{F}_t^p(\mathbf{x}) \right), \text{ s.t. } \mathbf{x} \in X. \quad (4)$$

In the literature, problem (4) is often transformed into a single-objective online optimization problem with weights $\boldsymbol{\lambda} \in \Lambda$ by the weighted sum method, i.e., $(P_{\boldsymbol{\lambda}}) : \min_{\mathbf{x} \in X} \boldsymbol{\lambda}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x})$. Correspondingly, the weighted sum regret for fixed $\boldsymbol{\lambda}$ is $\text{regret}_T(\boldsymbol{\lambda}) = \boldsymbol{\lambda}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \min_{\mathbf{x} \in X} \boldsymbol{\lambda}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x})$. However, in the weighted sum method, not only a priori knowledge is required to determine the weights, but the trade-off relationship and the interaction among multiple objectives cannot be adequately reflected. To extend the regret of the OCO framework to multi-objective cases, based on the translative scalarization method (3), we define a novel multi-objective regret as follows.

Definition 4. The multi-objective regret in the OMCO framework is

$$R_T = \max_{\mathbf{x} \in X, \delta \in \mathbb{R}} \left\{ \delta : \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \delta \mathbf{e} \geq \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) \right\}. \quad (5)$$

Let Y' be the cone expansion of the feasible set Y in the objective space for an MCO problem (4) and (δ^*, \mathbf{x}^*) be an optimal solution of problem (5). It should be noted that δ^* actually measures the difference between the cumulative loss and Y'_{WN} in the \mathbf{e} -direction. Here, δ^* could be non-negative or negative depending on whether $\sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) \in Y'$ or not. Also, \mathbf{x}^* is a (weakly) efficient solution and $\sum_{t=1}^T \mathbf{F}_t(\mathbf{x}^*)$ is a (weakly) non-dominated point of the problem (4). Moreover, R_T and regret_T have the following relationship.

Theorem 1. When the number of objectives is degenerated to one, R_T is equal to the regret in the OCO framework, i.e., $R_T = \text{regret}_T$, when $p = 1$.

Proof. When $p = 1$, we have $R_T = \max_{\mathbf{x} \in X, \delta \in \mathbb{R}} \{ \delta : \sum_{t=1}^T f_t(\mathbf{x}_t) - \delta \geq \sum_{t=1}^T f_t(\mathbf{x}) \} = \max_{\mathbf{x} \in X} (\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x})) = \sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in X} \sum_{t=1}^T f_t(\mathbf{x}) = \text{regret}_T$. Therefore, Theorem 1 holds. \square

Remark 1. According to Theorem 1, the regret in single-objective cases is actually a degenerate version of the regret in multi-objective cases. Therefore, we can use ' R_T ' to denote the regret of the online convex optimization framework with any number of objectives.

For the form of regret (5) in the OMCO framework, it is difficult to directly guide the design of OMCO algorithms and analyze their properties. Therefore, we next derive its equivalent form as shown in Theorem 2.

Theorem 2. Assume that there exists $\delta_0 \in \mathbb{R}$ and $\mathbf{x}_0 \in \text{int}X$ such that $\sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \delta_0 \mathbf{e} > \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_0)$ holds, then an equivalent form of the multi-objective regret (5) is

$$R_T = \min_{\lambda \in \Lambda} \max_{\mathbf{x} \in X} \lambda^T \left(\sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) \right). \quad (6)$$

Proof. Problem (5) is a convex optimization problem and its Lagrangian dual problem is

$$\min_{\lambda \in \mathbb{R}_{\geq}^p} \max_{\mathbf{x} \in X, \delta \in \mathbb{R}} \delta + \lambda^T \left(\sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \delta \mathbf{e} - \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) \right), \quad (7)$$

where λ is the dual variable. According to the assumption, it is known that Slater's condition is satisfied. Thus, the problem pair (5) and (7) has strong duality. Consequently, the optimal values of (5) and (7) are equal. Furthermore, by the KKT conditions, the derivative of objective function of problem (7) with respect to δ is zero, we have $\lambda \in \Lambda$. Then, problem (7) is equivalent to the right-hand side of (6). Therefore, Theorem 2 holds. \square

Based on the regret form in Theorem 2, we can easily find that this regret is strongly related to the weighted sum form of OMCO problem, which is shown in Proposition 1 and 2.

Proposition 1. Suppose that $(\mathbf{x}^*, \lambda^*)$ is an optimal solution of the above problem (6), if an OMCO problem is transformed into an OCO problem (P_λ) with weights λ^* , then $R_T = \text{regret}_T(\lambda^*)$.

Proof. Since $(\mathbf{x}^*, \lambda^*)$ is the optimal solution of problem (6), we then have $R_T = \min_{\lambda \in \Lambda} \max_{\mathbf{x} \in X} \lambda^T \sum_{t=1}^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x})) = \max_{\mathbf{x} \in X} \lambda^{*T} \sum_{t=1}^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x})) = \lambda^{*T} \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \min_{\mathbf{x} \in X} \lambda^{*T} \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) = \text{regret}_T(\lambda^*)$. \square

Proposition 2. For an OMCO problem, which is transformed into an OCO problem (P_λ) with weights $\bar{\lambda} \in \Lambda$, if an OCO algorithm achieves a sublinear bound for the weighted sum regret $\text{regret}_T(\bar{\lambda})$, then this algorithm also guarantees a sublinear multi-objective regret in the OMCO framework.

270 *Proof.* Let $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ be an optimal solution of problem (6), from Proposition 1, we have $R_T =$
 271 $\boldsymbol{\lambda}^{*T} \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \min_{\mathbf{x} \in X} \boldsymbol{\lambda}^{*T} \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) \leq \bar{\boldsymbol{\lambda}}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \min_{\mathbf{x} \in X} \bar{\boldsymbol{\lambda}}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) =$
 272 $\text{regret}_T(\bar{\boldsymbol{\lambda}}).$ \square
 273

274 Similarly, it is easy to establish a multi-objective regret with respect to a time-varying comparator
 275 sequence in dynamic settings (see Section C for details).
 276

277 5 ONLINE JOINT GRADIENT DESCENT ALGORITHM

278 In this section, we try to minimize the worst-case objective value and design an online joint gradient
 279 descent algorithm for OMCO problems, as well as provide its a multi-objective regret bound.

280 5.1 ALGORITHM DEVELOPMENT

281 To resolve potential conflicts among objectives in problem (4) from a global perspective, we use a
 282 robustness-driven method that tries to minimize the worst-case objective value as follows.

$$283 \min_{\mathbf{x} \in X} \max \left\{ \sum_{t=1}^T F_t^1(\mathbf{x}), \dots, \sum_{t=1}^T F_t^p(\mathbf{x}) \right\}. \quad (8)$$

284 **Lemma 2.** *An optimal solution \mathbf{x}^* of the problem (8) is a weakly efficient solution of problem (4).*

285 *Proof.* We prove Lemma 2 by contradiction. Since \mathbf{x}^* is an optimal solution of problem (8),
 286 then for all $\mathbf{x} \in X$, we have $\max_{i \in [p]} \sum_{t=1}^T F_t^i(\mathbf{x}^*) \leq \max_{i \in [p]} \sum_{t=1}^T F_t^i(\mathbf{x})$. Suppose \mathbf{x}^* is
 287 not a weakly efficient solution of the problem (4), then there exists another solution $\bar{\mathbf{x}} \in X$
 288 such that $\sum_{t=1}^T F_t^i(\bar{\mathbf{x}}) < \sum_{t=1}^T F_t^i(\mathbf{x}^*)$ for all i . This means that $\max_{i \in [p]} \sum_{t=1}^T F_t^i(\bar{\mathbf{x}}) <$
 289 $\max_{i \in [p]} \sum_{t=1}^T F_t^i(\mathbf{x}^*)$, which contradicts with $\max_{i \in [p]} \sum_{t=1}^T F_t^i(\mathbf{x}^*) \leq \max_{i \in [p]} \sum_{t=1}^T F_t^i(\mathbf{x})$.
 290 Therefore, Lemma 2 holds. \square
 291

292 To better facilitate the design of OMCO algorithm, we introduce weight factors to transform problem
 293 (8) into the following form.

$$294 \min_{\mathbf{x} \in X} \max_{\boldsymbol{\lambda} \in \Lambda} \boldsymbol{\lambda}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}). \quad (9)$$

295 We further establish the relationship between the problems (8) and (9) according to the following
 296 lemma, the proof of which is provided in Section D.

297 **Lemma 3.** *An optimal solution of the problem (9) is an optimal solution of the problem (8).*

298 According to Lemma 2 and 3, by solving problem (9), we can obtain a solution that performs
 299 well for the worst-case in the multi-objective trade-off. As a result, this solution is ensured to
 300 be weakly non-dominated and robust over the entire feasible domain in the objective space. For this
 301 minimax problem (9), let its instantaneous optimization function be $\mathcal{L}_t(\mathbf{x}, \boldsymbol{\lambda}) := \boldsymbol{\lambda}^T \mathbf{F}_t(\mathbf{x})$. Then
 302 we introduce the online joint gradient descent algorithm, which takes a descent step to update the
 303 decision $\mathbf{x} \in X$ and an ascent step to update the weights $\boldsymbol{\lambda} \in \Lambda$ at each t -th round, i.e.,

$$304 \mathbf{x}_{t+1} = \Pi_X(\mathbf{x}_t - \eta_t^p \nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)), \quad \boldsymbol{\lambda}_{t+1} = \Pi_{\Lambda}(\boldsymbol{\lambda}_t + \eta_t^d (\nabla_{\boldsymbol{\lambda}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) + \alpha_t \Delta(\boldsymbol{\lambda}_t))),$$

305 where $\nabla_{\mathbf{x}} \mathcal{L}_t$ and $\nabla_{\boldsymbol{\lambda}} \mathcal{L}_t$ are the gradients of \mathcal{L}_t with respect to \mathbf{x} and $\boldsymbol{\lambda}$, respectively. $\Delta(\boldsymbol{\lambda}_t) :=$
 306 $\frac{\boldsymbol{\lambda}_t - \bar{\boldsymbol{\lambda}}}{\|\boldsymbol{\lambda}_t - \bar{\boldsymbol{\lambda}}\|}$ is a regularization term used to avoid the weights drifting over time. η_t^p and η_t^d are the
 307 decision stepsize and weights stepsize, respectively. α_t is a trade-off factor that is used to balance
 308 exploration and convergence. In fact, letting $\nabla \mathbf{F}_t(\mathbf{x})$ be the Jacobian matrix of $\mathbf{F}_t(\mathbf{x})$, then
 309 $\nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) = \nabla \mathbf{F}_t(\mathbf{x}_t)^T \boldsymbol{\lambda}_t$ is a joint gradient and all objective functions decrease along the joint
 310 gradient direction. In addition, the decision stepsize is actually the learning rate in applications. The
 311 complete algorithm is summarized in Algorithm 1.
 312

Algorithm 1 Online Joint Gradient Descent (OJGD)

Require: X, T , learning rates $\{\eta_t^p\}$, weights stepsizes $\{\eta_t^d\}$, $\mathbf{x}_1 \in X, \boldsymbol{\lambda}_1 \in \Lambda$.

- 1: **for** $t = 1$ to T **do**
- 2: Predict \mathbf{x}_t and then receive \mathbf{F}_t
- 3: Suffer loss $\mathbf{F}_t(\mathbf{x}_t)$, and observe $\nabla \mathbf{F}_t(\mathbf{x}_t)$
- 4: Update the decisions:

$$\mathbf{x}_{t+1} = \Pi_X(\mathbf{x}_t - \eta_t^p \nabla \mathbf{F}_t(\mathbf{x}_t)^T \boldsymbol{\lambda}_t) \quad (10)$$

- 5: Update the weights: $\Delta(\boldsymbol{\lambda}_t) = \mathbf{0}$ if $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_1$ and $\Delta(\boldsymbol{\lambda}_t) = \frac{\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_t}{\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_t\|}$ otherwise, then

$$\boldsymbol{\lambda}_{t+1} = \Pi_\Lambda(\boldsymbol{\lambda}_t + \eta_t^d (\mathbf{F}_t(\mathbf{x}_t) + \alpha_t \Delta(\boldsymbol{\lambda}_t))) \quad (11)$$

- 6: **end for**

It's worth noting that the projection operation for the update of weights in Algorithm 1 is a projection onto a simplex, it is straightforward to implement in linear time (Duchi et al., 2008). Therefore, the OJGD algorithm is easier to implement and takes less time than the methods based on minimizing (regularized) the joint gradient (Jiang et al., 2023; Désidéri, 2012; Sener & Koltun, 2018), especially in complex problems, which is verified in Section 6.

5.2 PERFORMANCE ANALYSIS

We provide performance analyses of Algorithm 1 based on two basic assumptions (Hazan, 2022).

Assumption 1. *The domain $X \subset \mathbb{R}^n$ is convex and bounded, i.e., $\max_{\mathbf{x}_1, \mathbf{x}_2 \in X} \|\mathbf{x}_1 - \mathbf{x}_2\| \leq D$.*

Assumption 2. *The gradients of all loss functions are convex and bounded by a constant G , i.e., $\forall t \in [T], i \in [p], \|\nabla F_t^i(\mathbf{x})\| \leq G$.*

Theorem 3. *Suppose Assumptions 1 and 2 hold, assume F_t^i is bounded, i.e., $\forall \mathbf{x} \in X, t \in [T], i \in [p], |F_t^i(\mathbf{x})| \leq F$, set the trade-off factor $\alpha_t = Ft^\tau$ and the directional gain factor $\tau \geq 0$, the OJGD attains the following multi-objective regret*

$$R_T \leq \frac{D^2}{2\eta_T^p} + \frac{G^2}{2} \sum_{t=1}^T \eta_t^p + \frac{\sqrt{2}}{2\eta_T^d} + 2pF^2 \sum_{t=1}^T \eta_t^d t^{2\tau}.$$

Remark 2. *When stepsizes $\eta_t^p = \frac{D}{G\sqrt{t}}$, $\eta_t^d = \frac{1}{F\sqrt{t}}$ and $\tau \in [0, 0.25)$, the OJGD achieve a sublinear multi-objective regret bound. The proof of Theorem 3 is provided in Section D. Moreover, we provide the regret analysis of OJGD under the strongly convex and dynamic settings in Sections D.*

Dynamic phase properties of α_t . According to the setting of α_t in Theorem 3, the dynamic phase properties of the trade-off factor α_t have an adaptive adjustment mechanism. Specifically, for the optimization direction $(\mathbf{F}_t(\mathbf{x}_t) + \alpha_t \Delta(\boldsymbol{\lambda}_t))$ of $\boldsymbol{\lambda}_{t+1}$ in (11), the first term $\mathbf{F}_t(\mathbf{x}_t)$ reflects the performance of each objective at t -th round, the second term $\Delta(\boldsymbol{\lambda}_t)$ reflects the orientation of weights $\boldsymbol{\lambda}_t$ to weights $\boldsymbol{\lambda}_1$. Since $\alpha_t = Ft^\tau$ and $\tau \in [0, 0.25)$, in the early stages of the iteration, OJGD improves the underperforming objectives while the optimization direction is also influenced by the initial weights, i.e., it is in the exploration stage. As the iterative process evolves, the weights of the joint gradient are guided by the initial weights to ensure the asymptotic stability of the (weakly) efficient solution, i.e., OJGD gradually transitions to the convergence stage.

6 EXPERIMENTS

In this section, we validate the effectiveness of our algorithm on classical convex machine learning models and verify its generalizability on non-convex models. We conduct experiments to compare OJGD with three baselines: (i) uniform scaling: linearization performs single-objective online learning on scalar losses with equal weights; (ii) min-norm: obtaining the joint gradient via min-norm of gradient composition (Désidéri, 2012; Sener & Koltun, 2018); (iii) DR-OMMD: using weights for the gradient composition via min-regularized-norm (Jiang et al., 2023). For each experiment, initial weights are set as equal weights. More experimental results and details are deferred to Section E.

378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431

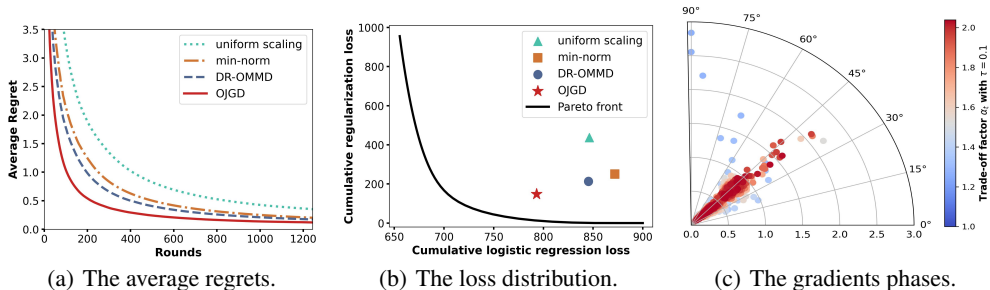


Figure 1: The results on the svmguide3 dataset. Figure 1(a) plots the time-average multi-objective regrets of all the algorithms; Figure 1(b) shows the distribution of cumulative loss values for the algorithms in the objective space and the Pareto front of the OMCO problem; the phases of the joint gradients of the OJGD algorithm during the optimization process are plotted in Figure 1(c).

6.1 CONVEX EXPERIMENTS

Linear regression and logistic regression have well-studied convex objectives. In machine learning, the loss function usually combines an empirical loss and a regularization term, which often conflict. Specifically, minimizing empirical loss may lead to overfitting, while strong regularization may lead to underfitting of the data. Thus, model training can be formulated as a multi-objective optimization problem, treating the empirical loss and regularization as two separate objectives that need to be balanced. We use **mg** and **svmguide3** datasets from the UCI Machine Learning repository (Dua et al., 2017). For the prediction/classification problem, we use linear regression/logistic regression loss as the first objective function and squared the L2-norm of the model parameters as the second objective function. We restrict the model parameters to a sphere X with a radius $R = 10$.

In Figure 1(a), we plot the time-average regret R_T/T curve on the svmguide3 dataset. The results show that OJGD achieves the lowest regret among all the algorithms. Figure 1(b) plots the cumulative losses obtained by the algorithms and the Pareto fronts of the OMCO problems at T -th round, where the Pareto fronts are obtained by the RNBI method (Shao & Ehrgott, 2016). This figure intuitively shows that the cumulative loss obtained by OJGD is closer to the Pareto front than those obtained by other algorithms and reflects that OJGD can better trade off the two objectives.

Furthermore, we show the phase of the joint gradients under the variation of the trade-off factor α_t during the optimization process in Figure 1(c). Specifically, for the t -th round, the phase θ_t is obtained from the joint gradient weights via $\theta_t = \arctan 2(\lambda_t^2, \lambda_t^1)$, and the radius r_t is the norm of the joint gradient, i.e., $r_t = \|\nabla F_t(\mathbf{x}_t)^T \lambda_t\|$. The colors of the points in Figure 1(c) are determined by the size of α_t , reflecting the optimization progress. From Figure 1(c), OJGD explores various optimization directions in the early stage. As the trade-off factor α_t increases, the phase of the joint gradient is guided by the initial weight direction and is gradually pulled back to the initial weight direction, and the norm of the joint gradient gradually becomes smaller. The above variation process of the joint gradient phase reflects the evolution of OJGD from the exploration stage to the convergence stage.

6.2 NON-CONVEX EXPERIMENTS

We use standard multi-task supervised learning datasets: MultiMNIST (Sabour et al., 2017) and NYU-v2 (Silberman et al., 2012). The MultiMNIST dataset contains 2 tasks: simultaneously classifying the digit on the top-left (task-L) and the bottom-right (task-R) separately. The NYU-v2 dataset contains 3 tasks: 13-class semantic segmentation, depth estimation, and surface normal estimation. For MultiMNIST, we follow the setup of Sener & Koltun (2018) and use the LeNet network architecture. For NYU-v2, we run all experiments using the MTL benchmark framework LibMTL (Lin & Zhang, 2023) with the MTAN (Liu et al., 2019) network architecture.

For MultiMNIST, Figure 2 shows the distribution of the average cumulative losses obtained by the algorithms in the objective space and accuracy profile in detecting the left and right digits for the algorithms. Specifically, the train losses are plotted in Figure 2(a), the test losses are plotted in

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485

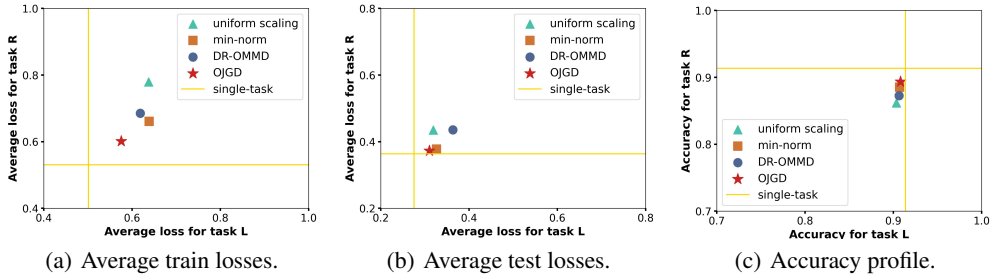


Figure 2: The results on the MultiMNIST dataset. Figure 2(a) and 2(b) plot the distribution of the average cumulative train losses and test losses in the objective space of all the algorithms; Figure 2(c) shows the distribution of accuracy in detecting the left and right digits for all the algorithms.

Figure 2(b), and the accuracy profile is shown in Figure 2(c). Since it is difficult to obtain the Pareto front of this non-convex problem model, we consider constructing the ideal point of this problem to compare the algorithms’ results more clearly, where the ideal point consists of the optimal value of each objective. We use the **single-task** (i.e., online solving tasks independently) as an added baseline. The vector consisting of the single-task’s results can be approximated as an ideal point, i.e., the intersection of the two yellow lines in Figure 2. From Figure 2, OJGD’s results are both closer to the approximate ideal points constituted by the single-task’s results than the results of the other algorithms.

For NYU-v2, we follow the evaluation metrics in Liu et al. (2019) for the three tasks. Moreover, we follow the work of Maninis et al. (2019) and report the overall performance metric $\Delta M\%$ by computing the average per-task performance drop of an algorithm a versus the single-task baseline b . Specifically, $\Delta M\% = \frac{1}{N_m} \sum_{n=1}^{N_m} (-1)^{l_n} (m_{a,n} - m_{b,n}) / m_{b,n} \times 100$, where N_m is the number of metrics, $m_{a,n}$ and $m_{b,n}$ are respectively the n -th metric values for algorithms a and single-task b , $l_n = 1$ if higher n -th metric is better and 0 otherwise. In addition, to measure the performance efficiency per unit time of each algorithm, we report the average running time Δt of each algorithm for a single epoch and the Performance-Time (P-T) ratio, i.e., $\frac{1/\Delta M\%}{\Delta t} \times 100$. We report the above metrics in Table 1, which shows that OJGD obtains better or competitive results than other algorithms. Furthermore, OJGD has the best value of P-T ratio, which means the overall performance is the best over the same time.

Table 1: The results on the NYU-v2 dataset. The mean of each metric (\uparrow : higher better; \downarrow : lower better) is reported over 3 independent runs. The best average result is marked in bold.

Method	Segmentation		Depth		Surface Normal					$\Delta M\% \downarrow$	$\Delta t(s) \downarrow$	P-T \uparrow
	mIoU \uparrow	Pix Acc \uparrow	Abs Err \downarrow	Rel Err \downarrow	Angle Distance \downarrow		Within $t^\circ \uparrow$					
					Mean	Median	11.25	22.5	30			
single-task	52.89	74.69	0.4185	0.1706	21.81	14.72	40.03	65.95	75.74			
uniform scaling	52.88	74.79	0.3848	0.1565	22.98	16.32	36.46	62.46	73.20	1.93	73.18	0.7351
min-norm	45.87	69.56	0.4172	0.1707	21.86	15.07	39.18	65.24	75.39	2.92	278.93	0.1247
DR-OMMD	51.57	73.85	0.3932	0.1623	22.47	15.76	37.55	63.79	74.62	1.53	281.45	0.2339
OJGD	53.11	75.04	0.3812	0.1552	22.98	16.39	37.12	63.01	73.47	1.41	78.26	0.9130

7 CONCLUSIONS

In this paper, we extended the OCO framework to multi-objective cases and established an OMCO framework, which unifies the online convex optimization framework for any number of objectives. Additionally, we derived an equivalent form of multi-objective regret. Then, we proposed an Online Joint Gradient Descent (OJGD), which is proved to guarantee a sublinear regret bound. Finally, numerical experiments consistently demonstrate that OJGD outperforms the other algorithms across multiple real-world datasets. While our analysis is mainly based on convex objectives, an interesting direction is to consider non-convex objectives under an online multi-objective optimization setup.

486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539

ETHICS STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REPRODUCIBILITY STATEMENT

The implementation of the algorithm proposed in this paper and all experiments are reproducible. We have uploaded the anonymized source code to the supplementary materials. The assumptions and proofs of the lemmas and theorems are detailed in Section 5 and D. The dataset sources and experimental settings are fully described in Section 6 and E.

REFERENCES

- Juliette Achddou, Nicolò Cesa-Bianchi, and Pierre Laforgue. Multitask online learning: Listen to the neighborhood buzz. In *International Conference on Artificial Intelligence and Statistics*, pp. 1846–1854. PMLR, 2024.
- Evripidis Bampis, Dimitris Christou, Bruno Escoffier, and Kim Thang NGuyen. Online learning for min-max discrete problems. *Theoretical Computer Science*, 930:209–217, 2022.
- Harold P Benson and Serpil Sayin. Towards finding global representations of the efficient set in multiple objective mathematical programming. *Naval Research Logistics (NRL)*, 44(1):47–67, 1997.
- David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 1956.
- Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, and Shie Mannor. Multi-objective bandits: Optimizing the generalized gini index. In *International Conference on Machine Learning*, pp. 625–634. PMLR, 2017.
- Massimiliano Caramia, Paolo Dell’Olmo, Massimiliano Caramia, and Paolo Dell’Olmo. Multi-objective optimization. *Multi-objective management in freight logistics: Increasing capacity, service level, sustainability, and safety with optimization algorithms*, pp. 21–51, 2020.
- Nicolò Cesa-Bianchi, Pierre Laforgue, Andrea Paudice, and Massimiliano Pontil. Multitask online mirror descent. *Transactions on Machine Learning Research*, 2022(9):1–30, 2022.
- Shijie Chen, Yu Zhang, and Qiang Yang. Multi-task learning in natural language processing: An overview. *arXiv preprint arXiv:2109.09138*, 2021.
- Yaran Chen, Dongbin Zhao, Le Lv, and Qichao Zhang. Multi-task learning for dangerous object detection in autonomous driving. *Information Sciences*, 432:559–571, 2018.
- Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, 8(3):631–657, 1998.
- Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathématique*, 350(5-6):313–318, 2012.
- Dheeru Dua, Casey Graff, et al. Uci machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>, 2017.
- John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l_1 -ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pp. 272–279, 2008.
- Matthias Ehrgott. *Multicriteria optimization*, volume 491. Springer Science & Business Media, 2005.

- 540 Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathe-*
541 *matical methods of operations research*, 51:479–494, 2000.
- 542
- 543 Kartik Gupta, Arun Sai Suggala, Adarsh Prasad, Praneeth Netrapalli, and Pradeep Ravikumar.
544 Learning minimax estimators via online learning. *arXiv preprint arXiv:2006.11430*, 2020.
- 545 Elad Hazan. *Introduction to Online Convex Optimization*. MIT Press, 2022.
- 546
- 547 Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex
548 optimization. *Machine Learning*, 69(2):169–192, 2007.
- 549 Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey.
550 *Neurocomputing*, 459:249–289, 2021.
- 551
- 552 Amir Ismail-Yahaya and Achille Messac. Effective generation of the pareto frontier using the normal
553 constraint method. In *40th AIAA Aerospace Sciences Meeting & Exhibit*, pp. 178, 2002.
- 554 Jiyan Jiang, Wenpeng Zhang, Shiji Zhou, Lihong Gu, Xiaodong Zeng, and Wenwu Zhu. Multi-
555 objective online learning. In *The Eleventh International Conference on Learning Representations*,
556 2023.
- 557 Daniel Lee, Georgy Noarov, Mallesh Pai, and Aaron Roth. Online minimax multiobjective opti-
558 mization: Multicalibeating and other applications. *Advances in Neural Information Processing*
559 *Systems*, 35:29051–29063, 2022.
- 560
- 561 Jianquan Li, Xiaokang Liu, Wenpeng Yin, Min Yang, Liqun Ma, and Yaohong Jin. Empirical
562 evaluation of multi-task learning in deep neural networks for natural language processing. *Neural*
563 *Computing and Applications*, 33:4417–4428, 2021.
- 564 Baijiong Lin and Yu Zhang. Libmtl: A python library for deep multi-task learning. *Journal of*
565 *Machine Learning Research*, 24(209):1–7, 2023.
- 566
- 567 Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent
568 for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890,
569 2021.
- 570 Meitong Liu, Xiaoyuan Zhang, Chulin Xie, Kate Donahue, and Han Zhao. Online mirror descent
571 for tchebycheff scalarization in multi-objective optimization. *arXiv preprint arXiv:2410.21764*,
572 2024.
- 573 Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention.
574 In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp.
575 1871–1880, 2019.
- 576
- 577 Guodong Lyu, Wang Chi Cheung, Chung-Piaw Teo, and Hai Wang. Multi-objective online ride-
578 matching. *Available at SSRN*, 3356823, 2019.
- 579 Guodong Lyu, Wang Chi Cheung, Chung-Piaw Teo, and Hai Wang. Multiobjective stochastic opti-
580 mization: A case of real-time matching in ride-sourcing markets. *Manufacturing & Service*
581 *Operations Management*, 26(2):500–518, 2024.
- 582
- 583 Mehrdad Mahdavi, Tianbao Yang, and Rong Jin. Stochastic convex optimization with multiple
584 objectives. *Advances in neural information processing systems*, 26, 2013.
- 585 Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multi-
586 ple tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recogni-*
587 *tion*, pp. 1851–1860, 2019.
- 588
- 589 Shie Mannor, Vianney Perchet, and Gilles Stoltz. Approachability in unknown games: Online
590 learning meets multi-objective optimization. In *Conference on Learning Theory*, pp. 339–355.
591 PMLR, 2014.
- 592 Aryan Mokhtari, Shahin Shahrapour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization
593 in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE*
55th Conference on Decision and Control (CDC), pp. 7195–7201. IEEE, 2016.

- 594 Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*,
595 2019.
- 596
- 597 Adriano Pascoletti and Paolo Serafini. Scalarizing vector optimization problems. *Journal of Opti-*
598 *mization Theory and Applications*, 42:499–524, 1984.
- 599 Žiga Povalej. Quasi-newton’s method for multiobjective optimization. *Journal of Computational*
600 *and Applied Mathematics*, 255:765–777, 2014.
- 601
- 602 Ratheesh Ravindran, Michael J Santora, and Mohsin M Jamali. Multi-object detection and tracking,
603 based on dnn, for autonomous vehicles: A review. *IEEE Sensors Journal*, 21(5):5668–5677,
604 2020.
- 605 Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances*
606 *in neural information processing systems*, 30, 2017.
- 607
- 608 Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in*
609 *neural information processing systems*, 31, 2018.
- 610 Lizhen Shao and Matthias Ehrgott. Discrete representation of non-dominated sets in multi-objective
611 linear programming. *European Journal of Operational Research*, 255(3):687–698, 2016.
- 612
- 613 Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and sup-
614 port inference from rgb-d images. In *Computer Vision–ECCV 2012: 12th European Conference*
615 *on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pp. 746–760.
616 Springer, 2012.
- 617 Morten Tiedemann, Jonas Ide, and Anita Schöbel. Competitive analysis for multi-objective online
618 algorithms. In *WALCOM: Algorithms and Computation: 9th International Workshop, WALCOM*
619 *2015, Dhaka, Bangladesh, February 26-28, 2015. Proceedings 9*, pp. 210–221. Springer, 2015.
- 620
- 621 Eralp Turgay, Doruk Oner, and Cem Tekin. Multi-objective contextual bandit problem with similar-
622 ity information. In *International Conference on Artificial Intelligence and Statistics*, pp. 1673–
623 1681. PMLR, 2018.
- 624 Guanghui Wang, Shiyin Lu, Yao Hu, and Lijun Zhang. Adapting to smoothness: A more universal
625 algorithm for online convex optimization. In *Proceedings of the AAAI Conference on Artificial*
626 *Intelligence*, volume 34, pp. 6162–6169, 2020.
- 627
- 628 Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn.
629 Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*,
630 33:5824–5836, 2020.
- 631 Lijun Zhang, Tie-Yan Liu, and Zhi-Hua Zhou. Adaptive regret of convex and smooth functions. In
632 *International Conference on Machine Learning*, pp. 7414–7423. PMLR, 2019.
- 633
- 634 Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In
635 *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936,
636 2003.
- 637
- 638
- 639
- 640
- 641
- 642
- 643
- 644
- 645
- 646
- 647

Online Multi-objective Convex Optimization: A Unified Framework and Joint Gradient Descent (Supplementary material)

This supplementary material is organized as follows:

- **Appendix A** introduces more related works.
- **Appendix B** analyzes why multi-objective regret under the domain of real numbers matters.
- **Appendix C** further discusses multi-objective regret of OMCO in a dynamic environment.
- **Appendix D** supplements the proofs omitted for the OJGD algorithm in the main paper.
- **Appendix E** provides more experimental details and results.
- **Appendix F** discloses the use of Large Language Models (LLMs).

A MORE RELATED WORKS

Since our OJGD algorithm is designed based on the minimax problem (9), some intersection works between online optimization methods and minimax problems are also related to our work. There have been some efforts to study the OCO methods to solve minimax problems with time-invariant objective functions (Orabona, 2019; Gupta et al., 2020; Bampis et al., 2022). Moreover, Lee et al. (2022) studied the online minimax problem with vector losses. In the setting of Lee et al. (2022), the learner makes a decision after the loss functions are revealed.

However, our work is largely different from these works in the following aspects. In our OMCO framework, the objective functions (i.e., loss functions) change with time, which is different from the settings in the works of Orabona (2019); Gupta et al. (2020); Bampis et al. (2022). Furthermore, in the setting of the OMCO framework, the loss functions in each round are revealed only after the decision has been made, which is different from the setting in Lee et al. (2022). Therefore, the methods in all the above works cannot handle the minimax problem (9) under the OMCO framework.

B WHY THE MULTI-OBJECTIVE REGRET UNDER THE DOMAIN OF REAL NUMBERS MATTERS

In this section, we will present why the multi-objective regret under the domain of real numbers matters. Specifically, this work has both theoretical value and practical significance. Our decision to extend regret to \mathbb{R} is motivated by three key considerations:

Mathematical Completeness: Prior work (Jiang et al., 2023) restricts a multi-objective regret to $\text{Reg}_T \geq 0$, which introduces a discontinuity between the single-objective setting $\text{Reg}_T \in \mathbb{R}$ and multi-objective regimes. Our unified regret framework resolves this discrepancy (formally proven in Theorem 1) and establishes a more complete theoretical foundation for future algorithm design.

Theoretical Possibility: As mentioned in Section 1, it is theoretically possible to observe negative regret when the algorithm’s decisions x_t happen to coincide with the instantaneous optimal decision of an objective. For instance, in our new synthetic experiments on a two-objective linear regression task (i.e., the following problem (12)), the uniform scaling algorithm indeed achieves a convergent negative regret bound, illustrating that negative values can occur under specific conditions.

Practical Guidance: From our synthetic experiments for problem (12), achieving negative regret correlates strongly with three factors: the problem structure, sample distribution, and the algorithm’s update strategy. These findings suggest that negative regret is not just a theoretical curiosity, but provides practical insights guiding algorithm design tailored to specific problem instances.

Next, we report the setup and results of this synthetic experiment for problem (12).

Negative Regret Experiment Setup and Results: We consider an online two-objective optimization problem:

$$\min_{x \in [0,10]} \left(\sum_{t=1}^T (a_t x - b_t)^2, \sum_{t=1}^T x^2 \right), \quad T = 200, \quad (12)$$

where all data pairs $\forall t \in [T], (a_t, b_t) \in \mathbb{R} \times \mathbb{R}$ are generated before running the experiment. We apply the uniform scaling algorithm with step size $\eta_t = 0.08/\sqrt{t}$, initializing $x_1 = 1$. The following Figure 3 displays the average regret and the distribution of the average cumulative losses in the objective space obtained by the uniform scaling algorithm.

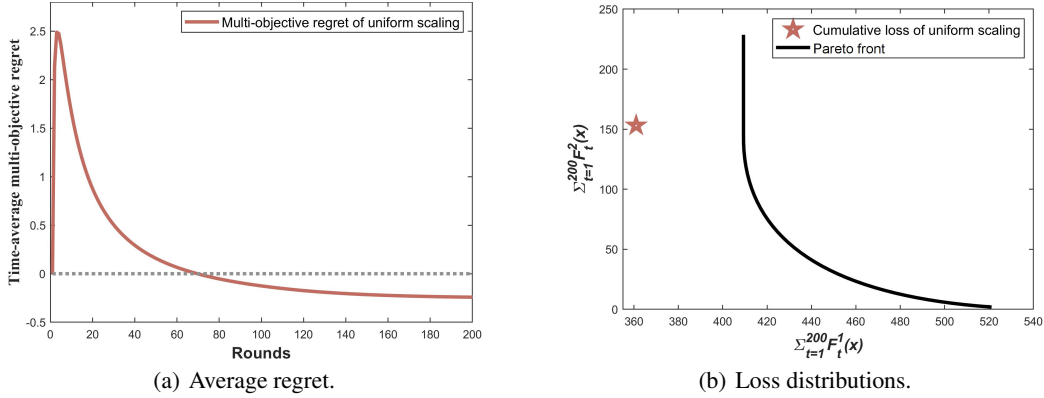


Figure 3: The average regret and the distribution of the average cumulative losses in the objective space obtained by the uniform scaling algorithm.

In this experiment, negative regret occurs because at multiple times, the algorithm’s decision x_t exactly matches the optimal decision of the first objective at the following time. Thus, on average, the incurred loss is lower than that of the hindsight best fixed decision for that objective.

This finding supports our theoretical claim that negative regret is possible under specific conditions, and demonstrates how our regret framework extending into \mathbb{R} can capture such phenomena.

Given the above reasons, relaxing the non-negativity constraint on multi-objective regret is both a significant theoretical contribution and offers practical value.

C DYNAMIC REGRET IN OMCO

The multi-objective regret we proposed in the main paper is analyzed based on a fixed comparator in a static setting, which is also called static regret. Next, we analyze the dynamic regret with respect to time-varying comparators in the OMCO framework. We first review single-objective cases, the dynamic regret in OCO is defined with respect to a time-varying comparator sequence as follows.

$$\text{regret}_T^D = \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T \min_{\mathbf{y}_t \in X} f_t(\mathbf{y}_t). \quad (13)$$

In dynamic settings, for all $t \in [T]$, we set $\mathbf{y}_t^* = \arg \min_{\mathbf{y}_t \in X} f_t(\mathbf{y}_t)$. Then, the regret bounds of algorithms are often related to the path-length of the minimizer sequence $\{\mathbf{y}_t^*\}$, i.e., $L_* = \sum_{t=2}^T \|\mathbf{y}_t^* - \mathbf{y}_{t-1}^*\|$. For example, OGD enjoys an $O(L_* \sqrt{T})$ regret for convex functions (Zinkevich, 2003) and an $O(L_*)$ regret for strongly convex and smooth convex functions (Mokhtari et al., 2016). Now we continue our study of the OMCO framework for dynamic situations.

Definition 5. The dynamic regret in multi-objective cases is

$$R_T^D = \sum_{t=1}^T \max_{\mathbf{y}_t \in X, \delta_t \in \mathbb{R}} \delta_t, \text{ s.t. } \mathbf{F}_t(\mathbf{x}_t) - \delta_t \mathbf{e} \geq \mathbf{F}_t(\mathbf{y}_t), \forall t \in [T]. \quad (14)$$

Similar to static regret, we have the following theorems and propositions.

Theorem 4. When the number of objectives is degenerated to one, R_T^D is equal to the dynamic regret in single-objective case, i.e.,

$$R_T^D = \text{regret}_T^D, \text{ when } p = 1.$$

Proof. When $p = 1$, we have

$$\begin{aligned} R_T^D &= \sum_{t=1}^T \max_{\mathbf{y}_t \in X, \delta_t \in \mathbb{R}} \{\delta_t, \text{ s.t. } f_t(\mathbf{x}_t) - \delta_t \geq f_t(\mathbf{y}_t), t \in [T]\} \\ &= \sum_{t=1}^T \max_{\mathbf{y}_t \in X} (f_t(\mathbf{x}_t) - f_t(\mathbf{y}_t)) \\ &= \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T \min_{\mathbf{y}_t \in X} f_t(\mathbf{y}_t) = \text{regret}_T^D. \end{aligned}$$

Therefore, Theorem 4 holds. \square

Remark 3. Similarly, we can use ' R_T^D ' to denote the dynamic regret of an optimization algorithm for an online convex optimization problem with any number of objectives.

Theorem 5. For each $t \in [T]$, assume that there exists $\delta_t^0 \in \mathbb{R}$ and $\mathbf{x}_0 \in \text{int}X$ such that $\mathbf{F}_t(\mathbf{x}_t) - \delta_t^0 \mathbf{e} > \mathbf{F}_t(\mathbf{x}_0)$ holds, then an equivalent form of the dynamic online multi-objective regret is

$$R_T^D = \sum_{t=1}^T \min_{\lambda_t \in \Lambda} \max_{\mathbf{y}_t \in X} \lambda_t^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{y}_t)). \quad (15)$$

Proof. The problem (14) is a convex optimization problem and its Lagrangian dual problem is as follows,

$$\sum_{t=1}^T \min_{\lambda_t \in \mathbb{R}_{\geq}} \max_{\mathbf{y}_t \in X, \delta_t \in \mathbb{R}} \delta_t + \lambda_t^T (\mathbf{F}_t(\mathbf{x}_t) - \delta_t \mathbf{e} - \mathbf{F}_t(\mathbf{y}_t)), \quad (16)$$

where λ_t is the dual variable.

According to the assumption, it is known that Slater's condition is satisfied. Thus, the problem pair (14) and (16) has strong duality, i.e., their optimal values are equal. Further, by the KKT conditions, the derivative of objective function of problem (16) with respect to δ_t is zero, we have $\lambda_t \in \Lambda$. Then, problem (16) is equivalent to the right-hand side of (15). Therefore, Theorem 5 holds. \square

Next, we introduce a weighted sum method under dynamic settings to give the relationship between our proposed dynamic regret and weighted sum method. We consider a single-objective optimization problem with time-varying weights $\lambda_t \in \Lambda, \forall t \in [T]$ as follows.

$$\min \sum_{t=1}^T \lambda_t^T \mathbf{F}_t(\mathbf{y}_t), \mathbf{y}_t \in X \text{ and } \forall t \in [T]. \quad (17)$$

The the weighted sum dynamic regret is

$$\text{regret}_T^D(\lambda_t) = \sum_{t=1}^T \lambda_t^T \mathbf{F}_t(\mathbf{x}_t) - \sum_{t=1}^T \min_{\mathbf{y}_t \in X} \lambda_t^T \mathbf{F}_t(\mathbf{y}_t). \quad (18)$$

Proposition 3. Suppose that $(\mathbf{y}_t^*, \boldsymbol{\lambda}_t^*)$ is an optimal solution of the above problem (15), if an OMCO problem is transformed into a weighted sum OCO problem (17) with weights $\boldsymbol{\lambda}_t^*$, then

$$R_T^D = \text{regret}_T^D(\boldsymbol{\lambda}_t^*). \quad (19)$$

Proof. Since $(\mathbf{y}_t^*, \boldsymbol{\lambda}_t^*)$ is the optimal solution of problem (15) for t -th round, then we have

$$\begin{aligned} R_T^D &= \sum_{t=1}^T \min_{\boldsymbol{\lambda}_t \in \Lambda} \max_{\mathbf{y}_t \in X} \boldsymbol{\lambda}_t^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x})) \\ &= \sum_{t=1}^T \max_{\mathbf{y}_t \in X} \boldsymbol{\lambda}_t^{*T} (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{y}_t)) \\ &= \sum_{t=1}^T \boldsymbol{\lambda}_t^{*T} \mathbf{F}_t(\mathbf{x}_t) - \sum_{t=1}^T \min_{\mathbf{y}_t \in X} \boldsymbol{\lambda}_t^{*T} \mathbf{F}_t(\mathbf{y}_t) \\ &= \text{regret}_T^D(\boldsymbol{\lambda}_t^*). \end{aligned}$$

□

Proposition 4. For an OMCO problem, which is transformed into an OCO problem (17) with weights $\bar{\boldsymbol{\lambda}}_t \in \Lambda$, if an OCO algorithm achieve sublinear bound for the weighted sum dynamic regret (18), then the OCO algorithm can also achieve sublinear dynamic regret in the OMCO framework.

Proof. Let $(\mathbf{y}_t^*, \boldsymbol{\lambda}_t^*)$ is an optimal solution of problem (15), from Proposition 3, we have

$$\begin{aligned} R_T^D &= \sum_{t=1}^T \boldsymbol{\lambda}_t^{*T} \mathbf{F}_t(\mathbf{x}_t) - \sum_{t=1}^T \min_{\mathbf{y}_t \in X} \boldsymbol{\lambda}_t^{*T} \mathbf{F}_t(\mathbf{y}_t) \\ &\leq \sum_{t=1}^T \bar{\boldsymbol{\lambda}}_t^T \mathbf{F}_t(\mathbf{x}_t) - \sum_{t=1}^T \min_{\mathbf{y}_t \in X} \bar{\boldsymbol{\lambda}}_t^T \mathbf{F}_t(\mathbf{y}_t) \\ &= \text{regret}_T(\bar{\boldsymbol{\lambda}}). \end{aligned}$$

□

D OMITTED PROOFS FOR THE OJGD ALGORITHM

D.1 PROOF OF LEMMA 3

Before proving Lemma 3 1, we introduce the following lemma.

Lemma 4. The problems (8) and (9) have the following relationship: for all $\mathbf{x} \in X$,

$$\max\left\{\sum_{t=1}^T F_t^1(\mathbf{x}), \dots, \sum_{t=1}^T F_t^p(\mathbf{x})\right\} = \max_{\boldsymbol{\lambda} \in \Lambda} \boldsymbol{\lambda}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}).$$

Proof. Suppose that the problem $\max_{i \in [p]} \sum_{t=1}^T F_t^i(\mathbf{x})$ achieves the optimal value at $i = j \in [p]$, i.e., $\max\{\sum_{t=1}^T F_t^1(\mathbf{x}), \dots, \sum_{t=1}^T F_t^p(\mathbf{x})\} = \sum_{t=1}^T F_t^j(\mathbf{x})$. Then for all $\boldsymbol{\lambda} \in \Lambda$, we have $\boldsymbol{\lambda}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) - \sum_{t=1}^T F_t^j(\mathbf{x}) = \boldsymbol{\lambda}^T (\sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) - \mathbf{e} \cdot \sum_{t=1}^T F_t^j(\mathbf{x})) \leq 0$. Therefore, $\forall \boldsymbol{\lambda} \in \Lambda$, $\boldsymbol{\lambda}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) \leq \sum_{t=1}^T F_t^j(\mathbf{x})$. This implies that $\max_{\boldsymbol{\lambda} \in \Lambda} \boldsymbol{\lambda}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) = \sum_{t=1}^T F_t^j(\mathbf{x})$, where an optimal solution $\boldsymbol{\lambda}^* \in \Lambda$ could be taken as $\boldsymbol{\lambda} = \mathbf{e}_j$. Consequently, Lemma 4 holds. □

The proof of Lemma 3 is as follows.

864 *Proof.* Let $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ be an optimal solution of the problem (9). Based on Lemma 4, $\forall \mathbf{x} \in X$, we
 865 have

$$866 \max_{i \in [p]} \left\{ \sum_{t=1}^T F_t^i(\mathbf{x}) \right\} = \max_{\boldsymbol{\lambda} \in \Lambda} \boldsymbol{\lambda}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) \geq \min_{\mathbf{x} \in X} \max_{\boldsymbol{\lambda} \in \Lambda} \boldsymbol{\lambda}^T \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) = \boldsymbol{\lambda}^{*T} \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}^*).$$

870 According to Lemma 4, we have $\boldsymbol{\lambda}^{*T} \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}^*) = \max\{\sum_{t=1}^T F_t^1(\mathbf{x}^*), \dots, \sum_{t=1}^T F_t^p(\mathbf{x}^*)\}$
 871 and can take $\mathbf{x} = \mathbf{x}^*$ to obtain the problem (8)'s optimal value. Thus, Lemma 3 holds. \square

873 D.2 PROOF OF THEOREM 3

874 The proof of Theorem 3 is as follows.

875 *Proof.* Let \mathbf{x}^* be an optimal solution of problem (6) and considering Theorem 2, we have

$$876 R_T \leq \underbrace{\boldsymbol{\lambda}_1^T \left(\sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}^*) \right)}_{\text{Part 1}} + \underbrace{\sum_{t=1}^T (\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_t)^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x}^*))}_{\text{Part 2}}. \quad (20)$$

886 For Part 1 in (20), by using the update rule for \mathbf{x}_{t+1} in (10), we have

$$887 \|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 \leq \|\mathbf{x}_t - \mathbf{x}^*\|^2 + \eta_t^{p2} \|\nabla \mathbf{F}_t(\mathbf{x}_t)^T \boldsymbol{\lambda}_t\|^2 - 2\eta_t^p \boldsymbol{\lambda}_t^T \nabla \mathbf{F}_t(\mathbf{x}_t) (\mathbf{x}_t - \mathbf{x}^*).$$

888 Rearranging the above terms, it follows that

$$889 2\boldsymbol{\lambda}_t^T \nabla \mathbf{F}_t(\mathbf{x}_t) (\mathbf{x}_t - \mathbf{x}^*) \leq \frac{\|\mathbf{x}_t - \mathbf{x}^*\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2}{\eta_t^p} + \eta_t^p \|\nabla \mathbf{F}_t(\mathbf{x}_t)^T \boldsymbol{\lambda}_t\|^2.$$

890 We define $\frac{1}{\eta_0^p} := 0$ and assume the learning rate η_t^p is non-increasing over time. Using the convexity
 891 and summing the above terms from $t = 1$ to T , then

$$892 \begin{aligned} 893 2 \sum_{t=1}^T \boldsymbol{\lambda}_t^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x}^*)) &\leq 2 \sum_{t=1}^T \boldsymbol{\lambda}_t^T \nabla \mathbf{F}_t(\mathbf{x}_t) (\mathbf{x}_t - \mathbf{x}^*) \\ 894 &\leq \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2 \left(\frac{1}{\eta_t^p} - \frac{1}{\eta_{t-1}^p} \right) + G^2 \sum_{t=1}^T \eta_t^p \\ 895 &\leq D^2 \frac{1}{\eta_T^p} + G^2 \sum_{t=1}^T \eta_t^p \end{aligned} \quad (21)$$

896 For Part 2 in (20), by using the update rule for $\boldsymbol{\lambda}_{t+1}$ in (11), we have

$$897 \begin{aligned} 898 \|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_1\|^2 &= \|\Pi_\Lambda(\boldsymbol{\lambda}_t + \eta_t^d (\mathbf{F}_t(\mathbf{x}_t) + \alpha_t \Delta(\boldsymbol{\lambda}_t))) - \boldsymbol{\lambda}_1\|^2 \\ 899 &\leq \|\boldsymbol{\lambda}_t + \eta_t^d (\mathbf{F}_t(\mathbf{x}_t) + \alpha_t \Delta(\boldsymbol{\lambda}_t)) - \boldsymbol{\lambda}_1\|^2 \\ 900 &= \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}_1\|^2 + 2\eta_t^d (\boldsymbol{\lambda}_t - \boldsymbol{\lambda}_1)^T (\mathbf{F}_t(\mathbf{x}_t) + \alpha_t \Delta(\boldsymbol{\lambda}_t)) + (\eta_t^d)^2 \|\mathbf{F}_t(\mathbf{x}_t) + \alpha_t \Delta(\boldsymbol{\lambda}_t)\|^2 \end{aligned}$$

901 Rearranging the above terms, it follows that

$$902 2(\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_t)^T \mathbf{F}_t(\mathbf{x}_t) \leq \frac{\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}_1\|^2 - \|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_1\|^2}{\eta_t^d} + \eta_t^d p(F + \alpha_t)^2 - 2\alpha_t (\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_t)^T \Delta(\boldsymbol{\lambda}_t)$$

Both sides of the above inequality are subtracted from $2(\lambda_1 - \lambda_t)^T \mathbf{F}_t(\mathbf{x}^*)$ at the same time, We then consider the following two cases.

1) case 1: when $\lambda_t = \lambda_1$, we have

$$\begin{aligned} & 2(\lambda_1 - \lambda_t)^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x}^*)) \\ & \leq \frac{\|\lambda_t - \lambda_1\|^2 - \|\lambda_{t+1} - \lambda_1\|^2}{\eta_t^d} + \eta_t^d p(F + \alpha_t)^2 - 2\alpha_t (\lambda_1 - \lambda_t)^T \Delta(\lambda_t) - 2(\lambda_1 - \lambda_t)^T \mathbf{F}_t(\mathbf{x}^*) \\ & = \frac{\|\lambda_t - \lambda_1\|^2 - \|\lambda_{t+1} - \lambda_1\|^2}{\eta_t^d} + \eta_t^d p(F + \alpha_t)^2 \end{aligned}$$

2) case 2: when $\lambda_t \neq \lambda_1$, set $\alpha_t = Ft^\tau$, where $\tau \geq 0$, then $\Delta(\lambda_t) = \frac{\lambda_1 - \lambda_t}{\|\lambda_1 - \lambda_t\|}$ and we have

$$\begin{aligned} & 2(\lambda_1 - \lambda_t)^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x}^*)) \\ & \leq \frac{\|\lambda_t - \lambda_1\|^2 - \|\lambda_{t+1} - \lambda_1\|^2}{\eta_t^d} + \eta_t^d p(F + \alpha_t)^2 - 2\alpha_t (\lambda_1 - \lambda_t)^T \Delta(\lambda_t) - 2(\lambda_1 - \lambda_t)^T \mathbf{F}_t(\mathbf{x}^*) \\ & \leq \frac{\|\lambda_t - \lambda_1\|^2 - \|\lambda_{t+1} - \lambda_1\|^2}{\eta_t^d} + \eta_t^d p(F + \alpha_t)^2 - 2\alpha_t \|\lambda_1 - \lambda_t\| + 2F \|\lambda_1 - \lambda_t\| \\ & = \frac{\|\lambda_t - \lambda_1\|^2 - \|\lambda_{t+1} - \lambda_1\|^2}{\eta_t^d} + \eta_t^d p(F + \alpha_t)^2 + 2\|\lambda_1 - \lambda_t\|(F - \alpha_t) \\ & \leq \frac{\|\lambda_t - \lambda_1\|^2 - \|\lambda_{t+1} - \lambda_1\|^2}{\eta_t^d} + \eta_t^d p(F + \alpha_t)^2 \end{aligned}$$

Thus, combining cases 1 and 2, we have

$$2(\lambda_1 - \lambda_t)^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x}^*)) \leq \frac{\|\lambda_t - \lambda_1\|^2 - \|\lambda_{t+1} - \lambda_1\|^2}{\eta_t^d} + \eta_t^d p(F + \alpha_t)^2.$$

Then, summing the above terms from $t = 1$ to T and set $\alpha_t = Ft^\tau$, where $\tau \geq 0$, then we can obtain an upper bound of Part 2 in (20) as follows.

$$\begin{aligned} 2 \sum_{t=1}^T (\lambda_1 - \lambda_t)^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x}^*)) & \leq \sum_{t=1}^T \frac{\|\lambda_t - \lambda_1\|^2 - \|\lambda_{t+1} - \lambda_1\|^2}{\eta_t^d} + \eta_t^d p(F + \alpha_t)^2 \\ & \leq \sum_{t=2}^T \|\lambda_t - \lambda_1\|^2 \left(\frac{1}{\eta_t^d} - \frac{1}{\eta_{t-1}^d} \right) + 4pF^2 \sum_{t=1}^T \eta_t^d t^{2\tau} \\ & \leq \sqrt{2} \frac{1}{\eta_T^d} + 4pF^2 \sum_{t=1}^T \eta_t^d t^{2\tau} \end{aligned} \quad (22)$$

Finally, according to (21) and (22), we obtain an upper bound on the regret of OJGD for convex functions as shown in Theorem 3. \square

D.3 REGRET ANALYSIS IN THE STRONGLY CONVEX SETTING

In this section, we analyze the multi-objective regret bound of the OJGD algorithm in the strongly convex setting, where the strongly convex function is defined as follows.

Definition 6. A function f is H -strongly convex over the convex set X if $f(\mathbf{x}_1) \geq f(\mathbf{x}_2) + \nabla f(\mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2) + \frac{H}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|^2, \forall \mathbf{x}_1, \mathbf{x}_2 \in X$.

Theorem 6. Suppose Assumptions 1 and 2 hold, assume all functions are H -strongly convex functions and F_t^i is bounded, i.e., $\forall \mathbf{x} \in X, t \in [T], i \in [p], |F_t^i(\mathbf{x})| \leq F$, let stepsizes $\eta_t^p = \frac{1}{Ht}$ and $\eta_t^d = \frac{1}{F\sqrt{t}}$, set $\alpha_t = Ft^\tau$ and $\tau \in [0, 0.25)$, the OJGD attains the following multi-objective regret

$$R_T \leq \frac{G^2}{H} (1 + \log T) + F\sqrt{2T} + \frac{4pF}{2\tau + 0.5} T^{2\tau+0.5}.$$

972 *Proof.* Let \mathbf{x}^* be an optimal solution of problem (5). From Theorem 2, we have

$$\begin{aligned}
973 R_T &= \min_{\lambda \in \Lambda} \max_{\mathbf{x} \in X} \lambda^T \left(\sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}) \right) \\
974 &\leq \lambda_1^T \left(\sum_{t=1}^T \mathbf{F}_t(\mathbf{x}_t) - \sum_{t=1}^T \mathbf{F}_t(\mathbf{x}^*) \right) \\
975 &= \underbrace{\sum_{t=1}^T \lambda_t^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x}^*))}_{\text{Part 1}} + \underbrace{\sum_{t=1}^T (\lambda_1 - \lambda_t)^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x}^*))}_{\text{Part 2}}. \tag{23}
\end{aligned}$$

976 For the above Part 1 and 2 in (23), we further separately prove the upper bounds. For Part 1, based

$$\begin{aligned}
977 \|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 &= \|\Pi_X(\mathbf{x}_t - \eta_t^p \nabla \mathbf{F}_t(\mathbf{x}_t)^T \lambda_t) - \mathbf{x}^*\|^2 \\
978 &\leq \|\mathbf{x}_t - \eta_t^p \nabla \mathbf{F}_t(\mathbf{x}_t)^T \lambda_t - \mathbf{x}^*\|^2 \\
979 &\leq \|\mathbf{x}_t - \mathbf{x}^*\|^2 + \eta_t^{2p} \|\nabla \mathbf{F}_t(\mathbf{x}_t)^T \lambda_t\|^2 - 2\eta_t^p \lambda_t^T \nabla \mathbf{F}_t(\mathbf{x}_t) (\mathbf{x}_t - \mathbf{x}^*).
\end{aligned}$$

980 Rearranging the above terms, it follows that

$$981 2\lambda_t^T \nabla \mathbf{F}_t(\mathbf{x}_t) (\mathbf{x}_t - \mathbf{x}^*) \leq \frac{\|\mathbf{x}_t - \mathbf{x}^*\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2}{\eta_t^p} + \eta_t^p \|\nabla \mathbf{F}_t(\mathbf{x}_t)^T \lambda_t\|^2.$$

982 Let $\eta_t^p = \frac{1}{Ht}$, by the H -strong convexity of the loss functions, we have

$$\begin{aligned}
983 2 \sum_{t=1}^T \lambda_t^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x}^*)) &\leq 2 \sum_{t=1}^T \lambda_t^T \nabla \mathbf{F}_t(\mathbf{x}_t) (\mathbf{x}_t - \mathbf{x}^*) - \sum_{t=1}^T H \|\mathbf{x}_t - \mathbf{x}^*\|^2 \\
984 &\leq \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2 \left(\frac{1}{\eta_t^p} - \frac{1}{\eta_{t-1}^p} - H \right) + G^2 \sum_{t=1}^T \eta_t^p \\
985 &\leq 0 + G^2 \sum_{t=1}^T \frac{1}{Ht} \\
986 &\leq \frac{G^2}{H} (1 + \log T). \tag{24}
\end{aligned}$$

987 Let $\eta_t^p = \frac{1}{F\sqrt{t}}$, $\alpha_t = F t^\tau$ and $\tau \in [0, 0.25)$, the analyze Part 2 in (23) is similar to that of Part 2 of

$$\begin{aligned}
988 2 \sum_{t=1}^T (\lambda_1 - \lambda_t)^T (\mathbf{F}_t(\mathbf{x}_t) - \mathbf{F}_t(\mathbf{x}^*)) &\leq \sqrt{2} \frac{1}{\eta_T^d} + 4pF^2 \sum_{t=1}^T \eta_t^{d+2\tau} \leq F\sqrt{2T} + \frac{4pF}{2\tau + 0.5} T^{2\tau+0.5}. \tag{25}
\end{aligned}$$

989 Finally, we summarize the above results. According to (24) and (25), the Theorem 3 holds.

990 \square

1018 D.4 DYNAMIC REGRET ANALYSIS FOR OJGD

1019 In this section, based on the dynamic multi-objective regret in (14), we analyze the dynamic multi-

1020 objective regret bound of the OJGD algorithm.

1021 **Theorem 7.** *Suppose Assumptions 1 and 2 hold and all functions are convex. If $\eta_t^p = \eta :=$*

$$1022 \frac{\sqrt{D(D+2L_*)}}{G\sqrt{T}}, \text{ where } L_* = \sum_{t=2}^T \|\mathbf{y}_{t-1}^* - \mathbf{y}_t^*\| \text{ and } \mathbf{y}_t^* \text{ is an optimal solution of problem (14) at} \\
1023 t\text{-th round, the OJGD attains the following dynamic regret} \\
1024 R_T^D \leq G\sqrt{D(D+2L_*)T}.$$

1026 *Proof.* Based on the update rule for \mathbf{x}_{t+1} , we have

$$\begin{aligned}
1027 & \|\mathbf{x}_{t+1} - \mathbf{y}_t^*\|^2 = \|\Pi_X(\mathbf{x}_t - \eta \nabla F_t(\mathbf{x}_t)^T \boldsymbol{\lambda}_t) - \mathbf{y}_t^*\|^2 \\
1028 & \leq \|\mathbf{x}_t - \eta \nabla F_t(\mathbf{x}_t)^T \boldsymbol{\lambda}_t - \mathbf{y}_t^*\|^2 \\
1029 & \leq \|\mathbf{x}_t - \mathbf{y}_t^*\|^2 + \eta^2 \|\nabla F_t(\mathbf{x}_t)^T \boldsymbol{\lambda}_t\|^2 - 2\eta \boldsymbol{\lambda}_t^T \nabla F_t(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{y}_t^*).
\end{aligned}$$

1032 Thus,

$$1033 \quad 2\boldsymbol{\lambda}_t^T \nabla F_t(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{y}_t^*) \leq \frac{\|\mathbf{x}_t - \mathbf{y}_t^*\|^2 - \|\mathbf{x}_{t+1} - \mathbf{y}_t^*\|^2}{\eta} + \eta \|\nabla F_t(\mathbf{x}_t)^T \boldsymbol{\lambda}_t\|^2.$$

1034 Summing the above inequality from $t = 1$ to T , and considering the convexity of the loss functions
1035 and Theorem 5, we have

$$\begin{aligned}
1036 & 2R_T^D = 2 \sum_{t=1}^T \min_{\bar{\boldsymbol{\lambda}}_t \in \Lambda} \max_{\mathbf{y}_t \in X} \bar{\boldsymbol{\lambda}}_t^T (F_t(\mathbf{x}_t) - F_t(\mathbf{y}_t)) \\
1037 & \leq 2 \sum_{t=1}^T \boldsymbol{\lambda}_t^T (F_t(\mathbf{x}_t) - F_t(\mathbf{y}_t^*)) \\
1038 & \leq 2 \sum_{t=1}^T \boldsymbol{\lambda}_t^T \nabla F_t(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{y}_t^*) \\
1039 & \leq \sum_{t=1}^T \frac{\|\mathbf{x}_t - \mathbf{y}_t^*\|^2 - \|\mathbf{x}_{t+1} - \mathbf{y}_t^*\|^2}{\eta} + \sum_{t=1}^T \eta \|\nabla F_t(\mathbf{x}_t)^T \boldsymbol{\lambda}_t\|^2 \\
1040 & \leq \frac{1}{\eta} \|\mathbf{x}_1 - \mathbf{y}_1^*\|^2 + \frac{1}{\eta} \sum_{t=2}^T (\|\mathbf{x}_t - \mathbf{y}_t^*\|^2 - \|\mathbf{x}_t - \mathbf{y}_{t-1}^*\|^2) - \frac{1}{\eta} \|\mathbf{x}_{T+1} - \mathbf{y}_T^*\|^2 + \eta G^2 T \\
1041 & \leq \frac{D^2}{\eta} + \frac{1}{\eta} \sum_{t=2}^T \|\mathbf{x}_t - \mathbf{y}_t^* + \mathbf{x}_t - \mathbf{y}_{t-1}^*\| \|\mathbf{y}_{t-1}^* - \mathbf{y}_t^*\| + \eta G^2 T \\
1042 & \leq \frac{D^2}{\eta} + \frac{1}{\eta} \sum_{t=2}^T 2D \|\mathbf{y}_{t-1}^* - \mathbf{y}_t^*\| + \eta G^2 T \\
1043 & = \frac{D(D + 2L_*)}{\eta} + \eta G^2 T.
\end{aligned}$$

1044 Set $\eta := \frac{\sqrt{D(D+2L_*)}}{G\sqrt{T}}$, then Theorem 7 holds. □

1045 E MORE EXPERIMENTAL RESULTS

1046 In this section, we provide more details of the experiment and the experimental results. All experi-
1047 ments are conducted on a single NVIDIA GeForce RTX 4090 GPU, PyTorch 1.8.1 platform.

1048 E.1 MORE RESULTS FOR CONVEX EXPERIMENTS

1049 The **mg** and **svmguide3** datasets used in our experiments are publicly available in Dua et al. (2017).
1050 The mg dataset contains 1385 instances with 6 features and the svmguide3 dataset contains 1243
1051 instances with 21 features. For the two convex problems, the experiments are conducted in the
1052 online setting, i.e., the samples arrive sequentially. For the mg dataset, the directional gain factor τ
1053 in $\alpha_t = Ft^T$ of OJGD is set as $\tau = 0.2$. For the svmguide3 dataset, the directional gain factor τ is
1054 set as $\tau = 0.1$. We supplement the results in Figure 4 and Figure 5.

1055 Figure 4 show that the OJGD algorithm has the best overall performance. Remarkably, the min-norm
1056 method obtains the lowest average loss in the regularization loss, however, it performs the worst on

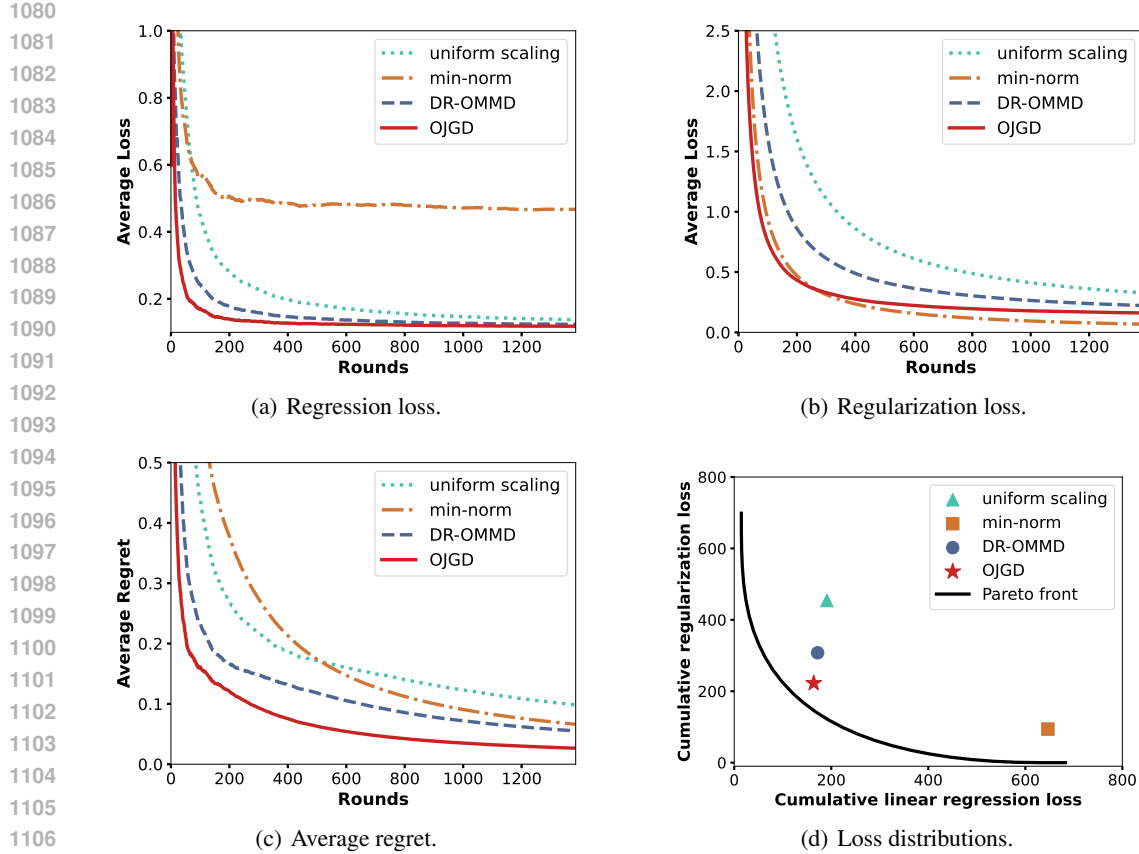


Figure 4: More empirical results of the algorithms on the mg dataset. The plots show the average cumulative losses in (a) and (b). The average multi-objective regrets are shown in (c). The distributions of cumulative losses are shown in (d).

the regression objective. This may be because the min-norm method updates the weights only based on minimizing the instantaneous joint gradient and lacks a regularization strategy for the weights, which may change drastically following the gradient, making it difficult to trade off multiple objectives in the optimization process stably. Figure 5 shows that OJGD consistently outperforms other algorithms on the svmguide3 dataset.

Furthermore, we explore the effect of different directional gain factors τ on the gradient phase. We set $\tau \in \{0, 0.05, 0.1, 0.15, 0.2, 0.24\}$ and implement the OJGD algorithm with the same settings as in the above experiments. The results on the svmguide3 dataset are shown in Figure 6. Specifically, for the t -th round, the phase θ_t is obtained from the joint gradient weights via $\theta_t = \arctan 2(\lambda_t^2, \lambda_t^1)$, and the radius r_t is the norm of the joint gradient, i.e., $r_t = \|\nabla F_t(\mathbf{x}_t)^T \lambda_t\|$. The colors of the points in Figure 6 are determined by the size of α_t , reflecting the optimization progress.

The results in Figure 6 show that even when $\tau = 0$, the OJGD algorithm still has some exploratory ability to adaptively adjust the weights during the optimization process. Compared with the performance of OJGD when $\tau = 0$, the OJGD algorithm with $\tau \in \{0.05, 0.1\}$ exhibits a similar exploratory ability in the early stage, but its convergence ability in the later stage is significantly enhanced, i.e., the weights stabilize around the initial weights in the later stage. It is worth noting that the OJGD algorithm’s exploratory ability in the medium stages is further enhanced when $\tau \in \{0.15, 0.2, 0.24\}$. This is because when the directional gain factor τ becomes large, the trade-off factor α_t also becomes large quickly. In the weight update direction $(F_t(\mathbf{x}_t) + \alpha_t \Delta(\lambda_t))$, the second term $\alpha_t \Delta(\lambda_t)$ quickly becomes the dominant factor compared to the first term $F_t(\mathbf{x}_t)$. In this case, the force in the algorithm to pull the weights back to the initial weights is too large, making the algorithm have exploration capabilities in the medium stage. As the stepsize η_t^d becomes gradually smaller, this pulling force also becomes smaller, allowing the algorithm to converge gradually. We

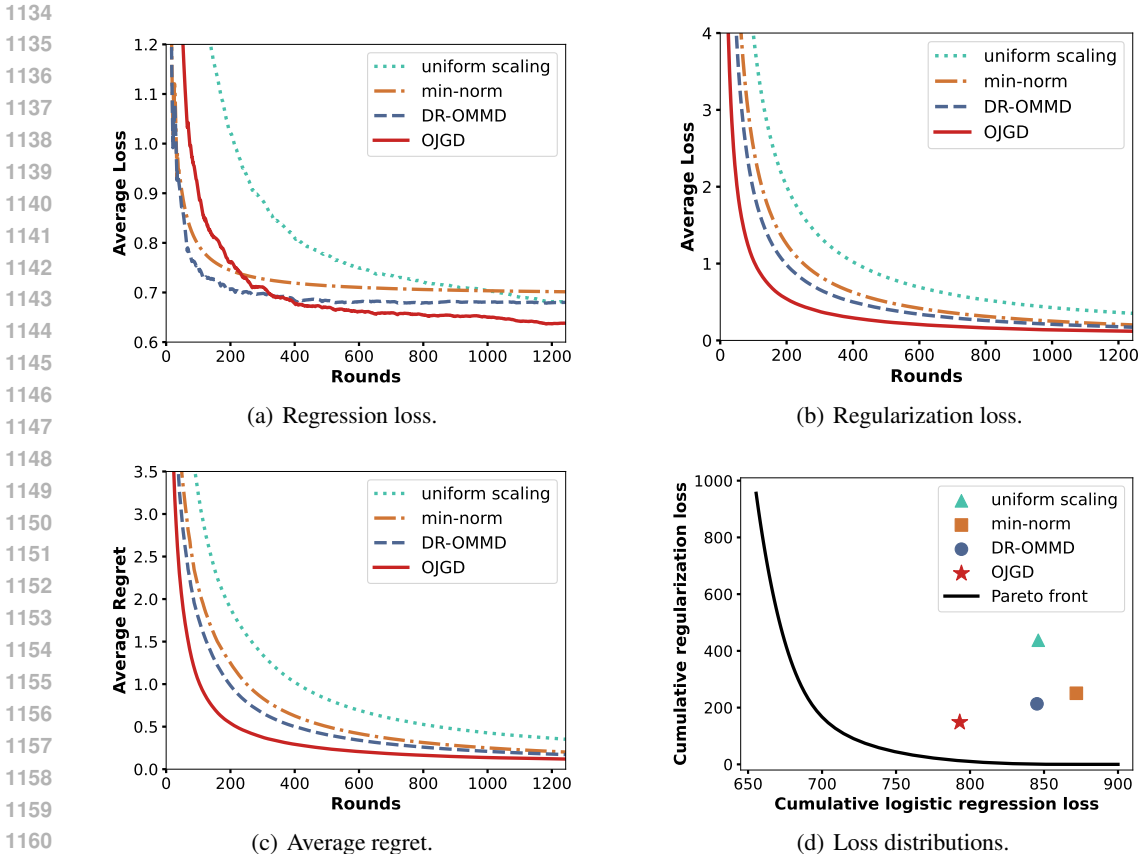


Figure 5: More empirical results of the algorithms on the svmguide3 dataset. The plots show the average cumulative losses in (a) and (b). The average multi-objective regrets are shown in (c). The distributions of cumulative losses are shown in (d).

further plot the distribution of the obtained cumulative losses in Figure 7, showing that the OJGD algorithm with $\tau = 0.2$ performs best.

E.2 MORE RESULTS FOR NON-CONVEX EXPERIMENTS

The MultiMNIST dataset is acquired by the code provided by Sener & Koltun (2018). It is a multi-task version of the MNIST dataset, where each sample consists of an overlap of two digital images from MNIST at the top-left and another one at the bottom-right. The tasks are simultaneously classifying the digit on the top-left (task-L) and the bottom-right (task-R) separately. Following the experimental setup of Jiang et al. (2023), this experiments are also conducted in the online setting and the learning rates of the algorithms are decided by a grid search over $\{0.0001, 0.001, 0.01, 0.1\}$. The stepsizes η_t^d and α_t of OJGD is set according to Algorithm 1. We supplement the results of the algorithms on MultiMNIST in Figure 8. The results show that OJGD consistently outperforms other algorithms.

The NYU-v2 dataset (Silberman et al., 2012) consists of images from indoor video sequences, including image segmentation, depth estimation, and surface normal estimation tasks. We follow the evaluation metrics in Liu et al. (2019) for the three tasks. Specifically, for semantic segmentation, we use Mean Intersection over Union (mIoU) and Pixel Accuracy (Pix Acc). For depth estimation, we use Absolute Error (Abs Err) and Relative Error (Rel Err). For surface normal estimation, we use five metrics, i.e., mean absolute of the error (Mean), median absolute of the error (Median), and the percentages of pixels with the angular error below 11.25° , 22.5° , and 30° (denoted as 11.25, 22.5, 30). In addition, following Maninis et al. (2019), we also compute the following metrics to reflect the overall performance of an algorithm a : the average per-task performance drop versus the

1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241

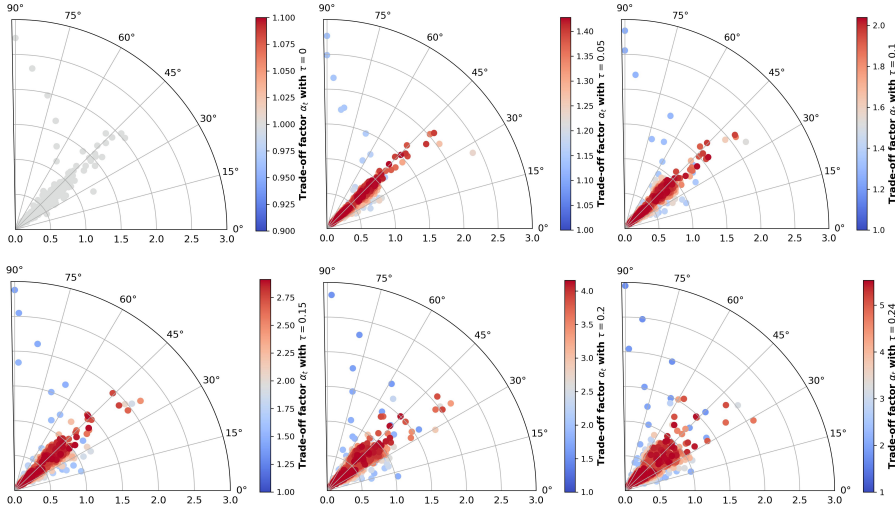


Figure 6: The phases of the joint gradients of the OJGD algorithm with different directional gain factors $\tau \in \{0, 0.05, 0.1, 0.15, 0.2, 0.24\}$ during the optimization process on the svmguide3 dataset.

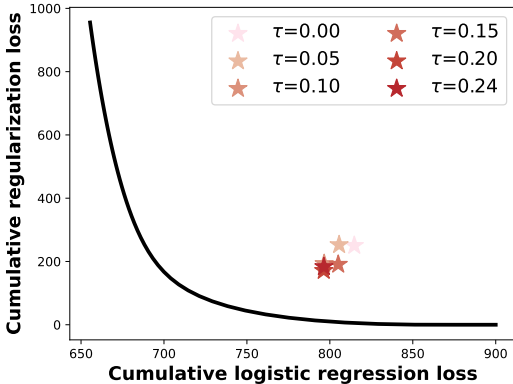


Figure 7: The distribution of the cumulative losses obtained by OJGD on svmguide3.

single-task baseline b , i.e., $\Delta M\% = \frac{1}{N_m} \sum_{n=1}^{N_m} (-1)^{l_n} (m_{a,n} - m_{b,n}) / m_{b,n} \times 100$, where $m_{a,n}$ and $m_{b,n}$ are respectively the n -th metric values for algorithms a and single-task b , $l_n = 1$ if higher n -th metric is better and 0 otherwise. We run all experiments using the MTL benchmark framework LibMTL (Lin & Zhang, 2023) with the MTAN (Liu et al., 2019) network architecture. The stepsizes η_t^d and α_t of OJGD is set according to Algorithm 1. For all algorithms, we trained MTAN with 200 epochs and a batch size of 2, and the default LibMTL configuration is used for all other experiment setups as shown in Table 2.

Table 2: Summary of hyper-parameter choices for NYU-v2 dataset.

Method	optimizer of \mathbf{x}_t	learning rate	weights stepsize	weight decay
uniform scaling	Adam	10^{-4}	-	10^{-5}
min-norm	Adam	10^{-4}	-	10^{-5}
DR-OMMD	Adam	10^{-4}	-	10^{-5}
OJGD	Adam	10^{-4}	10^{-4}	10^{-5}

The results of all algorithms on the NYU-v2 dataset are reported in Table 3 and show that OJGD obtains better or comparable results than other algorithms. Among all the algorithms, the OJGD

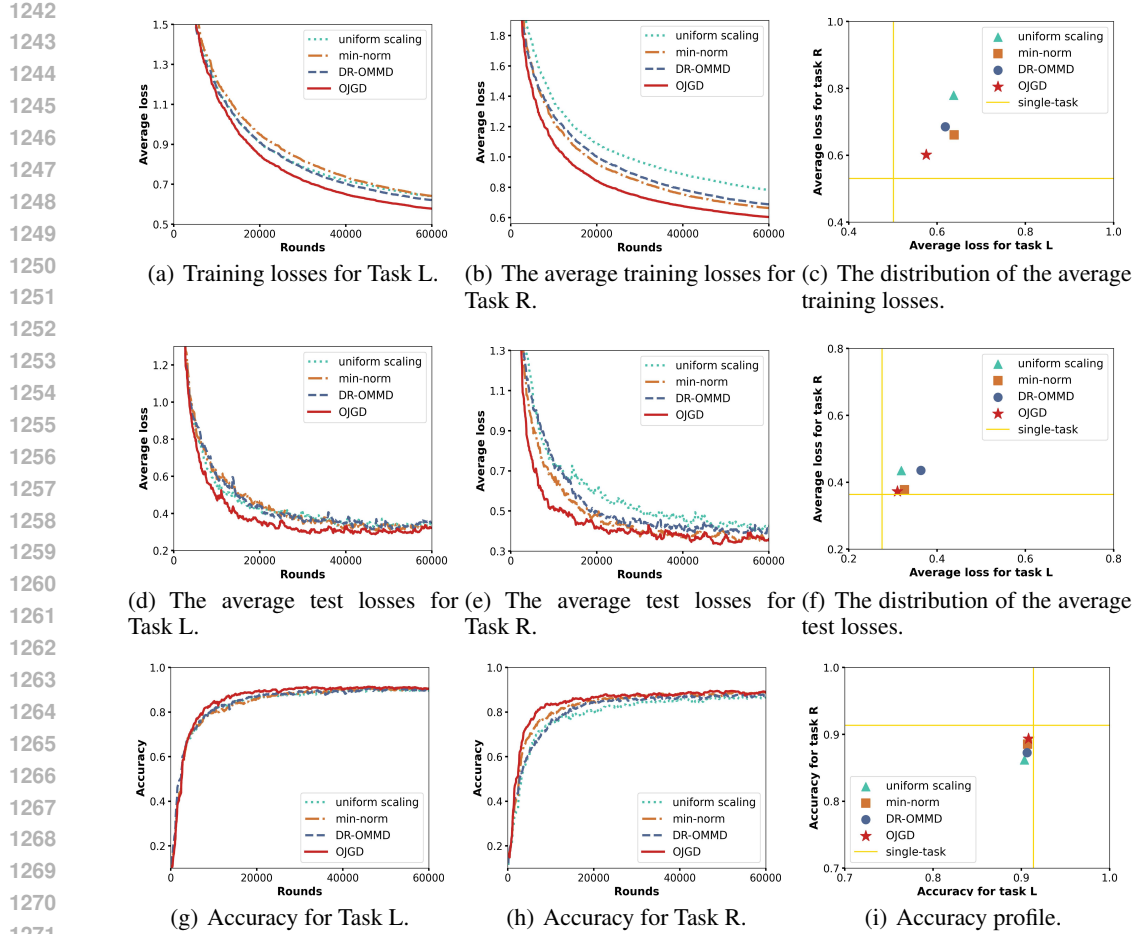


Figure 8: More empirical results of the algorithms for two tasks on MultiMNIST. The average training losses are plotted in Figure 8(a), 8(b) and 8(c). The average test losses are shown in Figure 8(d), 8(e) and 8(f). The accuracies of the models trained by the algorithms on the test dataset are shown in Figure 8(g), 8(h) and 8(i).

algorithm achieves the best results in four single performance metrics (mIoU, Pix Acc, Abs Err and Rel Err) and the overall performance metric ($\Delta M\%$).

Table 3: Segmentation, depth, and surface normal estimation results on NYU-v2 dataset. The mean and standard deviation are reported over 3 independent runs, where \uparrow indicates higher better and \downarrow indicates lower better. The best average result among all the algorithms is marked in bold.

Method	Segmentation		Depth		Surface Normal					$\Delta M\% \downarrow$
	mIoU \uparrow	Pix Acc \uparrow	Abs Err \downarrow	Rel Err \downarrow	Angle Distance \downarrow		Within $t^\circ \uparrow$			
					Mean	Median	11.25	22.5	30	
single-task	52.89 \pm 0.16	74.69 \pm 0.29	0.4185 \pm 0.0042	0.1706 \pm 0.0014	21.81 \pm 0.06	14.72 \pm 0.05	40.03 \pm 0.11	65.95 \pm 0.15	75.74 \pm 0.11	
uniform scaling	52.88 \pm 0.26	74.79 \pm 0.19	0.3848 \pm 0.0037	0.1565 \pm 0.0036	22.98 \pm 0.07	16.32 \pm 0.11	36.46 \pm 0.25	62.46 \pm 0.20	73.20 \pm 0.11	1.93 \pm 0.45
min-norm	45.87 \pm 0.15	69.56 \pm 0.13	0.4172 \pm 0.0034	0.1707 \pm 0.0007	21.86 \pm 0.05	15.07 \pm 0.15	39.18 \pm 0.32	65.24 \pm 0.33	75.39 \pm 0.17	2.92 \pm 0.42
DR-OMMD	51.57 \pm 0.33	73.85 \pm 0.33	0.3932 \pm 0.0056	0.1623 \pm 0.0040	22.47 \pm 0.13	15.76 \pm 0.22	37.55 \pm 0.47	63.79 \pm 0.51	74.62 \pm 0.27	1.53 \pm 0.17
OJGD	53.11 \pm 0.28	75.04 \pm 0.26	0.3812 \pm 0.0032	0.1552 \pm 0.0022	22.98 \pm 0.20	16.39 \pm 0.15	37.12 \pm 0.41	63.01 \pm 0.29	73.47 \pm 0.46	1.41 \pm 0.14

Moreover, to measure the performance efficiency per unit time of each algorithm, we report the average running time Δt of each algorithm for a single epoch and the Performance-Time (P-T) ratio ($\frac{1/\Delta M\%}{\Delta t} \times 100$) in Table 4. Combining the results in Tables 3 and 4, the OJGD algorithm can achieve the best overall performance among all the algorithms in the same time.

Table 4: The results for the performance of the algorithm and running time on NYU-v2 dataset. The mean and standard deviation are reported over 3 independent runs, where \uparrow indicates higher better and \downarrow indicates lower better. The best average result among all the algorithms is marked in bold.

Method	$\Delta M\% \downarrow$	$\Delta t(s) \downarrow$	P-T \uparrow
uniform scaling	1.93 ± 0.45	73.18 ± 0.61	0.7351 ± 0.1851
min-norm	2.92 ± 0.42	278.93 ± 3.47	0.1247 ± 0.0176
DR-OMMD	1.53 ± 0.17	281.45 ± 0.60	0.2339 ± 0.0263
OJGD	1.41 ± 0.14	78.26 ± 1.18	0.9130 ± 0.0884

F THE USE OF LARGE LANGUAGE MODELS (LLMs)

In this work, we used a large language model (LLM) as an assistive tool for language polishing. We clarify that while the LLM was involved in this way, all content ultimately attributed to the authors has been reviewed, verified, and edited by the authors. We take full responsibility for all parts of the manuscript. This usage of the LLM is disclosed here and in the submission form according to ICLR 2026 policies.