On the Convergence of Stochastic Multi-Objective Gradient Alteration and Beyond

Anonymous Author(s)
Affiliation
Address
email

Abstract

The conflicting gradients problem is the major bottleneck of the effective training of models that deal with multiple objectives. To resolve this problem, several gradient alteration techniques, such as PCGrad, MGDA, and CAGrad, are developed by directly altering the conflicting gradients to refined ones with fewer or even no conflicts. However, the design and analysis of these techniques are mainly conducted under the exact full-batch gradient setting, ignoring that they are primarily applied with stochastic mini-batch gradients. In this paper, we point out that the stochastic gradient alteration algorithms may fail to converge to Pareto optimal solutions. We summarize these seemingly different algorithms into a unified algorithmic framework, where the descent direction is given by the composition of the gradients w.r.t. the multiple objectives. Then we provide a simple two-objective convex optimization instance to illustrate the non-convergence issue in detail, which shows that the non-convergence results from the determination of the composite weights solely by the stochastic gradients. To fix this issue, we propose a novel composite weights determination scheme that exponentially averages the past calculated weights. Finally, we show that the resulting new variants of stochastic gradient alteration converge to Pareto optimality under the unified framework and also give rise to improved empirical performance for PCGrad, MGDA, and CAGrad.

1 Introduction

Multi-objective optimization (MOO) refers to optimizing towards more than 2 different objectives simultaneously [6, 5, 4]. MOO problems appear in many real-world scenarios, e.g., the online advertising model [27, 28, 21, 36] aims to maximize both the Click-Through Rate (CTR) and the Post-Click Conversion Rate (CVR); autonomous driving system [12, 26] needs to learn to solve multiple tasks such as self-localization and object identification at the same time. The main challenge lies in the inherent conflict between multiple objectives that impairs the training [17, 37, 22]. For example, in online advertising, merely optimizing either CVR or CTR will degrade the performance of the other [27, 28]. Various MOO algorithms [31, 37, 22, 35, 23, 14] were proposed to deal with the conflicting problem from by altering the gradients evaluated from multi-objective losses. Typically multiple gradient alteration (MOGA) methods such as MGDA [31], PCGrad [37], and CAGrad [22] adjust the multiple gradients to seek a local direction that descents all the objectives in the meantime.

While these algorithms have been successfully applied in a wide range of applications and achieved significant performance gain, the current design and analysis are derived with exact (e.g., full-batch) objective gradients [31, 37, 22, 35, 23, 14]. However, in most real-world scenarios, evaluating stochastic (e.g., mini-batch) gradients is much more practical from the computational cost aspect, especially for the training of deep neural networks [18, 29].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Since altering the stochastic gradients could further amplify the negative impact of the stochastic noise, such a gap between full and stochastic gradients may cause significant convergence issues. In particular, current MOGA algorithms only seek a local direction that only focuses on the multiple stochastic gradients in one step, while the common descent direction for the stochastic objectives possibly is not the right direction to descend the true ones in expectation.

In this paper, we analyze this problem and rigorously prove that such concern indeed exists. Surprisingly, we find that in some cases, *altering the stochastic gradients yields a direction that even degrades all the objectives in expectation.* Such an inverse direction that moves the algorithms in the wrong direction will resist the solution from convergence. This implies that current MOGA algorithms are suboptimal for MOO. In particular, we make the following main contributions.

To uniformity analyze different MOGA algorithms, we summarize them in a general framework as $\boldsymbol{d}_k = -\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)$, where $f^1, \ldots, f^m$ be the objectives of interest, $\{\lambda^i\}_{i=1}^m$ are coefficient of the different gradients at iteration k. Since the composite gradient $\boldsymbol{d}_k$ denotes the alternated direction from the receive gradients, the composite weights $\{\lambda^i\}_{i=1}^m$ are functions of the multiple gradients. We provide strict equivalent transformation for typical methods as MGDA, PCGrad, CAGrad into the sumarized form. This framework also contains recently proposed GradVac [35], IMTL-G [23], etc. In the stochastic setting, it represents as $\boldsymbol{d}_k = -\sum_{i=1}^m \lambda^i \boldsymbol{g}^i(\boldsymbol{x}_k)$, where $\{\boldsymbol{g}^i(\boldsymbol{x}_k)\}_{i=1}^m$ are stochastic gradients that are unbiased with the exact ones.

Under the above general framework, we elucidate how the MOGA algorithms can cause non-convergence by providing an example of a simple stochastic convex optimization problem where MGDA, PCGrad, and CAGrad provably do not converge to an optimal solution. The key reason is that the composite weights $\{\lambda^i\}_{i=1}^m$ are calculated only from the instantaneous gradients, thus sharing the same randomness. This strong correlation between the composite weights and stochastic gradients could lead to estimation bias, which forces the composite gradient in the wrong direction that even degrades all the objectives. Since MGDA, PCGrad, and CAGrad possess typical properties among MOGA algorithms, our analysis easily extends to others contained in our general framework as well. Furthermore, we point out that the previous analysis of convergence for the MGDA algorithm [25] is established on the unjustified assumption, where $\{\lambda^i\}_{i=1}^m$ are Lipschitz functions w.r.t. each gradient. We rigorously prove that this assumption is not true for MGDA, PCGrad, and CAGrad.

The above analysis suggests that in order to achieve guaranteed convergence, the optimization algorithm must reduce the correlation between the composite weights and stochastic gradients. To resolve this issue while preserving the practical advantages of MOGA, we fix MOGA algorithms with a principled and practical momentum mechanism that exponentially averages the composite weights instead of just using the values computed from the instantaneous gradients. Theoretically, momentum reduces the variance of the composite weight [3], making the correlation reduced as well. Thus, the inverse direction problem can be alleviated, and we prove that gradient alteration algorithms can converge to optimal with the same rate as single objective stochastic optimization by adding this mechanism. Our approach is simple to implement and agnostic to model and optimizer and hence can be easily plugged into various gradient alteration algorithms that follow the general framework.

Finally, we empirically verify the proposed algorithm in different multi-objective learning scenarios, including simulation and real-world deep multi-task learning tasks, and we observe consistent performance improvement without introducing computational or memory overhead.

2 Preliminaries

In this section, we briefly review the basic background knowledge of multiple-objective optimization, and introduce typical algorithms as well as their convergence analysis.

2.1 Basic Concepts of Multiple-Objective Optimization

Multiple-objective optimization (MOO) is concerned with solving the problems of optimizing multiple objective functions simultaneously [38, 31], termed as

$$\min_{\boldsymbol{x} \in \mathcal{K}} \boldsymbol{F}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x}))^\top, \tag{1}$$

where $m \geq 2$ denotes the number of objectives, and $\mathcal{K} \subset \mathbb{R}^d$ denotes the feasible set in a d dimension space. Each objective $f^i : \mathcal{K} \to \mathbb{R}$, $i \in \{1, \ldots, m\}$ is the i-th loss function. Different from single-

objective optimization where solutions $\boldsymbol{x}, \boldsymbol{x}'$ can be order by $f(\boldsymbol{x}) \leq f(\boldsymbol{x}')$ or $f(\boldsymbol{x}) \geq f(\boldsymbol{x}')$. In MOO, it could have two parameter vectors where one performs better for task i and the other performs better for task $j \neq i$. Therefore, Pareto optimality is defined to deal with such an incomparable case.

Definition 1 (Pareto optimality)**.** *For any two solutions $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{K}$, we say that $\boldsymbol{x}$ dominates $\boldsymbol{x}'$, denoted as $\boldsymbol{x} \prec \boldsymbol{x}'$ or $\boldsymbol{x}' \succ \boldsymbol{x}$, if $f^i(\boldsymbol{x}) \leq f^i(\boldsymbol{x}')$ for all i, and there exists one i such that $f^i(\boldsymbol{x}) < f^i(\boldsymbol{x}')$; otherwise, we say that $\boldsymbol{x}$ does not dominate $\boldsymbol{x}'$, denoted as $\boldsymbol{x} \not\prec \boldsymbol{x}'$ or $\boldsymbol{x}' \not\succ \boldsymbol{x}$. A solution $\boldsymbol{x}^* \in \mathcal{K}$ is called Pareto optimal if it is not dominated by any other solution in $\mathcal{K}$.*

Note that there are a set of Pareto optimal solutions, termed as **Pareto set**. The goal of MOO is to find a Pareto optimal solution, which is necessary to be Pareto critical [2].

Definition 2 (Pareto criticality)**.** *A solution $\boldsymbol{x}^* \in \mathcal{K}$ is called Pareto critical if there is no direction $\boldsymbol{d} \in \mathbb{R}^d$ such that $\nabla f^i(\boldsymbol{x}^*)^\top \boldsymbol{d} < 0, i = 1, \ldots m$ for all objectives.*

This definition indicates that if $\boldsymbol{x}$ is not Pareto critical, such a direction $\boldsymbol{d}$ is a descent direction for $\boldsymbol{F}$ at point $\boldsymbol{x}$. Optimizing through $\boldsymbol{d}$ in the local neighborhood of $\boldsymbol{x}$ is able to get a better solution that dominates $\boldsymbol{x}$ [6]. Since Pareto criticality reflects the local property compared with Pareto optimality, it is often used as the local minimal condition for MOO with non-convex objectives [7]. We then present the sufficient conditions for determining Pareto criticality and Pareto optimality, which appear as metrics to study the convergence for MOO algorithm [7, 34]. Denotes the probability simplex as $S_m = \{(\lambda^1, \ldots, \lambda^m) | \sum_{i=1}^m \lambda^i = 1, \lambda \in [0, 1]\}$, we have

Proposition 1. *For MOO problem 1, **(a)** If there exists $\boldsymbol{\lambda} \in S_m$ such that $\| \sum_{i=1}^m \lambda^i \nabla f^i(\boldsymbol{x}^*) \| = 0$, then $\boldsymbol{x}^* \in \mathcal{K}$ is Pareto critical. **(b)** For convex objectives $\boldsymbol{F}(\cdot)$. If there exists $\boldsymbol{\lambda} \in S_m$ such that $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathcal{K}} \boldsymbol{\lambda}^T \boldsymbol{F}(\boldsymbol{x})$, then $\boldsymbol{x}^*$ is Pareto optimal*[1].

(a) is derived from the an equivalent definition of Pareto critical points in [22]. (b) refers to Theorem 5.13 and Lemma 5.14 in [13], it implies that the minimizer of any linearization is Pareto optimal.

2.2 Multi-Objective Gradient Alteration

Similar to single-objective optimization, MOO can be solved by running iteratively with gradient-based algorithms. Suppose n denotes the number of iterations. At each $k = 1, \ldots, n$ round, the key challenge is the gradient conflicting problem, i.e., $\nabla f^i(\boldsymbol{x}_k)^\top \nabla f^j(\boldsymbol{x}_k) < 0, i \neq j, i, j \in [m]$, which leads to performance drop for an objective when optimizing another one [37], i.e. optimizing f^i along the negative gradient direction $-\nabla f^i(\boldsymbol{x}_k)$ will degrade f^j. Multi-Objective Gradient Alteration (MOGA) algorithms aim to search for a direction $\boldsymbol{d}_k$ that is not conflicting with each gradient. Using such a non-conflicting $\boldsymbol{d}_k$ to execute the gradient descent step to update the decision i.e., $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \eta \boldsymbol{d}_k$ where η is the step size, is shown to get better performance in practice [31, 37, 22]. Mathematically, we know that $f^i(\boldsymbol{x}_k + \eta \boldsymbol{d}_k) - f^i(\boldsymbol{x}_k) \approx \eta \boldsymbol{d}_k^\top \nabla f^i(\boldsymbol{x}_k), i = 1, \ldots, m$ by the first-order Taylor approximation assuming η is small [22]. Therefore, if $\boldsymbol{d}_k^\top \nabla f^i(\boldsymbol{x}_k) < 0$ for each objective i, then this direction is able to descent all objectives simultaneously. There are various ways to compute such kind of direction, and we next introduce typical MOGA methods as follows.

Multiple Gradient Descent Algorithm (MGDA) [31] directly optimizes towards the Pareto criticality by leveraging Definition 2. In each iteration k, MGDA aims to find a direction $\boldsymbol{d}_k$ to maximize the minimum decrease across the losses, i.e. $\max_{\boldsymbol{d} \in \mathbb{R}^d} \min_{i \in [m]} (f^i(\boldsymbol{x}_k) - f^i(\boldsymbol{x}_k + \eta \boldsymbol{d}_k)) \approx -\eta \min_{\boldsymbol{d} \in \mathbb{R}^d} \max_{i \in [m]} \nabla f^i(\boldsymbol{x}_k)^\top \boldsymbol{d}_k$. By regularizing the norm for $\boldsymbol{d}_k$ on the right side, it computes the direction by $\boldsymbol{d}_k = \arg\min_{\boldsymbol{d} \in \mathbb{R}^d} \{\max_{i \in [m]} \nabla f_i(\boldsymbol{x}_k)^\top \boldsymbol{d} + \frac{1}{2} \|\boldsymbol{d}\|_2^2\}$. This subproblem can be rewritten equivalently as the following differentiable quadratic optimization $\boldsymbol{d}_k, \mu = \arg\min_{d \in \mathbb{R}^d, \mu \in \mathbb{R}} \mu + \frac{1}{2} \|\boldsymbol{d}\|_2^2$, s.t. $\nabla f_i(\boldsymbol{x}_k)^\top \boldsymbol{d} \leq \mu$. If $\mu < 0$, then $\nabla f_i(\boldsymbol{x}_k)^\top \boldsymbol{d}_k < 0$, which means $\boldsymbol{x}_k$ is not Pareto critical from Definition 2, and $\boldsymbol{d}_k$ is the direction to descent all the objectives simultaneously [6, 7]. To simplify the optimization, such a primal problem has a dual objective as a *min-norm* oracle $\boldsymbol{\lambda}_k = \arg\min_{\lambda_k \in S_m} \| \sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k) \|$, The direction is then calculated by $\boldsymbol{d}_k = -\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)$.

Projecting Conflicting Gradients (PCGrad) [37] alters the conflicting gradients by projecting each onto the normal plane of the other, preventing the interfering components of the gradient from

[1]Precisely, $\boldsymbol{x}^*$ is weak Pareto optimal. If $\boldsymbol{x}^*$ is the unique minimizer, $\boldsymbol{x}^*$ is Pareto optimal. For convenience of understanding, we do not distinguish them in the main text, and provide detailed comparison in Appendix.

being applied to the network. Specifically, it initializes the projected gradients as $\boldsymbol{v}_i^{\text{PC}} = \nabla f^i(\boldsymbol{x}_k)$. If projected gradient $\boldsymbol{v}_i$ and raw gradient $\nabla f^j(\boldsymbol{x}_k)$ are conflicting, i.e., $\boldsymbol{v}_i^\top \nabla f^j(\boldsymbol{x}_k) < 0$, then it iteratively updates the projected gradients by $\boldsymbol{v}_i^{\text{PC}} = \boldsymbol{v}_i^{\text{PC}}(\boldsymbol{x}_k) - \frac{\boldsymbol{v}_i^{\text{PC}}(\boldsymbol{x}_k)^\top \nabla f^j(\boldsymbol{x}_k)}{\|\nabla f^j(\boldsymbol{x}_k)\|_2^2} \nabla f^j(\boldsymbol{x}_k)$, for $i, j = 1, \ldots, m$ until the projected gradients do not conflict with the received gradients. Since the conflicting components are overall eliminated, it makes the direction $\boldsymbol{d}_k = -\frac{1}{m}\sum_{i=1}^m \boldsymbol{v}_i^{\text{PC}}$ satisfies $\boldsymbol{d}_k^\top \nabla f^i(\boldsymbol{x}_k) < 0$ for each objective i, thus optimizing all objectives simultaneously.

Conflict-Averse Gradient (CAGrad) [22] generalizes averaging linearization and MGDA. Specifically, it constrains search region for the common direction as a circle around the average gradient $\boldsymbol{v}_0 = \sum_{i=1}^m \frac{1}{m}\nabla f^i(\boldsymbol{x}_k)$ with diagram $c\,\|\boldsymbol{v}_0\|$. Specifically, the direction $\boldsymbol{d}_k$ is yielded by $\boldsymbol{d}_k = \arg\min_{d\in\mathcal{R}^d} \max_{i\in[m]} \nabla f_i(\boldsymbol{x}_k)^\top \boldsymbol{d}, \quad s.t. \|\boldsymbol{d} - \boldsymbol{v}_0\| \leq c\|\boldsymbol{v}_0\|$. It can minimize the average loss function, while leveraging the worst local improvement of individual tasks to regularize the algorithm trajectory by using the common descent property of MGDA. Note that the gradient conflicting issue is not absolutely eliminated due to the new constrain, but the primal objective forces direction to approach the common descent direction as closed as possible. The dual objective is $\boldsymbol{\lambda} = \arg\min_{\boldsymbol{\lambda}\in S_m} \left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)^\top \boldsymbol{v}_0 + c\,\|\boldsymbol{v}_0\| \left\|\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right\|$. Then the direction is calculated by $\boldsymbol{d}_k = -\left(\boldsymbol{v}_0 + c\,\|\boldsymbol{v}_0\| \left(\sum_{i=1}^m \lambda_k^i \nabla f^i(\boldsymbol{x}_k)\right)\right)$.

Convergence analysis. MGDA has been shown to converge to an arbitrary Pareto critical/optimal point with the same rate as single-objective optimization [7]. A similar result has been proved with PCGrad [37]. CAGrad does not directly optimize towards Pareto optimality, but the averaging loss $\frac{1}{m}\sum_{i=1}^m f^i(\boldsymbol{x})$. It has been shown to converge to the minimizer of the averaging loss when $c \in [0, 1)$, and an arbitrary Pareto critical/optimal point when $c \geq 1$ [22]. By Proposition 1, we can infer that the minimizer of the averaging loss also lies in the Pareto set. Hence, we analyze them uniformly under the Pareto optimality in the following analysis.

3 General Framework for Multi-Objective Gradient Alteration

In this section, we summarize multi-objective gradient alteration (MOGA) algorithms into a general framework with the goal of uniformly analyzing the above algorithms.

3.1 The Framework with Full-bath Gradient

Without loss of generality, we summarize the direction search of MOGA in the following framework

$$\boldsymbol{d}_k = -\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k, \tag{2}$$

where $\boldsymbol{\lambda}_k = (\lambda_k^1, \ldots, \lambda_k^m)^\top$ is a function of the multiple gradients, since $\boldsymbol{\lambda}_k$ is obtained by a subproblem with respect to the received gradients, e.g. the dual form for MGDA and CAGrad. We here consider a more general assumption on the boundness of $\boldsymbol{\lambda}_k$ for including various MOGA methods than the probability simplex space in MGDA.

Assumption 1. *For the general framework 2, for any $k = 1, \ldots, n, i = 1, \ldots, m$, there exists a finite constant $B \in \mathcal{R}$, such that $0 \leq \lambda_k^i \leq B, \sum_{i=1}^m \lambda_k^i \geq 1$.*

We next introduce how this framework fits with typical methods. The dual form of MGDA directly follows the framework 2. For PCGrad, we take the two-objective case as an example. When the gradients are not conflicting, we have $\boldsymbol{\lambda}_k = (1/2, 1/2)$. When they are conflicting, we can rewrite $\boldsymbol{d}_k = -\frac{1}{2}(\boldsymbol{v}_1^{\text{PC}} + \boldsymbol{v}_2^{\text{PC}})$ by extending $\boldsymbol{v}_1^{\text{PC}}$ and $\boldsymbol{v}_2^{\text{PC}}$

$$\boldsymbol{d}_k = -\frac{1}{2}\left(\left(1 - \frac{\nabla f^1(\boldsymbol{x}_k)\cdot\nabla f^2(\boldsymbol{x}_k)}{\|\nabla f^1(\boldsymbol{x}_k)\|_2^2}\right)\nabla f^1(\boldsymbol{x}_k) + \left(1 - \frac{\nabla f^1(\boldsymbol{x}_k)\cdot\nabla f^2(\boldsymbol{x}_k)}{\|\nabla f^2(\boldsymbol{x}_k)\|_2^2}\right)\nabla f^2(\boldsymbol{x}_k)\right).$$

Then $\boldsymbol{\lambda}_k = -(\frac{1}{2} - \frac{\nabla f^1(\boldsymbol{x}_k)\cdot\nabla f^2(\boldsymbol{x}_k)}{2\|\nabla f^1(\boldsymbol{x}_k)\|_2^2}, \frac{1}{2} - \frac{\nabla f^1(\boldsymbol{x}_k)\cdot\nabla f^2(\boldsymbol{x}_k)}{2\|\nabla f^2(\boldsymbol{x}_k)\|_2^2})$, Since the more than three objectives cases should be presented as a protocol, we leave more details for the derivation in the Appendix. The direction $\boldsymbol{d}_k$ of CAGrad is essentially an aggregation of two linear combinations as shown in its dual objective, thus also following the general framework. Furthermore, all three algorithms satisfy Assumption 1, as discussed in Appendix.

Remark. The general framework also contains more recent approaches that are based on the fundamentals of the introduced methods, such as GradVac [35], IMTL-G [23]. Note that we focus on the

optimization side for multi-objective problems, thus we provide the derivation of existing fundamental optimizers. A vertical line of works that focus on neural architecture, such as RotoGrad [14], are not included in this work.

3.2 The Framework with Stochastic Gradients

MOO can be solved with gradient alteration algorithms with full gradients with theoretical guarantees [7, 37, 22]. However, training with mini-batch gradient is much more practical for deep learning applications [29], while MOGA with stochastic gradients is largely underexplored. Specifically, we only obtain noisy gradient feedback $\boldsymbol{G}_k = \left(\boldsymbol{g}^1(\boldsymbol{x}_k, \boldsymbol{\xi}_k^1), \ldots, \boldsymbol{g}^m(\boldsymbol{x}_k, \boldsymbol{\xi}_k^m)\right)$ that is corrupted by stochastic noise $\boldsymbol{\xi}_k^i, i = 1, \ldots, m$. Following 2, we have its stochastic version

$$\boldsymbol{d}_k = -\boldsymbol{G}_k \boldsymbol{\lambda}_k.$$

Note that here $\boldsymbol{\lambda}_k$ depends on stochastic gradients, not the exact ones. We then make the following standard assumption similar to single objective setting [16, 10], which says that we have access to unbiased stochastic estimates of the gradients for all objectives.

Assumption 2. *Let $\{\boldsymbol{x}_k\}_{k=1}^n$ be the iterate sequence for any algorithm. For all $k = 1, \ldots, n$, we have access to $\boldsymbol{g}^1(\boldsymbol{x}_k, \boldsymbol{\xi}_k^1), \ldots, \boldsymbol{g}^m(\boldsymbol{x}_k, \boldsymbol{\xi}_k^m)$ which are unbiased estimates of $f^1(\boldsymbol{x}_k), \ldots, f^m(\boldsymbol{x}_k)$, i.e. $\mathbb{E}_{\boldsymbol{\xi}_k}\left[\boldsymbol{g}^i(\boldsymbol{x}_k, \boldsymbol{\xi}_k^i)\right] = \nabla f^i(\boldsymbol{x}_k), i = 1, \ldots, m$. Further, each gradient variance is bounded by $\mathbb{E}_{\boldsymbol{\xi}_k}\left[\|\boldsymbol{g}^i(\boldsymbol{x}_k, \boldsymbol{\xi}_k^i) - \nabla f^i(\boldsymbol{x}_k)\|_2^2\right] \leq \sigma^2, i = 1, \ldots, m$.*

To analyze MOGA in the stochastic setting, we state the following definition to determine a sequence generated by an iterative algorithm to asymptotic converge to Pareto critical and optimal.

Definition 3 (Convergence condition)**.** ***(a)*** *A sequence $\{\boldsymbol{x}_k\}_{k=1}^\infty$ asymptotic converges to Pareto critical points if $\lim_{k\to\infty} \mathbb{E}[\min_{\boldsymbol{\lambda}_k^* \in S_m} \|\nabla \boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k^*\|] \to 0$.*
(b) *For convex multi-objective function $\boldsymbol{F}(\cdot)$. A sequence $\{\boldsymbol{x}_k\}_{k=1}^\infty$ asymptotic converges to Pareto optimal if $\lim_{k\to\infty} \mathbb{E}[\max_{\boldsymbol{x}_k^* \in \mathcal{K}} \min_{\boldsymbol{\lambda}_k^* \in S_m} (\boldsymbol{\lambda}_k^{*\top} \boldsymbol{F}(\boldsymbol{x}_k) - \boldsymbol{\lambda}_k^{*\top} \boldsymbol{F}(\boldsymbol{x}_k^*))] \to 0$.*

The definition (a) directly follows Proposition 1 (a) and the definition of Pareto critical. The definition (b) is because $\boldsymbol{x}_k^*$ is the minimum of $\boldsymbol{\lambda}_k^{*\top} \boldsymbol{F}(\boldsymbol{x})$, thus is Pareto optimal (Proposition 1 (b)). Asymptotically converging to $\boldsymbol{\lambda}_k^{*\top} \boldsymbol{F}(\boldsymbol{x}_k^*)$ implies that $\boldsymbol{x}_k$ approaches Pareto optimal. Note that the minimization operator of $\boldsymbol{\lambda}_k^*$ is from the existence condition in Proposition 1 (b). This definition can also be viewed as the stochastic version of metric function [33] and Pareto suboptimal gap [15].

4 The Non-convergence of Stochastic Multi-Objective Gradient Alteration

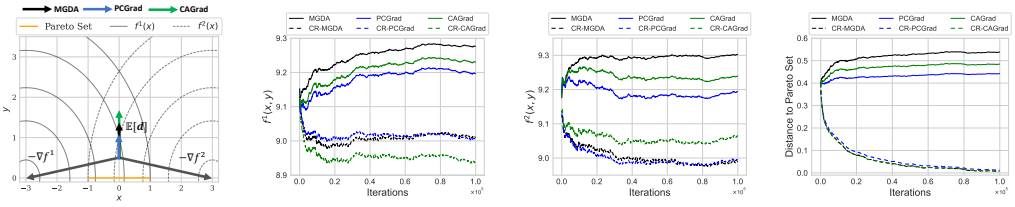
With the problem setup of stochastic MOGA in the previous section, we now discuss fundamental flaws among the current gradient alteration methods that are all designed and analyzed based on full-batch gradients. We show that such batch algorithms fail to converge to a Pareto optimal solution even in a simple two-objective two-dimensional convex stochastic setting. The main issue lies in the strong dependence of $\boldsymbol{\lambda}_k$ on the received stochastic gradients $\boldsymbol{G}_k$. Surprisingly, we find that combining such correlated $\boldsymbol{\lambda}_k$ with the stochastic gradient yields a biased composite gradient $\mathbb{E}[\boldsymbol{d}_k] = -\mathbb{E}[\boldsymbol{G}_k \boldsymbol{\lambda}_k]$ that even conflicts with all full-batch gradients in expectation. This inverse direction causes the algorithms even degrade all the objectives simultaneously.

An illustrative example. Next, we provide a simple problem instance in which MGDA, PCGrad, CAGrad can not converge to Pareto optimal. Consider the following two-dimensional stochastic optimization setting over the domain $\mathcal{K} = \{(x, y) \mid x \in [-1, 1], y \geq 0\}$

$$\tilde{f}^1(x, y) = \begin{cases} (x+1)^2 + (y-2)^2 & p = 1/2 \\ (x+5)^2 + (y+2)^2 & p = 1/2 \end{cases}, \tilde{f}^2(x, y) = \begin{cases} (x-1)^2 + (y-2)^2 & p = 1/2 \\ (x-5)^2 + (y+2)^2 & p = 1/2 \end{cases}. \quad (3)$$

The expected function is $f^1(x, y) = (x+3)^2 + y^2 + 8, f^2(x, y) = (x-3)^2 + y^2 + 8$. Thus, by ignoring the constant term, the optimization goal is to minimize the distance towards $(-3, 0)$ and $(3, 0)$ simultaneously. We can easily know that the Pareto set is the line segment $\{(a, 0) \mid a \in [-1, 1]\}$.

In the first step, by calculating the expected gradient of the point $(0, 0.4)$, we find that all the gradient alteration algorithms generate a vertical-up direction in expectation, as shown in Figure 1a, which



(a) Inverse Direction (b) Loss 1 (c) Loss 2 (d) Suboptimal Gap

Figure 1: (a) Gradient alteration algorithms designed based on full-batch gradient could yield an expected composite gradient that conflicts with the true gradients in the stochastic setting. (b) (c) This leads to the unexpected consequence of degrading all the objectives, making the objectives insufficiently optimized. (d) Hence, MGDA, PCGrad, CAGrad can not converge to Pareto optimal solution, and suffer a large suboptimal gap.

moves the algorithm in the wrong direction away from Pareto optimal setting. More seriously, the expected direction $\mathbb{E}[\boldsymbol{d}_k] = (0, -\epsilon), \epsilon > 0$ is conflicting with both true gradients, suggesting that it degrades both objectives.

Furthermore, we could prove that the inverse direction problem happens with Pareto optimal points, which resists the algorithms to approach Pareto optimal. Using the intuition above, we show that gradient alteration algorithms can not converge to Pareto optimal for this setting.

Theorem 1. *There is a stochastic convex optimization problem for which MGDA, PCGrad, CAGrad do not converge to the Pareto optimal solution.*

We relegate all proofs to the appendix. A few remarks are in order. In the proof, we provide stronger support that at least in the range $\{(x, y) \mid x \in [-1, 1], y \in [0, 0.05]\}$, the expected directions for the three typical algorithms all move the solution away from Pareto set. This means there exists a constant gap for gradient alteration algorithms to approach the Pareto optimal. Note that this gap is a sufficient condition to verify the non-convergence and can be much larger, as shown in Figure 1d.

Intuitively, the convergence gap is caused by the strong correlation between the composite weights $\boldsymbol{\lambda}_k$ and the stochastic gradients $\boldsymbol{G}_k$, i.e., $\boldsymbol{\lambda}_k$ is solely determined by $\boldsymbol{G}_k$, which could produce an inverse direction in the above example. When the composite weights is independent of the stochastic gradients, the expected composite gradient $\mathbb{E}[\boldsymbol{d}_k] = -\mathbb{E}[\boldsymbol{G}_k \boldsymbol{\lambda}_k] = \nabla F(\boldsymbol{x}_k)\boldsymbol{\lambda}_k$ will be a positive direction to approach the Pareto set. Hence, methods like linear scalarization are able to attain a point in the Pareto optimal set [9], but are deprived of the practical benefits of gradient alteration.

We would also like to emphasize that while the example of non-convergence is carefully constructed to more easily demonstrate the problems in the above three typical algorithms, such an issue also happens with more general Gaussian noise, as shown in Appendix. Hence, it is not unrealistic to suspect scenarios where such an issue can impair practical performance. It is also reasonable to imagine that this problem is especially aggravated in high dimensional settings, where the variance of the gradients is more significant.

Comparison with existing convergence results. As introduced in Section 2.2, the optimal solution for the multi-task objective $\frac{1}{m}\sum_{i=1}^{m} f^i(\boldsymbol{x})$ must be Pareto optimal. Therefore, the above non-convergence analysis generates well with MOGA algorithms that optimize the averaging loss, such as CAGrad. This result shows that the previous convergence analyses in the full-batch setting are all invalid in the stochastic setting.

To the best of our knowledge, there is only MGDA that has convergence analysis in the stochastic setting. Liu et al. [25] establishes the convergence of MGDA with stochastic gradients, but based on two unjustified assumptions. The first is that the gradient variance asymptotically converges to 0. The second is that the subproblem for $\boldsymbol{\lambda}_k$, denoted as function $\boldsymbol{\lambda}_k(\boldsymbol{g}^1, \ldots, \boldsymbol{g}^m)$, is Lipschitz continuous with right to each gradient. The first assumption needs to gradually increase the batch size to full batch; however, the reliance on giant batch sizes is impractical for deep learning training [3]. The second one can significantly simplify the convergence analysis, but is actually not true as shown in the following proposition.

Algorithm 1 Correlation-Reduced Stochastic Multi-Objective Gradient Alteration (CRMOGA)

1: **Input:** Number of iterations K, regularization parameter α_k, learning rate η_k.
2: **Initialize:** x_0, $\boldsymbol{\lambda}_0 \in \Delta_m$.
3: **for** $k = 1, \ldots, K$ **do**
4: Compute $\hat{\boldsymbol{\lambda}}_k$ by gradient alteration algorithms using the stochastic gradients.
5: Update the weights for the gradient composition $\boldsymbol{\lambda}_k = \alpha_k \boldsymbol{\lambda}_{k-1} + (1-\alpha_k)\hat{\boldsymbol{\lambda}}_k$.
6: Compute the composite gradient $\boldsymbol{d}_k = -\boldsymbol{G}_k(\boldsymbol{x}_k, \xi)\boldsymbol{\lambda}_k$.
7: Update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \eta_k \boldsymbol{d}_k$.
8: **end for**

Proposition 2. *Subproblem $\boldsymbol{\lambda}_k(\cdot)$ is not Lipschitz continuous with right to each gradient for MGDA, i.e., there exist no finite β such that $\|\boldsymbol{\lambda}_k(\boldsymbol{g}^1, \ldots, \boldsymbol{g}^i, \ldots, \boldsymbol{g}^m) - \boldsymbol{\lambda}_k(\boldsymbol{g}^1, \ldots, \boldsymbol{g'}^i, \ldots, \boldsymbol{g}^m)\| \leq \beta \|\boldsymbol{g}^i - \boldsymbol{g'}^i\|$ for $i = 1, \ldots, m$ and gradient $\boldsymbol{g}^1, \ldots, \boldsymbol{g}^i, \boldsymbol{g'}^i, \ldots, \boldsymbol{g}^m$.*

Similar results can be proved for PCGrad and CAGrad, and we relegate the proof to the appendix.

5 Correlation-Reduced Stochastic Multi-Objective Gradient Alteration

In this section, we develop a new principled exponential moving average mechanism on the composite weights $\boldsymbol{\lambda}_k$, and provide its theoretical benefit for fixing the convergence issue for MOGA algorithms. Our aim is to devise a new strategy with guaranteed convergence while preserving the practical benefits of MOGA.

Algorithmic insight. To understand the design of such a mechanism, recall that the main issue of the non-convergence for MOGA lies in the strong correlation between $\boldsymbol{\lambda}_k$ and the stochastic gradients $\boldsymbol{G}_k$, which could cause the composite gradient $\mathbb{E}[\boldsymbol{d}_k] = -\mathbb{E}[\boldsymbol{G}_k \boldsymbol{\lambda}_k]$ degrading all the objectives in expectation. It is natural to consider reducing such correlation, which can be bounded by

Lemma 1. *Under Assumption 1, 2. The correlation between $\boldsymbol{\lambda}_k$ and $\boldsymbol{G}_k$ can be bounded as $\|\mathbb{E}[\boldsymbol{G}_k \boldsymbol{\lambda}_k] - \mathbb{E}[\boldsymbol{G}_k]\mathbb{E}[\boldsymbol{\lambda}_k]\|_2^2 \leq mB^2\sigma^2 \mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{\lambda}_k)$, where $\mathbb{V}_{\boldsymbol{\xi}_k}(\boldsymbol{\lambda}_k) = \mathbb{E}_{\boldsymbol{\xi}_k}[\|\boldsymbol{\lambda}_k - \mathbb{E}_{\boldsymbol{\xi}_k}[\boldsymbol{\lambda}_k]\|_2^2]$.*

From Lemma 1, we know that the correlation can be reduced by reducing the variances of $\boldsymbol{\lambda}_k$. From this intuition, we can modify MOGA algorithms to reduce such correlation by regularizing the variation of $\boldsymbol{\lambda}_k$. Correlation-reduced MOGA (CR-MOGA) adopts a momentum mechanism with coefficient α_k on the update of composite weights as Line 5 in Algorithm 1, which is shown to be able to reduce variance as

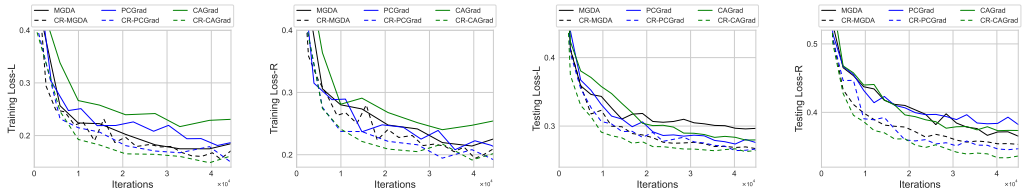
Lemma 2. *Under Assumption 1, $\boldsymbol{\lambda}_k$ variance for Algorithm 1 is bounded by $\mathbb{V}_{\boldsymbol{\xi}_k}[\boldsymbol{\lambda}_k] \leq B^2(1-\alpha_k)^2$.*

This result shows that the correlation can be controlled by choosing α_k closed enough to 1. With a reduced correlation, the wrong direction generated by MOGA could be corrected to a positive direction, and thus the algorithm would be able to converge. Note that the momentum on $\boldsymbol{\lambda}_k$ only requires an additional negligible computation cost.

On the other hand, the proposed exponential moving average mechanism also preserves the advantage of MOGA by carefully setting the momentum parameter α_k to be 0 and asymptotically convergent to 1. Intuitively, CRMOGA performs similarly to vanilla MOGA at the initial stage, and approximates like fixed linear scalarization in the late stage. Initially, the stochastic noise occupies a relatively small component of the stochastic gradient. Thus, methods like MGDA, PCGrad, CAGrad can search for good common descent directions. When the number of iterations increases and the solution approaches the Pareto optimal, the impact of the stochastic noise stands out, causing the inverse direction problem for MODA as in the previous section. At this time, α_k becomes larger, maintaining relatively stable $\boldsymbol{\lambda}_k$ eliminates the strong correlation and guarantees the convergence.

Convergence analysis. We next provide convergence analysis for CRMOGA 1 concluding all algorithms based on the general framework 2. Before that, we introduce basic assumptions.

Assumption 3. *$f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x})$ are all differentialable, H-Lipschitz and L-smoothness, suggesting that for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}, i \in [m]$, it holds $\|\nabla f^i(\boldsymbol{y})\| \leq H$ and $\|\nabla f^i(\boldsymbol{x}) - \nabla f^i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$.*



(a) Training Loss-L (b) Training Loss-R (c) Testing Loss-L (d) Testing Loss-R

Figure 2: Performance comparison of gradient alteration algorithms and their correlation-reduced counter part for MultiMNIST. The first two plots are for the training losses and the the last two are for the testing losses. The curves are smoothed by averaging 50 neighborhood points.

Lipschitz and smoothness are commonly used in complexity analysis for stochastic gradient algorithms [16, 10, 8, 3]. Based on Definition 3, we prove the following key results for CRMOGA.

Theorem 2 (Convex)**.** *Under Assumption 1, 2, 3. For the sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ generated by Algorithm 1, we assume objective functions $f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x})$ are all convex, and the distance from sequence to Pareto set is bounded, i.e., $\|\boldsymbol{x}_k - \boldsymbol{x}^*\| \leq D$ for $\boldsymbol{x}^*$ in Pareto set. Set $0 \leq \eta_n \leq \ldots \leq \eta_1, \alpha_k \in (0, 1]$ for $k = 1, \ldots, n$ in Algorithm 1, it achieves the following bound*

$$\frac{1}{n}\sum_{k=1}^{n}\sum_{i=1}^{m}\mathbb{E}\left[\max_{\boldsymbol{x}_k^* \in \mathcal{K}}\min_{\boldsymbol{\lambda}_k^* \in S_m}(\boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k) - \boldsymbol{\lambda}_k^{*\top}\boldsymbol{F}(\boldsymbol{x}_k^*))\right] \leq \frac{1}{2n\eta_n}D^2 + \frac{\sum_{k=1}^{n}\eta_k}{n}\frac{(2+D)}{2}m\sigma^2 + \frac{\sum_{k=1}^{n}B^2(1-\alpha_k)^2}{n\eta_k} + \frac{6HB\sum_{k=1}^{n}\eta_k(1-\alpha_k)}{n}\sigma(m+1).$$

Theorem 3 (Non-convex)**.** *Under Assumption 1, 2, 3. For the sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ generated by Algorithm 1, we assume objective functions $f^1(\boldsymbol{x}_k), \ldots, f^m(\boldsymbol{x}_k) \leq F$ are all bounded for $k = 1, \ldots, n$. Set $0 \leq \eta_n \leq \ldots \leq \eta_1 \leq 1$ and $\alpha_k \in (0, 1]$ in Algorithm 1, it achieves*

$$\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\min_{\boldsymbol{\lambda}_k^* \in S_m}\|\nabla\boldsymbol{F}(\boldsymbol{x}_k)\boldsymbol{\lambda}_k^*\|^2\right] \leq \frac{\sum_{k=1}^{n}4HB\sigma\eta_k(1-\alpha_k) + L\sigma^2\eta_k^2 + F(1-\alpha_k)|\lambda_{k-1}^i - \hat{\lambda}_k|}{\sum_{k=1}^{n}2\eta_k - L\eta_k^2}.$$

Remark. Note that our theoretical results imply that CRMOGA is convergent in averaging by choosing suitable η and α. Such an averaging analysis scheme is adopted by CAGrad [22], and is often used in stochastic optimization [32, 11, 3], e.g. Adagrad [20], Adam [18, 29].

The above results falls as an immediate corollary of the above Theorem.

Corollary 1. *Set $1 \geq \eta_k = O(1/\sqrt{k}), \alpha_k = \max\{1 - \eta_k/\eta_1, 1 - 1/(k\max_{i\in[m]}|\lambda_{k-1}^i - \hat{\lambda}_k^i|)\}$ in Theorem 2,3, Algorithm 1 converges with $O(n^{1/2})$, $O(n^{1/4})$ rates in the convex and non-convex case.*

This result indicates that Algorithm 1 has a convergence rate that matches the one in the single-objective stochastic setting. It demonstrates the efficacy of the proposed momentum mechanism.

Remark. One may argue that choosing $\alpha_k \approx 1$ lets CRMOGA approximate linear scalarization that is guaranteed to converge to Pareto optimal, and may impair the practical advantage for gradient alteration. Note that the coefficient $\alpha_k = 1 - 1/\sqrt{k}$ makes CRMOGA significantly different from fixed linearization. Such a parameter suggests the possible accumulated variation for $\boldsymbol{\lambda}_k$ is about $\sum_{k=1}^{n} 1/\sqrt{k} = O(\sqrt{n})$, which preserves an enough space for effective gradient alteration.

6 Experiments

In this section, we present empirical results on both synthetic with convex objectives and real-world datasets for multi-task learning with deep neural networks, representing convex and nonconvex settings, respectively. We apply CRMOGA to MGDA, PCGrad, CAGrad, and use CR- to distinguish whether it adopts our approach. We conduct experiments to answer the following questions:

Q1: Does our approach fix the non-convergence and improve the optimality for MOGA algorithms?

Method	Segmentation (Higher Better)				Depth (Lower Better)			
	(Training)		(Testing)		(Training)		(Testing)	
	mIoU	Pix Acc	mIoU	Pix Acc	Abs Err	Rel Err	Abs Err	Rel Err
MGDA [31]	74.97	93.79	68.84	91.54	0.0068	22.46	0.0309	33.50
CR-MGDA	**79.60**	**95.11**	**73.76**	**93.01**	**0.0063**	**18.51**	**0.0131**	**31.42**
PCGrad [37]	80.21	95.31	75.13	93.48	0.0088	27.73	0.0149	42.07
CR-PCGrad	**80.26**	**95.33**	**75.17**	**93.50**	**0.0087**	**26.77**	**0.0145**	**44.26**
CAGrad [22]	80.18	95.32	75.18	93.52	0.0081	25.34	0.0141	39.21
CR-CAGrad	**80.28**	**95.34**	**75.54**	**93.53**	**0.0080**	**24.94**	**0.0140**	**38.50**

Table 1: Multi-task learning results on CityScapes Challenge. Each experiment is repeated over 3 random seeds and the mean is reported.

Q2: Does our approach consistently improve the practical performance of MOGA in deep learning?

Q3: Is our approach flexible to be implemented and agnostic to models as well as optimizers?

Synthetic Experiments To demonstrate whether CRMOGA fixes the convergence issue for gradient alteration algorithm, we first consider the same simple convex setting as the example 3 in Section 4. We initialize $\boldsymbol{x}_0 = (0, 0.4)$ and use SGD as the optimizer. To enable fair comparison, the adaptive stepsize is set with $\eta_k = 0.006/\sqrt{k}$ for all the algorithms, and the momentum parameter for smoothing the composite weights is set with the suggested parameter setting in Corollary 1.

Figure 1 illustrates the multi-objective losses and distance to the Pareto set for this problem. We first note that both the two objective losses of MGDA, PCGrad, CAGrad are significantly larger than their counterparts applied with the proposed mechanism. Furthermore, the losses are even increasing at the initial stage as a result of the inverse direction issue demonstrated in Figure 1a. It moves $\boldsymbol{x}_k$ away from the Pareto set, which unfortunately causes an increasing suboptimal gap. In contrast, applying CRMOGA enhances the algorithms to converge to Pareto optimal. This positively answers **Q1**.

Table 2: Accuracy on MultiMNIST.

Method	task-L Acc	task-R Acc
MGDA	96.46%	95.29%
CR-MGDA	**96.92**% ↑	**95.44**% ↑
PCGrad	96.63%	95.37%
CR-PCGrad	**96.90**% ↑	**95.75**% ↑
CAGrad	96.47%	95.20%
CR-CAGrad	**96.85**% ↑	**95.70**% ↑

Multi-Task Classification For multi-task classification scenarios, we follow the experiment setup from Sener et al. [31], and consider the MultiMNIST dataset [30] with 60K examples. MultiMNIST contains two tasks: classifying the digit on the top-left (task-L) and classifying the digit on the bottom-right (task-R). We use the LeNet architecture [19], and use SGD as the optimizer. The stepsize and parameters are tuned from vanilla MGDA, PCGrad, CAGrad, and the correlation-reduced counterparts follow the same parameter setting to enable a fair comparison. The momentum parameter is set with suggested value in Corollary 1.

We report the train and test losses in Figure 2. We can see that algorithms with $\boldsymbol{\lambda}$ momentum perform better than original ones with respect to both train and test losses. We also observed that $\boldsymbol{\lambda}$ momentum consistently improves the accuracy for all the objectives in Table 2. We hence positively answer **Q2**.

Semantic Segmentation For semantic segmentation scenarios, we follow the experiment setup from Yu et al. [37] and Liu et al. [22], and consider the CityScapes. CityScapes similarly contains two tasks: 7-class semantic segmentation and depth estimation. We combine algorithms with MTAN [24] based on SegNet architecture [1]. We use adam [18] as the optimizer. All parameters are tuned from vanilla MOGA algorithms, and the correlation-reduced counterparts follow the same parameter setting to enable a fair comparison. The momentum parameter is set with the suggested value.

We observe consistent improvement when combining our approach with the three gradient alteration methods. In particular, the proposed mechanism enhances the optimality of solutions with respect to both train and test losses. Combining the above results, we demonstrate that our methods work for different models and optimizers. We hence give a positive answer to **Q3**.

7 Conclusions

In this work, we study multi-objective gradient alteration in the stochastic setting and identify a fundamental flaw in these algorithms. We prove their non-convergence issue of them via an illustrative example. We give fixes to this problem by slightly changing the algorithms. We prove the convergence of the resulting algorithms and show that they yield consistent empirical performance improvement.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] Ana Luísa Custódio, JF Aguilar Madeira, A Ismael F Vaz, and Luís Nunes Vicente. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21(3):1109–1140, 2011.

[3] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

[4] Kalyanmoy Deb. Multi-objective optimization. In *Search methodologies*, pages 403–449. Springer, 2014.

[5] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

[6] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[7] Jörg Fliege, A Ismael F Vaz, and Luís Nunes Vicente. Complexity of gradient descent for multiobjective optimization. *Optimization Methods and Software*, 34(5):949–959, 2019.

[8] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. Sgd: General analysis and improved rates. In *International Conference on Machine Learning*, pages 5200–5209, 2019.

[9] Walter J Gutjahr and Alois Pichler. Stochastic multi-objective optimization: a survey on non-scalarizing methods. *Annals of Operations Research*, 236(2):475–499, 2016.

[10] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234, 2016.

[11] Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

[12] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2702–2719, 2019.

[13] Johannes Jahn. *Vector optimization*. Springer, 2009.

[14] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. In *International Conference on Learning Representations*, 2022.

[15] Jiyan Jiang, Wenpeng Zhang, Shiji Zhou, Lihong Gu, Xiaodong Zeng, and Wenwu Zhu. Multi-objective online learning. 2021.

[16] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

[17] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

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

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

[20] Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 983–992, 2019.

[21] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto multi-task learning. In *Thirty-third Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

[22] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[23] Liyang Liu, Yi Li, Zhanghui Kuang, J Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021.

[24] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019.

[25] Suyun Liu and Luis Nunes Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *Annals of Operations Research*, pages 1–30, 2021.

[26] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6389–6398, 2019.

[27] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1930–1939, 2018.

[28] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1137–1140, 2018.

[29] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[30] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*, 2017.

[31] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 525–536, 2018.

[32] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning*, pages 71–79. PMLR, 2013.

[33] Hiroki Tanabe, Ellen H Fukuda, and Nobuo Yamashita. Proximal gradient methods for multi-objective optimization and their applications. *Computational Optimization and Applications*, 72(2):339–361, 2019.

[34] Hiroki Tanabe, Ellen H Fukuda, and Nobuo Yamashita. Convergence rates analysis of multiobjective proximal gradient methods. *arXiv preprint arXiv:2010.08217*, 2020.

[35] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2020.

[36] Ruobing Xie, Yanlei Liu, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. Personalized approximate pareto-efficient recommendation. In *Proceedings of the Web Conference 2021*, pages 3839–3849, 2021.

[37] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[38] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our work is concerning a general problem in multi-objective optimization with stochastic gradients.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 2, 3, 5.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In the supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 6.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite the source of datasets.
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] In the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We use publicly available datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We use publicly available datasets.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]