Smooth Tchebycheff Scalarization for Multi-Objective Optimization

Xi Lin [1] **Xiaoyuan Zhang** [1] **Zhiyuan Yang** [1] **Fei Liu** [1] **Zhenkun Wang** [2] **Qingfu Zhang** [1]

Abstract

Multi-objective optimization problems can be found in many real-world applications, where the objectives often conflict each other and cannot be optimized by a single solution. In the past few decades, numerous methods have been proposed to find Pareto solutions that represent optimal trade-offs among the objectives for a given problem. However, these existing methods could have high computational complexity or may not have good theoretical properties for solving a general differentiable multi-objective optimization problem. In this work, by leveraging the smooth optimization technique, we propose a lightweight and efficient smooth Tchebycheff scalarization approach for gradient-based multi-objective optimization. It has good theoretical properties for finding all Pareto solutions with valid trade-off preferences, while enjoying significantly lower computational complexity compared to other methods. Experimental results on various real-world application problems fully demonstrate the effectiveness of our proposed method.

1. Introduction

Many real-world applications involve multi-objective optimization problems, such as learning an accurate but also fair model (Martinez et al., 2020), building a powerful yet energy-efficient agent (Xu et al., 2020), and finding a drug design that satisfies various criteria (Xie et al., 2021). However, these objectives often conflict each other, making it impossible to optimize them simultaneously with a single solution. For each nontrivial multi-objective optimization problem, there exists a Pareto set that probably contains an infinite number of Pareto solutions that represent different optimal trade-offs among the objectives (Miettinen, 1999; Ehrgott, 2005). Numerous algorithms have been proposed to find a single solution or a finite set of solutions to approximate the Pareto set. This work focuses on the gradient-based methods for differentiable multi-objective optimization.

There are two main types of gradient-based multi-objective optimization algorithms: the scalarization approach (Miettinen, 1999; Ehrgott, 2005) and the adaptive gradient method (Fliege et al., 2019). However, both algorithms have their own disadvantages. The simple linear scalarization is the most straightforward method, but will completely miss all the solutions on the non-convex part of the optimal Pareto front (Das & Dennis, 1997; Ehrgott, 2005). The classical Tchebycheff scalarization can find all Pareto solutions that satisfy the decision-maker's exact trade-off preference (Bowman, 1976; Steuer & Choo, 1983), but it suffers from a slow convergence rate due to its nonsmooth (i.e., non-differentiable) nature. To overcome the limitations of scalarization approaches, many adaptive gradient methods have been proposed in the past few decades (Fliege & Svaiter, 2000; Schäffler et al., 2002; Désidéri, 2012). These methods aim to find a valid gradient direction that improves the performance of all objectives at each iteration. While these methods have theoretical guarantees for finding a Pareto stationary solution, their high pre-iteration computational complexity makes them less suitable for handling large-scale problems in real-world applications.

Instead of proposing another adaptive gradient method, this work revisits the straightforward scalarization approach. In particular, we leverage the powerful smooth approximation technique (Nesterov, 2005; Beck & Teboulle, 2012) to develop a lightweight yet efficient scalarization approach with promising theoretical properties. The main contributions of this work are summarized as follows. [1]

- We propose a smooth Tchebycheff (STCH) scalarization approach for gradient-based multi-objective optimization, which can serve as a fast alternative to the classic Tchebycheff scalarization.
- We provide detailed theoretical analyses to demonstrate that STCH scalarization has promising theoretical properties while enjoying low computational complexity.
- We further generalize STCH scalarization to support efficient Pareto set learning.

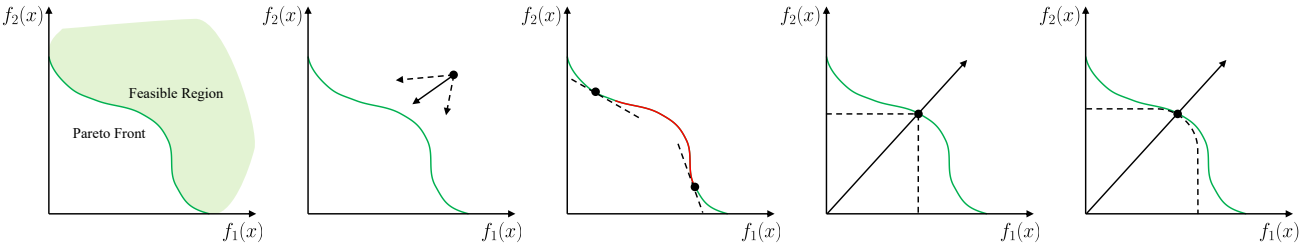
[1]City University of Hong Kong (email: xi.lin@my.cityu.edu.hk) [2]Southern University of Science and Technology. Correspondence to: Qingfu Zhang <qingfu.zhang@cityu.edu.hk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]Our source code is available at: github.com/Xi-L/STCH.

Table 1. Comparison of different methods for gradient-based multi-objective optimization.

	Required # Iterations	Pre-Iteration Complexity	Non-convex Pareto Front
Adaptive Gradient Method	Small	High	Yes
Linear Scalarization	Small	Low	No
Tchebycheff (TCH) Scalarization	Large	Low	Yes
STCH Scalarization (This Work)	Small	Low	Yes



(a) Problem (b) Adaptive Gradient (c) Linear Scalarization (d) TCH Scalarization (e) STCH (This Work)

Figure 1. **Different multi-objective optimization methods. (a) The Pareto Front** is the achievable boundary of the feasible region that represents different (maybe infinite) optimal trade-offs among the objectives. **(b) Adaptive Gradient Algorithm** aims to find a valid gradient direction to improve the performance of all objectives, which involves solving a quadratic programming problem at each iteration. **(c) Linear scalarization** cannot find any Pareto solution on the non-convex part of the Pareto front, of which those solutions do not have supporting hyperplanes. **(d) Tchebycheff (TCH) Scalarization** is capable of finding all Pareto solutions, but requires a large number of iterations. **(e) Smooth Tchebycheff (STCH) Scalarization** proposed in this work can find all Pareto solutions under mild conditions, while enjoying a much faster convergence speed.

- We conduct various experiments on diverse multi-objective optimization problems. The results confirm the effectiveness of our proposed STCH scalarization.

2. Preliminaries and Related Work for Multi-Objective Optimization

In this work, we consider the following continuous multi-objective optimization problem (MOP) with m differentiable objective functions $\boldsymbol{f} = \{f_1, \cdots, f_m\} : \mathcal{X} \to \mathbb{R}^m$ in the decision space $\mathcal{X} \subseteq \mathbb{R}^n$:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})). \tag{1}$$

For a non-trivial problem (1), the objective functions $\{f_1, \cdots, f_m\}$ will conflict each other and cannot be simultaneously optimized by a single best solution. For multi-objective optimization, we have the following definition on dominance relation among solutions (Miettinen, 1999):

Definition 2.1 (Dominance and Strict Dominance)**.** *Let $\boldsymbol{x}^{(a)}, \boldsymbol{x}^{(b)} \in \mathcal{X}$ be two solutions for problem (1), $\boldsymbol{x}^{(a)}$ is said to dominate $\boldsymbol{x}^{(b)}$, denoted as $\boldsymbol{f}(\boldsymbol{x}^{(a)}) \prec \boldsymbol{f}(\boldsymbol{x}^{(b)})$, if and only if $f_i(\boldsymbol{x}^{(a)}) \leq f_i(\boldsymbol{x}^{(b)}) \, \forall i \in \{1, ..., m\}$ and $f_j(\boldsymbol{x}^{(a)}) < f_j(\boldsymbol{x}^{(b)}) \, \exists j \in \{1, ..., m\}$. In addition, $\boldsymbol{x}^{(a)}$ is said to strictly dominate $\boldsymbol{x}^{(b)}$ (i.e., $\boldsymbol{f}(\boldsymbol{x}^{(a)}) \prec_{strict} \boldsymbol{f}(\boldsymbol{x}^{(b)})$), if and only if $f_i(\boldsymbol{x}^{(a)}) < f_i(\boldsymbol{x}^{(b)}) \, \forall i \in \{1, ..., m\}$.*

The (strict) dominance relation only establishes a partial order among different solutions since any two solutions may be non-dominated with each other (e.g., $\boldsymbol{f}(\boldsymbol{x}^{(a)}) \not\prec \boldsymbol{f}(\boldsymbol{x}^{(b)})$ and $\boldsymbol{f}(\boldsymbol{x}^{(b)}) \not\prec \boldsymbol{f}(\boldsymbol{x}^{(a)})$). In this case, the solution $\boldsymbol{x}^{(a)}$ and $\boldsymbol{x}^{(b)}$ are said to be incomparable. Therefore, we have the following definition of Pareto optimality to describe the optimal solutions for multi-objective optimization:

Definition 2.2 ((Weakly) Pareto Optimality)**.** *A solution $\boldsymbol{x}^* \in \mathcal{X}$ is Pareto optimal if there is no $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{f}(\boldsymbol{x}) \prec \boldsymbol{f}(\boldsymbol{x}^*)$. A solution $\boldsymbol{x}' \in \mathcal{X}$ is weakly Pareto optimal if there is no $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{f}(\boldsymbol{x}) \prec_{strict} \boldsymbol{f}(\boldsymbol{x}')$.*

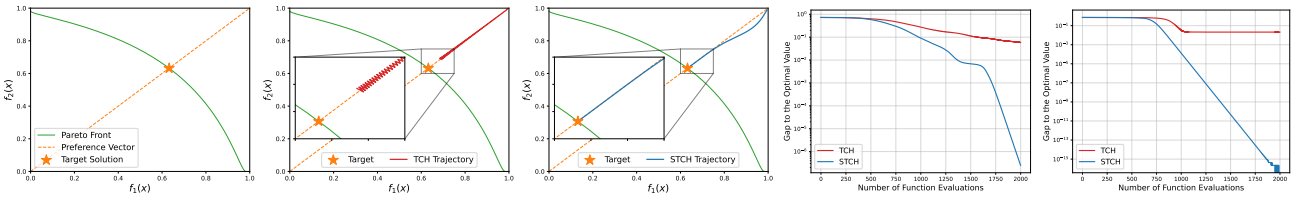
Similarly, a solution $\boldsymbol{x}^*$ is called local (weakly) Pareto optimal if it is (weakly) Pareto optimal in $\mathcal{X} \cap B(\boldsymbol{x}^*, \delta)$, of which $B(\boldsymbol{x}^*, \delta) = \{\boldsymbol{x} \in \mathbb{R}^n | ||x - x^*|| < \delta\}$ is an open ball around $\boldsymbol{x}^*$ with radius $\delta > 0$. In general, the Pareto optimal solution is not unique, and the set of all Pareto optimal solutions is called the *Pareto set*:

$$\boldsymbol{X}^* = \{\boldsymbol{x} \in \mathcal{X} | \boldsymbol{f}(\hat{\boldsymbol{x}}) \not\prec \boldsymbol{f}(\boldsymbol{x}) \, \forall \hat{\boldsymbol{x}} \in \mathcal{X}\}. \tag{2}$$

Its image in the objective space is called the *Pareto front*:

$$\boldsymbol{f}(\boldsymbol{X}^*) = \{\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^m | \boldsymbol{x} \in \boldsymbol{X}^*\}. \tag{3}$$

The weakly Pareto set $\boldsymbol{X}_w^*$ and front $\boldsymbol{f}(\boldsymbol{X}_w^*)$ can be defined accordingly. It is clear that $\boldsymbol{X}^* \subseteq \boldsymbol{X}_w^*$. Under mild conditions, the Pareto set $\boldsymbol{X}^*$ and the Pareto front $\boldsymbol{f}(\boldsymbol{X}^*)$ are both $(m-1)$-dimensional manifolds in the decision space $\mathbb{R}^n$ and objective space $\mathbb{R}^m$, respectively (Hillermeier, 2001). Many multi-objective optimization methods have been proposed to find a single or a finite set of solutions in the Pareto set $\boldsymbol{X}^*$ for the problem (1). The scalarization approach and the adaptive gradient algorithm are two popular methods when all objective functions are differentiable.



(a) Problem & Target (b) TCH (c) STCH (Ours) (d) Mean Gap (e) Median Gap

Figure 2. **The advantage of our proposed smooth Tchebycheff scalarization. (a)** Problem & Target: We want to find a Pareto solution with an exact trade-off $\boldsymbol{\lambda} = (0.5, 0.5)$ on the Pareto front. **(b)** Classical Tchebycheff (TCH) scalarization suffers from a slow convergence speed with an oscillation trajectory. **(c)** Our proposed smooth Tchebycheff (STCH) scalarization quickly converges to the exact target solution with a smooth trajectory. **(d) & (e)** The mean/median gaps v.s. number of function evaluations of different methods to the target objective value with 100 trials.

Scalarization Approaches Scalarization is a classical and popular method for tackling multi-objective optimization (Miettinen, 1999; Zhang & Li, 2007). The most straightforward approach is the simple linear scalarization (Geoffrion, 1967):

$$\min_{\boldsymbol{x}\in\mathcal{X}} g^{(\mathrm{LS})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}), \tag{4}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)$ is a preference vector on the simplex $\boldsymbol{\Delta}^{m-1} = \{\boldsymbol{\lambda} | \sum_{i=1}^{m} \lambda_i = 1 \text{ and } \lambda_i \geq 0, \forall i\}$ over the m objectives. Once a preference $\boldsymbol{\lambda}$ is given, we can obtain a solution by solving the single-objective scalarization problem (4). However, from the perspective of multi-objective optimization, this method can only find solutions on the convex hull of the Pareto front (Boyd & Vandenberghe, 2004; Ehrgott, 2005), and all solutions on the non-convex parts of the Pareto front will be missing (Das & Dennis, 1997). In fact, there is no guarantee that a general scalarization method can find all Pareto solutions, where the Tchebycheff scalarization method is an exception (Miettinen, 1999).

Tchebycheff Scalarization In this work, we focus on the Tchebycheff (TCH) scalarization with promising theoretical properties (Bowman, 1976; Steuer & Choo, 1983):

$$\min_{\boldsymbol{x}\in\mathcal{X}} g^{(\mathrm{TCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \min_{\boldsymbol{x}\in\mathcal{X}} \max_{1\leq i\leq m} \{\lambda_i (f_i(\boldsymbol{x}) - z_i^*)\}, \tag{5}$$

where $\boldsymbol{\lambda} \in \boldsymbol{\Delta}^{m-1}$ is the preference vector and $\boldsymbol{z}^* \in \mathbb{R}^m$ is the ideal objective values (e.g., $\boldsymbol{z}_i^* = \min f_i(\boldsymbol{x}) - \epsilon$ with a small $\epsilon > 0$). Under mild conditions, its optimal solution $\boldsymbol{x}_{\boldsymbol{\lambda}}^*$ has the desirable pattern (Ehrgott, 2005):

$$\lambda_i (f_i(\boldsymbol{x}_{\boldsymbol{\lambda}}^*) - z_i^*) = \lambda_j (f_j(\boldsymbol{x}_{\boldsymbol{\lambda}}^*) - z_j^*), \quad \forall 1 \leq i, j \leq m \tag{6}$$

which naturally satisfies the exact preferred trad-off requirement (Mahapatra & Rajan, 2020). There is also a promising necessary and sufficient condition for TCH scalarization to find *all* (weakly) Pareto solutions (Choo & Atkins, 1983):

Theorem 2.3. *A feasible solution $\boldsymbol{x} \in \mathcal{X}$ is weakly Pareto optimal for the original problem (1) if and only if there exists a valid preference vector $\boldsymbol{\lambda}$ such that $\boldsymbol{x}$ is an optimal solution of the Tchebycheff scalarization problem (5).*

In addition, if the optimal solution is unique for a given $\boldsymbol{\lambda}$, it is Pareto optimal (Miettinen, 1999). Although this scalarization approach is well known in the multi-objective optimization community for many years, it is rarely used for gradient-based optimization due to its nonsmoothness. Even when all objective functions $\{f_1, \cdots, f_m\}$ are differentiable, the Tchebycheff scalarization (5) is non-differentiable, and hence suffers from a slow convergence rate by subgradient descent (Goffin, 1977) as illustrated in Figure 2. A detailed analysis will be given in the next section.

Adaptive Gradient Algorithms Due to the shortcomings of scalarization methods, many adaptive gradient algorithms have been proposed to find a Pareto stationary solution (Fliege & Svaiter, 2000; Schäffler et al., 2002; Désidéri, 2012). We have the following definition:

Definition 2.4 (Pareto Stationary Solution)**.** *A solution $\boldsymbol{x} \in \mathcal{X}$ is Pareto stationary if there exists a set of weights $\boldsymbol{\alpha} \in \boldsymbol{\Delta}^{m-1} = \{\boldsymbol{\alpha} | \sum_{i=1}^{m} \alpha_i = 1, \alpha_i \geq 0 \ \forall i\}$ such that the convex combination of gradients $\sum_{i=1}^{m} \alpha_i \nabla f_i(\boldsymbol{x}) = \mathbf{0}$.*

The Pareto stationarity is a necessary condition for Pareto optimality. In addition, when all objectives are convex and $\alpha_i > 0 \ \forall i$, it is also the Karush-Kuhn-Tucker (KKT) sufficient condition for Pareto optimality (Miettinen, 1999).

Adaptive gradient algorithms typically aim to find a valid gradient direction to improve the performance of all objectives simultaneously at each iteration. For example, the multiple gradient descent algorithm (MGDA) (Désidéri, 2012) calculates a gradient direction $\boldsymbol{d}_t = \sum_{i=1}^{m} \alpha_i \nabla f_i(\boldsymbol{x})$ at iteration t that is a weighted convex combination of all objective gradients by solving the following problem:

$$\begin{aligned} &\min_{\alpha_i} ||\sum\nolimits_{i=1}^{m} \alpha_i \nabla f_i(\boldsymbol{x}_t)||_2^2, \\ &s.t. \ \sum\nolimits_{i=1}^{m} \alpha_i = 1, \quad \alpha_i \geq 0, \forall i = 1, ..., m. \end{aligned} \tag{7}$$

According to Désidéri (2012), the obtained $\boldsymbol{d}_t$ should either be a valid gradient direction to improve all objectives or $\boldsymbol{d}_t = \mathbf{0}$ which means the solution $\boldsymbol{x}_t$ is Pareto stationary. Therefore, we can use a simple gradient-based method to find a Pareto stationary solution (Fliege et al., 2019). In recent years, many work have adopted these gradient-based algorithms to tackle multi-objective optimization problems in the machine learning community (Sener & Koltun, 2018; Lin et al., 2019; Mahapatra & Rajan, 2020; Ma et al., 2020; Momma et al., 2022). Similar adaptive gradient methods are also developed to solve multi-task learning problems (Yu et al., 2020; Liu et al., 2021a;b; 2022; Navon et al., 2022; Senushkin et al., 2023; Lin et al., 2023). Stochastic multi-objective optimization is another important topic for real-world applications (Liu & Vicente, 2021; Zhou et al., 2022; Fernando et al., 2023; Chen et al., 2023; Xiao et al., 2023).

However, at each iteration, these adaptive gradient algorithms typically need to calculate the gradients for each objective separately (e.g., via total m different backpropagation) and solve a quadratic programming problem such as (7). This high pre-iteration complexity will lead to high computational overhead for solving large-scale problems such as training a deep neural network model. In addition, some recent works show that many adaptive gradient methods cannot significantly outperform a well-tuned or even random linear scalarization for deep multi-task learning problems (Kurin et al., 2022; Xin et al., 2022; Lin et al., 2022a; Royer et al., 2023), although linear scalarization fails to fully explore the whole Pareto front (Hu et al., 2023).

3. Smooth Tchebycheff Scalarization

Instead of proposing another adaptive gradient algorithm, this work revisits the straightforward scalarization approach. By leveraging the smooth optimization approach, we propose a lightweight and efficient smooth Tchebycheff scalarization with a fast convergence rate, low pre-iteration complexity, and promising theoretical properties for multi-objective optimization.

3.1. Smoothing Method for Nonsmooth Problem

We first analyze the nonsmoothness of classical Tchebycheff scalarization, and then briefly introduce the smooth optimization method that can be used to tackle this issue. We have the following definition of smoothness:

Definition 3.1 (Smoothness)**.** *A function $g(\boldsymbol{x})$ is L-smooth if it has L-Lipschitz continuous gradient $\nabla g(\boldsymbol{x})$:*

$$||\nabla g(\boldsymbol{x}) - \nabla g(\boldsymbol{y})|| \leq L||\boldsymbol{x} - \boldsymbol{y}||, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}. \quad (8)$$

Nonsmoothness of Tchebycheff Scalarization The nonsmoothness of classical Tchebycheff scalarization (5) comes from the non-differentiable `max` operator on all objectives. Even when all objective functions $f_i(\boldsymbol{x})$ are convex and smooth, the max function $g(x) = \max_i\{f_i(\boldsymbol{x})\}$ is convex but not smooth (Boyd & Vandenberghe, 2004). In other words, this function is not differentiable and only has subgradients with respect to the decision variable $\boldsymbol{x}$. It is well known that the subgradient method needs a large number of iterations in the order of $O(\frac{1}{\epsilon^2})$ to achieve an ϵ-optimal solution $\hat{\boldsymbol{x}}$ (e.g., $||f(\hat{\boldsymbol{x}}) - f(\boldsymbol{x}^*)|| \leq \epsilon$) for nonsmooth convex optimization (Goffin, 1977). In contrast, the required gradient descent iteration is $O(\frac{1}{\epsilon})$ for smooth convex optimization. The slow convergence rate of the Tchebycheff scalarization is also illustrated in Figure 2.

Smooth Optimization Smooth optimization is a principled and powerful approach to handling nonsmooth optimization problems (Nesterov, 2005; Beck & Teboulle, 2012; Chen, 2012). According to Nesterov (2005), the large $O(\frac{1}{\epsilon^2})$ iterations is the worst-case estimate for a general nonsmooth problem, which can be significantly reduced if we have prior knowledge about the problem structure. According to Beck & Teboulle (2012) and Chen (2012), we have the following definition of smoothing function:

Definition 3.2 (Smoothing Function)**.** *We call $g_\mu : \mathcal{X} \to \mathbb{R}$ a smoothing function of a continuous function g, if for any $\mu > 0$, g_μ is continuously differentiable in $\mathcal{X}$ and satisfies the following conditions:*

1. $\lim_{\boldsymbol{z} \to \boldsymbol{x}, \mu \downarrow 0} g_\mu(\boldsymbol{z}) = g(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{X};$
2. *(Lipschitz smoothness with respect of $\boldsymbol{x}$) there exist constants K and $\alpha > 0$ irrelevant to μ such that g_μ is smooth on $\mathcal{X}$ with Lipschitz constant $L_{g_\mu} = K + \frac{\alpha}{\mu}$.*

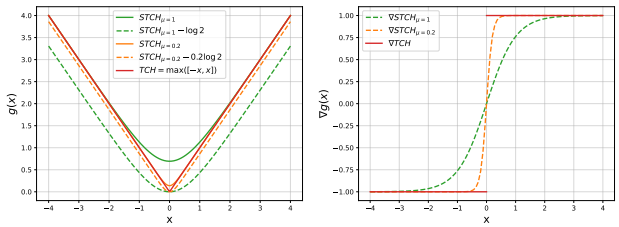
With a properly chosen smoothing parameter μ, we can find an ϵ-optimal solution to the *original nonsmooth function* $g(\boldsymbol{x})$ by optimizing the smoothing function $g_\mu(\boldsymbol{x})$ within $O(\frac{1}{\epsilon})$ iterations (Beck & Teboulle, 2012).

3.2. Smooth Tchebycheff Scalarization

By levering the infimal convolution smoothing method developed in Beck & Teboulle (2012), we propose the following smooth Tchebycheff (STCH) scalarization for multi-objective optimization:

$$g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \mu \log \left(\sum_{i=1}^{m} e^{\frac{\lambda_i (f_i(\boldsymbol{x}) - z_i^*)}{\mu}} \right), \quad (9)$$

where μ is the smoothing parameter, m is the number of objectives, $\boldsymbol{\lambda} \in \boldsymbol{\Delta}^{m-1}$ and $\boldsymbol{z}^* \in \mathbb{R}^m$ are the preference vector and ideal objective values respectively as in the classical Tchebycheff scalarization (5). Like other scalarization methods, once a specific preference $\boldsymbol{\lambda}$ is given, we can directly optimize the STCH scalarization with a straightforward gradient descent algorithm as shown in **Algorithm 1**.



(a) Smoothing Function (b) Smoothing Gradient

Figure 3. **Smoothing a Nonsmooth Function: (a)** The simple TCH scalarization function $g(x) = \max(f_1(x) = -x, f_2(x) = x)$ and its corresponding STCH scalarization with different smoothing parameters $\mu = 1$ and $\mu = 0.2$. The $g(x)$ is tightly bounded from above and below with a small μ. **(b)** The gradient and smoothed gradients. TCH scalarization does not have gradient at $x = 0$ while STCH is differentiable everywhere.

Its pre-iteration complexity is much lower than the adaptive gradient methods that need to solve a quadratic programming problem such as (7) at each iteration.

The STCH scalarization (9) has many good theoretical properties. In the rest of this subsection, we illustrate why it is a promising alternative to the classical nonsmooth TCH scalarization (5). A theoretical analysis and discussion from the viewpoint of multi-objective optimization will be provided in the next subsection.

First of all, the STCH scalarization is a proper smooth approximation to the classical nonsmooth TCH counterpart:

Proposition 3.3 (Smooth Approximation)**.** *The smooth Tchebycheff scalarization $g_\mu^{(STCH)}(\boldsymbol{x}|\boldsymbol{\lambda})$ is a smoothing function of the classical Tchebycheff scalarization $g^{(TCH)}(\boldsymbol{x}|\boldsymbol{\lambda})$ that satisfies Definition 3.2.*

In other words, it shares similar properties with TCH scalarization that is promising for multi-objective optimization, while it is much easier to optimize via gradient-based methods due to its differentiable nature. This proposition is a direct corollary of Proposition 4.1 in Beck & Teboulle (2012). It is easy to check that the classic TCH scalarization $g^{(\text{TCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$ is properly bounded from above and below:

Proposition 3.4 (Bounded Approximation)**.** *For any solution $\boldsymbol{x} \in \mathcal{X}$, we have*

$$g_\mu^{(STCH)}(\boldsymbol{x}|\boldsymbol{\lambda}) - \mu \log m \leq g^{(TCH)}(\boldsymbol{x}|\boldsymbol{\lambda}) \leq g_\mu^{(STCH)}(\boldsymbol{x}|\boldsymbol{\lambda}). \quad (10)$$

This means that the STCH scalarization $g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$ is a uniform smooth approximation (Nesterov, 2005) of $g^{(\text{TCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$. When the smoothing parameter μ is small enough, the point-wise bound is tight for all $\boldsymbol{x} \in \mathcal{X}$. This is also the case for the optimal solution $\boldsymbol{x}^* = \arg\min g^{(\text{TCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$. An illustration can be found in Figure 3. This good property motivates us to find an approximate optimal solution for the nonsmooth $g^{(\text{TCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$ by optimizing the smooth counterpart $g^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$. We have the following proposition:

Algorithm 1 STCH for Multi-Objective Optimization

Input: Preference $\boldsymbol{\lambda}$, Initial $\boldsymbol{x}_0$, Step Size $\{\eta_t\}_{t=0}^T$
for $t = 1$ to T **do**
  $\boldsymbol{x}_t = \boldsymbol{x}_{t-1} - \eta_t \nabla g_\mu^{(\text{STCH})}(\boldsymbol{x}_{t-1}|\boldsymbol{\lambda})$
end for
Output: Final Solution $\boldsymbol{x}_T$

Lemma 3.5 (Convexity)**.** *If all objective functions $\{f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})\}$ are convex, then the STCH scalarization $g_\mu^{(STCH)}(\boldsymbol{x}|\boldsymbol{\lambda})$ with any valid μ and $\boldsymbol{\lambda}$ is convex.*

Proposition 3.6 (Iteration Complexity)**.** *If all objective functions are convex, with a properly chosen smoothing parameter μ, we can obtain an ϵ-optimal solution to the nonsmooth $g_\mu^{(TCH)}(\boldsymbol{x}|\boldsymbol{\lambda})$ within $O(\frac{1}{\epsilon})$ iterations by solving the smooth counterpart $g_\mu^{(STCH)}(\boldsymbol{x}|\boldsymbol{\lambda})$.*

This proposition can be established by combining the above results on smooth approximation (Proposition 3.3), bounded approximation guarantee (Proposition 3.4) and the convexity of STCH scalarization (Lemma 3.5), as well as the convergence rate of fast gradient-based methods for smooth convex optimization in a straightforward manner. A seminal proof can be found in Theorem 3.1 of Beck & Teboulle (2012) for general smooth optimization. This required number of iterations is much lower than $O(\frac{1}{\epsilon^2})$ for directly optimizing the nonsmooth TCH scalarization (5) with subgradient descent.

3.3. Properties for Multi-Objective Optimization

In addition to being a promising smooth approximation of classical Tchebycheff scalarization, the smooth Tchebycheff scalarization itself also has several good theoretical properties for multi-objective optimization. First of all, its optimal solution is (weakly) Pareto optimal:

Theorem 3.7 (Pareto Optimality of the Solution)**.** *The optimal solution of STCH scalarization (9) is weakly Pareto optimal of the original multi-objective optimization problem (1). In addition, the solution is Pareto optimal if either*

1. *all preference coefficients are positive ($\lambda_i > 0\ \forall i$) or*
2. *the optimal solution is unique.*

Proof Sketch. These properties can be proved by contradiction based on Definition 2.2 for (weakly) Pareto optimality and the form of STCH Scalarization (9). A detailed proof can be found in Appendix A.1. □

According to Theorem 3.7, for *any* smoothing parameter μ, the solution of STCH scalarization (9) with *any* valid preference $\boldsymbol{\lambda}$ is (weakly) Pareto optimal. However, to find *all* Pareto solutions, we have an additional requirement on the smoothing parameter μ:

Theorem 3.8 (Ability to Find All Pareto Solutions)**.** *Under mild conditions, there exists a μ^* such that, for any $0 < \mu < \mu^*$, every Pareto solution of the original multi-objective optimization problem (1) is an optimal solution of the STCH scalarization problem (9) with some valid preference $\boldsymbol{\lambda}$.*

Proof Sketch. We can prove this theorem by analyzing the geometry relation between the level surface of STCH scalarization and the Pareto front. The key step is to find a theoretical upper bound of μ^* such that the level surface of STCH scalarization with any $\mu < \mu^*$ is totally enclosed by the Pareto front. The mild conditions are formally defined in Assumption A.1, and detailed proof can be found in Appendix A.2. □

Theorem 3.8 provides a sufficient condition for STCH scalarization to find all Pareto solutions of the multi-objective optimization problem (1). It can be treated as a smooth version of Theorem 2.3 for classical TCH scalarization. We have a stronger corollary for the convex Pareto front:

Corollary 3.9. *If the Pareto front is convex, then for any μ, every Pareto solution of the original multi-objective optimization problem (1) is an optimal solution of the STCH scalarization problem (9) with some valid preference $\boldsymbol{\lambda}$.*

The above analyses are for the global optimal properties of STCH scalarization. However, many real-world multi-objective optimization problems could be complicated and highly non-convex. In this case, we have the following local convergence guarantee.

Theorem 3.10 (Convergence to Pareto Stationary Solution)**.** *If there exists a solution $\hat{\boldsymbol{x}}$ such that $\nabla g_{\mu}^{(STCH)}(\hat{\boldsymbol{x}}|\boldsymbol{\lambda}) = 0$, then $\hat{\boldsymbol{x}}$ is a Pareto stationary solution of the original multi-objective optimization problem (1).*

Proof Sketch. This theorem can be proved by analyzing the form of the gradient for STCH scalarization. A crucial step is to show that the situation of Pareto stationarity in Definition 2.4 is satisfied when $\nabla g_{\mu}^{(\text{STCH})}(\boldsymbol{x}|\hat{\boldsymbol{\lambda}}) = 0$. A detailed proof can be found in Appendix A.3. □

There is a rich set of efficient gradient-based methods (Nocedal & Wright, 1999) that can be used to obtain a stationary solution for $g_{\mu}^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$. Therefore, our proposed gradient-based STCH scalarization approach has a Pareto stationary guarantee similar to those of adaptive gradient algorithms, while allowing decision-makers to assign their preference among the objectives easily. Additionally, at each iteration, STCH scalarization does not need to calculate separate gradients for each objective and can also avoid solving a quadratic programming problem such as (7).

3.4. STCH Scalarization for Pareto Set Learning

Classic scalarization and adaptive gradient methods all focus on finding a single or finite set of Pareto solutions, but the whole Pareto set could be an $(m-1)$-dimensional manifold that contains infinite solutions (Hillermeier, 2001). Recently, some works have been proposed to build a set model to approximate the whole Pareto set or front (Parisi et al., 2014; Dosovitskiy & Djolonga, 2019; Lin et al., 2020b; Navon et al., 2021; Ruchte & Grabocka, 2021; Lin et al., 2022b;c; Chen & Kwok, 2022; Jain et al., 2023; Zhang et al., 2023). A typical Pareto set model can be defined as:

$$\boldsymbol{x}^*(\boldsymbol{\lambda}) = h_{\boldsymbol{\theta}}(\boldsymbol{\lambda}) = \arg\min_{\boldsymbol{x}\in\mathcal{X}} g(\boldsymbol{x}|\boldsymbol{\lambda}), \forall \boldsymbol{\lambda} \in \Delta^{m-1}, \quad (11)$$

where a model $h_{\boldsymbol{\theta}}(\boldsymbol{\lambda})$ with parameter $\boldsymbol{\theta}$ maps any valid preference $\boldsymbol{\lambda} \in \Delta^{m-1}$ to its corresponding optimal solution $\boldsymbol{x}^*(\boldsymbol{\lambda})$ with respect to some scalarization function $g(\boldsymbol{x}|\boldsymbol{\lambda})$. The quality of this set model heavily depends on the scalarization function it uses. For example, a set model with linear scalarization might miss the solutions on the non-convex part of the Pareto front, and the classical Tchebycheff scalarization will lead to nonsmooth optimization for set model learning. Our proposed STCH scalarization $g_{\mu}^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$ can serve as a promising drop-in replacement to efficiently learn a better Pareto set model.

4. Experimental Studies

We evaluate the proposed STCH scalarization approach on finding a single Pareto solution and the whole Pareto set.

4.1. Multi-Task Learning

The STCH scalarization can be used to find a solution with balanced trade-off for multi-task learning problems.

Baseline Methods We compare the STCH scalarization with the single-task learning (STL) baseline and *(1) other scalarization methods*: equal weight (EW), geometric mean loss (GLS) (Chennupati et al., 2019), random weights (RLW) (Lin et al., 2022a), and classical Tchebycheff scalarization (TCH); *(2) adaptive loss methods*: UW (Kendall et al., 2018), DWA (Liu et al., 2019), IMTL-L, and IGBv2 (Dai et al., 2023); *(3) adaptive gradient methods:* MGDA (Sener & Koltun, 2018), GradNorm (Chen et al., 2018), PCGrad (Yu et al., 2020), GradDrop (Chen et al., 2020), GradVac (Wang & Tsvetkov, 2021), IMTL-G, CAGrad (Liu et al., 2021a), MTAdam (Malkiel & Wolf, 2021), Nash-MTL (Navon et al., 2022), MetaBalance (He et al., 2022), MoCo (Fernando et al., 2023), Alighed-MTL (Senushkin et al., 2023), and DB-MTL (Lin et al., 2023); *(4) hybrid loss-gradient balancing method:* IMTL (Liu et al., 2021b). All methods are implemented with the LibMTL library (Lin & Zhang, 2023). The problem and model training details can be found in Appendix D.1.

Table 2. Results on the NYUv2 dataset.

	Segmentation		**Depth Estimation**		**Surface Normal Prediction**					$\Delta_p \uparrow$
					Angle Distance		Within t°			
	mIoU↑	PAcc↑	AErr↓	RErr↓	**Mean**↓	**MED**↓	**11.25**↑	**22.5**↑	**30**↑	
Single Task Baseline										
STL	38.30	63.76	0.6754	0.2780	25.01	19.21	30.14	57.20	69.15	0.00
Adaptive Gradient Method										
MGDA	30.47	59.90	0.6070	0.2555	24.88	19.45	29.18	56.88	69.36	-1.66
GradNorm	20.09	52.06	0.7200	0.2800	**24.83**	**18.86**	**30.81**	**57.94**	**69.73**	-11.7
PCGrad	38.06	64.64	0.5550	0.2325	27.41	22.80	23.86	49.83	63.14	+1.11
GradDrop	39.39	65.12	0.5455	0.2279	27.48	22.96	23.38	49.44	62.87	+2.07
GradVac	37.53	64.35	0.5600	0.2400	27.66	23.38	22.83	48.66	62.21	-0.49
IMTL-G	39.35	65.60	0.5426	0.2256	26.02	21.19	26.20	53.13	66.24	+4.77
CAGrad	39.18	64.97	0.5379	0.2229	25.42	20.47	27.37	54.73	67.73	+5.81
MTAdam	39.44	65.73	0.5326	0.2211	27.53	22.70	24.04	49.61	62.69	+3.21
Nash-MTL	40.13	65.93	0.5261	0.2171	25.26	20.08	28.40	55.47	68.15	+7.65
MetaBalance	39.85	65.13	0.5445	0.2261	27.35	22.66	23.70	49.69	63.09	+2.67
MoCo	40.30	66.07	0.5575	0.2135	26.67	21.83	25.61	51.78	64.85	+4.85
Aligned-MTL	40.82	66.33	0.5300	0.2200	25.19	19.71	28.88	56.23	68.54	+8.16
IMTL	41.19	66.37	0.5323	0.2237	26.06	20.77	26.76	53.48	66.32	+6.45
DB-MTL	**41.42**	**66.45**	0.5251	0.2160	25.03	19.50	28.72	56.17	68.73	**+8.91**
Adaptive Loss Method										
UW	36.87	63.17	0.5446	0.2260	27.04	22.61	23.54	49.05	63.65	+0.91
DWA	39.11	65.31	0.5510	0.2285	27.61	23.18	24.17	50.18	62.39	+1.93
IMTL-L	39.78	65.27	0.5408	0.2347	26.26	20.99	26.42	53.03	65.94	+4.39
IGBv2	38.03	64.29	0.5489	0.2301	26.94	22.04	24.77	50.91	64.12	+2.11
Scalarization Method										
EW	39.29	65.33	0.5493	0.2263	28.15	23.96	22.09	47.50	61.08	+0.88
GLS	39.78	65.63	0.5318	0.2272	26.13	21.08	26.57	52.83	65.78	+5.15
RLW	37.17	63.77	0.5759	0.2410	28.27	24.18	22.26	47.05	60.62	-2.16
TCH	34.09	59.76	0.5931	0.2524	28.30	23.99	21.87	47.13	60.61	-5.67
STCH (Ours)	41.35	66.07	**0.4965**	**0.2010**	26.55	21.81	24.84	51.39	64.86	+8.54

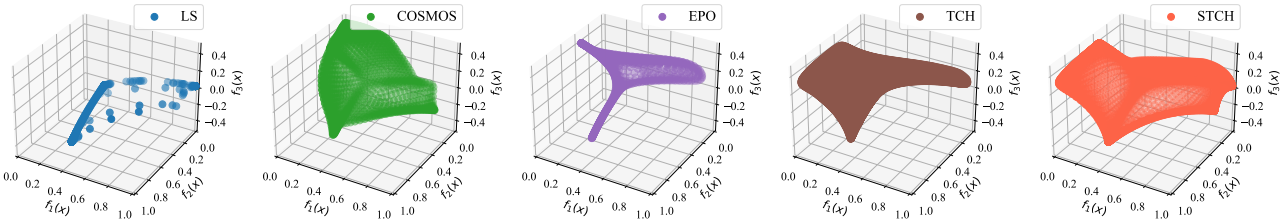
Evaluation Metrics In addition to reporting the commonly used metrics for each task, we also report the relative average performance improvement (Maninis et al., 2019; Vandenhende et al., 2021) of each method over the STL baseline $\Delta_p = \frac{1}{T}\sum_i^T \Delta_{p,t}$ with $\Delta_{p,t} = 100\% \times \frac{1}{N_t}\sum_{i=1}^{N_t}(-1)^{s_{t,i}}\frac{M_{t,i}-M_{t,i}^{\text{STL}}}{M_{t,i}^{\text{STL}}}$, where T is the number of tasks, N_t is the number of metrics for task t, $M_{t,i}$ is the i-th metric of the MTL method on task t, and $s_{t,i} = 0$ if a large value is better for the i-th metric on task t and 1 otherwise. A simple uniform preference is used for STCH. All methods are evaluated on three popular MTL datasets:

NYUv2 (Silberman et al., 2012) is an indoor scene understanding dataset with 3 tasks on semantic segmentation, depth estimation, and surface normal prediction. The result in Table 2 shows that STCH scalarization can find a good trade-off solution with the second best overall Δ_p among all methods, which is only outperformed by a very recently proposed adaptive gradient method DB-MTL. Its large performance improvement over the classic TCH scalarization counterpart fully demonstrates the importance of smoothness for multi-objective optimization.

Office-31 (Saenko et al., 2010) is an image classification dataset across 3 domains (Amazon, DSLR, and Webcam). According to Table 4, STCH scalarization can achieve the best overall performance among all methods. In fact, it is the only method that can dominate the STL baseline on all tasks. The full results can be found in Appendix E.1.

QM9 (Ramakrishnan et al., 2014) is a molecular property prediction dataset with 11 tasks. The results in Table 4 suggest that STCH scalarization has a very-close-to-best overall Δ_p with DB-MTL, while enjoying a much faster run time. The full results can be found in Appendix E.2.

Given its simplicity and promising performance, STCH scalarization can serve as a strong baseline for MTL.



(a) Linear Scalarization (b) COSMOS (c) EPO (d) TCH Scalarization (e) STCH Scalarization

Figure 4. The learned Pareto fronts for the 3-objective rocket injector design problem with different scalarization methods.

Table 3. Results (hypervolume difference ΔHV $\downarrow$) on 6 synthetic benchmark problems and 5 real-world engineering design problems.

	F1	F2	F3	F4	F5	F6	BarTruss	HatchCover	DiskBrake	GearTrain	RocketInjector
LS	1.64e-02	1.37e-02	9.40e-02	2.26e-01	1.72e-01	2.54e-01	8.03e-03	**7.89e-03**	4.05e-02	4.01e-03	1.42e-01
COSMOS	1.58e-02	1.52e-02	1.28e-02	1.49e-02	1.32e-02	1.90e-02	8.24e-03	2.87e-02	4.33e-02	3.50e-03	3.80e-02
EPO	1.13e-02	7.66e-03	2.02e-02	1.08e-02	8.29e-03	1.96e-02	1.13e-02	1.20e-02	3.38e-02	3.46e-03	5.82e-02
TCH	9.05e-03	7.97e-03	1.84e-02	8.76e-03	6.86e-03	1.45e-02	9.05e-03	1.01e-02	3.78e-02	3.91e-03	2.73e-02
STCH	**5.95e-03**	**5.73e-03**	**9.58e-03**	**6.73e-03**	**5.99e-03**	**1.16e-02**	**5.65e-03**	7.97e-03	**2.79e-02**	**3.17e-03**	**1.08e-02**

Table 4. Results on the Office-31 and QM9 datasets.

	Office-31	QM9	
	$\Delta_p \uparrow$	$\Delta_p \uparrow$	$T \downarrow$
Single Task Baseline			
STL	0.00	0.00	-
Adaptive Gradient Method			
MGDA	-0.27±0.15	-103.0±8.62	5.70
GradNorm	-0.59±0.94	-227.5±1.85	4.47
PCGrad	-0.68±0.57	-117.8±3.97	4.92
GradDrop	-0.59±0.46	-191.4±9.62	1.34
GradVac	-0.58±0.78	-150.7±7.41	5.06
IMTL-G	-0.97±0.95	-1250±90.9	4.18
CAGrad	-1.14±0.85	-87.25±1.51	4.71
MTAdam	-0.60±0.93	-1403±203	-
Nash-MTL	+0.24±0.89	-73.92±2.12	6.53
MetaBalance	-0.63±0.30	-125.1±7.98	-
MoCo	+0.89±0.26	-1314±65.2	4.51
Aligned-MTL	-0.90±0.48	-80.58±4.18	4.45
IMTL	-1.02±0.92	-104.3±11.7	4.62
DB-MTL	+1.05±0.20	**-58.10±3.89**	4.56
Adaptive Loss Method			
UW	-0.56±0.90	-92.35±13.9	1.00
DWA	-0.70±0.62	-160.9±16.7	1.00
IMTL-L	-0.63±0.58	-77.06±11.1	1.00
IGBv2	+0.56±0.25	-99.86±10.4	1.00
Scalarization Method			
EW	-0.61±0.67	-146.3± 7.86	1.00
GLS	-1.63±0.61	-81.16±15.5	1.00
RLW	-0.59±0.95	-200.9±13.4	1.00
TCH	-0.71±0.56	-252.2±16.6	1.00
STCH (Ours)	**+1.48±0.31**	-58.14±4.18	1.00

4.2. Efficient Pareto Set Learning

The STCH scalarization can also be used to improve the performance of Pareto set learning.

Baseline Methods We build and compare Pareto set learning models with different scalarization methods: (1) Linear Scalarization (LS), (2) Conditioned One-shot Multi-Objective Search (COSMOS) (Ruchte & Grabocka, 2021), (3) Exact Preference Optimization (EPO) (Mahapatra & Rajan, 2020), (4) classic TCH scalarization, and (5) our proposed STCH scalarization.

Evaluation Metric Following the related work, we use the hypervolume difference (ΔHV) (Zitzler et al., 2007) to measure the qualities of approximate Pareto fronts learned by different methods. A lower ΔHV means that the learned Pareto front is closer to the ground truth Pareto front and hence has better performance. A detailed definition of ΔHV can be found in Appendix D.2.3.

Results and Analysis We evaluate all methods on six synthetic benchmark problems (F1-F6) as well as five real-world engineering design problems for Bar Truss, Hatch Cover, Disk Brake, Gear Train, and Rocket Injector. According to Table 3, the Pareto set learning model with STCH scalarization performs the best on 10 out of 11 problems. The approximate Pareto fronts learned by different scalarization methods for the rocket injector design are shown in Figure 4. The Pareto front learned by STCH can find better trade-offs than others. These results demonstrate that STCH is a promising scalarization approach for Pareto set learning. Problem details can be found in Appendix D.2.1 and D.2.2.

Table 5. The results of STCH with different smooth parameters μ on 5 real-world engineering desgin problems.

	BarTruss	HatchCover	DiskBrake	GearTrain	RocketInjector
TCH	9.05e-03	1.01e-02	3.78e-02	3.91e-03	2.73e-02
STCH($\mu = 0.01$)	6.94e-03	8.47e-03	3.12e-02	3.79e-03	**1.95e-02**
STCH($\mu = 0.1$)	6.85e-03	8.76e-03	**2.79e-02**	**3.17e-03**	2.00e-02
STCH($\mu = 0.5$)	**6.21e-03**	**7.81e-03**	2.96e-02	3.36e-03	2.24e-02
STCH($\mu = 1$)	6.95e-03	8.02e-03	3.32e-02	3.62e-03	3.24e-02

4.3. Effect of Different Smooth Levels

The parameter μ controls the smoothing level of the STCH scalarization, which acts as a hyperparameter in our proposed method. In this work, we do not aggressively tune μ for each problem. To investigate the effect of μ, we conduct a new experiment to test STCH with different smooth levels μ for the multi-objective engineering design problem. As shown in Table 5, STCH scalarization with a small μ works reasonably well, but there is no single μ that can achieve the best performance for all problems.

The smooth parameter μ does have an impact on the convergence rate of the gradient-based algorithm. Roughly speaking, a larger μ can yield a lower iteration complexity for the smoothed function, while a small enough μ is required to obtain a good approximate solution for the original non-smooth problem. To better leverage this trade-off, a few homotopy methods (Allen-Zhu & Hazan, 2016; Xu et al., 2016) have been proposed to gradually decrease μ in a stagewise manner for the Nesterov smoothing approach (Nesterov, 2005). This method can reduce the iteration complexity from $O(1/\epsilon)$ to $\tilde{O}(1/\epsilon^{1-\theta})$ where $\theta \in (0, 1]$ represents the local sharpness around the optimal solution.

However, the homotopy schedule could be problem-specific and should be tuned for each new problem. We do not observe a clear performance improvement by using the homotopy method and, therefore, simply use a fixed μ in all our experiments. How to adaptively adjust the smooth parameter μ for each problem during the optimization process could be an interesting research direction in future work.

5. Local and Global Optimality Guarantee

Although Theorem 3.8 provides a promising sufficient condition for STCH scalarization to find all Pareto solutions, the simple gradient-based algorithm can only guarantee to find Pareto stationary solutions for a general multi-objective optimization problem as in Theorem 3.10. We would like to make the following remarks on this theoretical guarantee:

Similar Guarantee with the Other Methods The global optimality guarantee for STCH is on par with the optimality guarantee for the classic TCH scalarization counterpart as in Theorem 2.3. Every Pareto optimal solution can be found by TCH scalarization, but the subgradient method can only guarantee to find a Pareto stationary solution. This guarantee is also on par with the adaptive gradient methods that are popular in the current literature, but our proposed method has a much lower pre-iteration complexity. Under a stronger assumption, such as all objectives are convex (and hence the STCH function is convex), the proposed method can find the global optimal solution.

Advanced Method for Global Optimality It could be extremely hard, if not impossible, to provide a convergence guarantee to find the global optimal solution for a general non-convex optimization problem. Some advanced global optimization methods, such as homotopy/graduated optimization (Dunlavy & O'Leary, 2005; Hazan et al., 2016), could be helpful, but it is still a very difficult open problem for gradient-based optimization. We will explore this direction in future work. On the other hand, a local optimal solution is usually good enough for deep learning applications. According to the experimental results, our proposed method achieves promising performance on the MTL problems, while enjoying a significantly faster runtime than those adaptive gradient methods.

6. Conclusion, Limitation, and Future Work

Conclusion In this work, we have proposed a novel smooth Tchebycheff (STCH) scalarization approach for efficient gradient-based multi-objective optimization. It has a fast convergence rate and low pre-iteration complexity, while enjoying many good theoretical properties. Experimental studies also show that STCH scalarization can obtain promising performance on various multi-objective optimization problems. We believe it can serve as a powerful yet straightforward method to tackle differentiable multi-objective optimization.

Limitation This work only focuses on unconstrained and deterministic multi-objective optimization. It could be interesting to investigate how to properly handle different constraints and tackle stochastic optimization with STCH scalarization in future work. A brief discussion on these directions can be found in Appendix B.4 and B.5.

Acknowledgements

The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (GRF Project No. CityU11215622), the National Natural Science Foundation of China (Grant No. 62106096), the Natural Science Foundation of Guangdong Province (Grant No. 2024A1515011759), the National Natural Science Foundation of Shenzhen (Grant No. JCYJ20220530113013031).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Allen-Zhu, Z. and Hazan, E. Optimal black-box reductions between optimization objectives. *Advances in Neural Information Processing Systems*, 29, 2016.

Amir, H. M. and Hasegawa, T. Nonlinear mixed-discrete structural optimization. *Journal of Structural Engineering*, 115(3):626–646, 1989.

Beck, A. and Teboulle, M. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

Bowman, V. J. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Multiple Criteria Decision Making*, pp. 76–86. Springer, 1976.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge university press, 2004.

Chen, L., Fernando, H., Ying, Y., and Chen, T. Three-way trade-off in multi-objective learning: Optimization, generalization and conflict-avoidance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Chen, W. and Kwok, J. Multi-objective deep learning with adaptive reference vectors. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:32723–32735, 2022.

Chen, X. Smoothing methods for nonsmooth, nonconvex minimization. *Mathematical Programming*, 134:71–99, 2012.

Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 794–803, 2018.

Chen, Z., Ngiam, J., Huang, Y., Luong, T., Kretzschmar, H., Chai, Y., and Anguelov, D. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33: 2039–2050, 2020.

Cheng, F. and Li, X. Generalized center method for multiobjective engineering optimization. *Engineering Optimization*, 31(5):641–661, 1999.

Chennupati, S., Sistu, G., Yogamani, S., and A Rawashdeh, S. Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Choo, E. U. and Atkins, D. Proper efficiency in nonconvex multicriteria programming. *Mathematics of Operations Research*, 8(3):467–470, 1983.

Dai, Y., Fei, N., and Lu, Z. Improvable gap balancing for multi-task learning. In *Uncertainty in Artificial Intelligence*, pp. 496–506. PMLR, 2023.

Das, I. and Dennis, J. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69, 1997.

Deb, K. and Srinivasan, A. Innovization: Innovating design principles through optimization. In *Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1629–1636, 2006.

Désidéri, J.-A. Mutiple-gradient descent algorithm for multiobjective optimization. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012)*, 2012.

Dosovitskiy, A. and Djolonga, J. You only train once: Loss-conditional training of deep networks. *International Conference on Learning Representations (ICLR)*, 2019.

Dunlavy, D. M. and O'Leary, D. P. Homotopy optimization methods for global optimization. In *Technical Report*, 2005.

Ehrgott, M. *Multicriteria Optimization*, volume 491. Springer Science & Business Media, 2005.

Fernando, H. D., Shen, H., Liu, M., Chaudhury, S., Murugesan, K., and Chen, T. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *International Conference on Learning Representations (ICLR)*, 2023.

Fliege, J. and Svaiter, B. F. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

Fliege, J. and Vaz, A. I. F. A method for constrained multiobjective optimization based on sqp techniques. *SIAM Journal on Optimization*, 26(4):2091–2119, 2016.

Fliege, J., Vaz, A. I. F., and Vicente, L. N. Complexity of gradient descent for multiobjective optimization. *Optimization Methods and Software*, 34(5):949–959, 2019.

Geoffrion, A. M. Solving bicriterion mathematical programs. *Operations Research*, 15(1):39–54, 1967.

Gidel, G., Jebara, T., and Lacoste-Julien, S. Frank-wolfe algorithms for saddle point problems. In *Artificial Intelligence and Statistics*, pp. 362–371. PMLR, 2017.

Goffin, J.-L. On convergence rates of subgradient optimization methods. *Mathematical Programming*, 13:329–347, 1977.

Goh, C. and Yang, X. Convexification of a noninferior frontier. *Journal of Optimization Theory and Applications*, 97:759–768, 1998.

Hazan, E., Levy, K. Y., and Shalev-Shwartz, S. On graduated optimization for stochastic non-convex problems. In *Proc. International Conference on Machine Learning (ICML)*, 2016.

He, Y., Feng, X., Cheng, C., Ji, G., Guo, Y., and Caverlee, J. Metabalance: improving multi-task recommendations via adapting gradient magnitudes of auxiliary tasks. In *Proceedings of the ACM Web Conference 2022*, pp. 2205–2215, 2022.

Hillermeier, C. Generalized homotopy approach to multiobjective optimization. *Journal of Optimization Theory and Applications*, 110(3):557–583, 2001.

Hu, Y., Xian, R., Wu, Q., Fan, Q., Yin, L., and Zhao, H. Revisiting scalarization in multi-task learning: A theoretical perspective. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023.

Jain, M., Raparthy, S. C., Hernández-García, A., Rector-Brooks, J., Bengio, Y., Miret, S., and Bengio, E. Multi-objective GFlowNets. In *International Conference on Machine Learning (ICML)*, pp. 14631–14653. PMLR, 2023.

Kendall, A., Gal, Y., and Cipolla, R. 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 (CVPR)*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Kurin, V., De Palma, A., Kostrikov, I., Whiteson, S., and Mudigonda, P. K. In defense of the unitary scalarization for deep multi-task learning. *Advances in Neural Information Processing Systems*, 35:12169–12183, 2022.

Letcher, A., Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., and Graepel, T. Differentiable game mechanics. *Journal of Machine Learning Research (JMLR)*, 20(84):1–40, 2019.

Li, D. Convexification of a noninferior frontier. *Journal of Optimization Theory and Applications*, 88:177–196, 1996.

Lin, B. and Zhang, Y. Libmtl: A python library for deep multi-task learning. *Journal of Machine Learning Research*, 24(1-7):18, 2023.

Lin, B., Feiyang, Y., Zhang, Y., and Tsang, I. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022a.

Lin, B., Jiang, W., Ye, F., Zhang, Y., Chen, P., Chen, Y.-C., Liu, S., and Kwok, J. T. Dual-balancing for multi-task learning. *arXiv preprint arXiv:2308.12029*, 2023.

Lin, T., Jin, C., and Jordan, M. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning (ICML)*, pp. 6083–6093. PMLR, 2020a.

Lin, X., Zhen, H.-L., Li, Z., Zhang, Q., and Kwong, S. Pareto multi-task learning. In *Advances in Neural Information Processing Systems*, pp. 12060–12070, 2019.

Lin, X., Yang, Z., Zhang, Q., and Kwong, S. Controllable pareto multi-task learning. *arXiv preprint arXiv:2010.06313*, 2020b.

Lin, X., Yang, Z., and Zhang, Q. Pareto set learning for neural multi-objective combinatorial optimization. In *International Conference on Learning Representations (ICLR)*, 2022b.

Lin, X., Yang, Z., Zhang, X., and Zhang, Q. Pareto set learning for expensive multiobjective optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022c.

Liu, B., Liu, X., Jin, X., Stone, P., and Liu, Q. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 18878–18890, 2021a.

Liu, L., Li, Y., Kuang, Z., Xue, J., Chen, Y., Yang, W., Liao, Q., and Zhang, W. Towards impartial multi-task learning. In *International Conference on Learning Representations (ICLR)*, 2021b.

Liu, S. and Vicente, L. N. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *Annals of Operations Research*, pp. 1–30, 2021.

Liu, S., Johns, E., and Davison, A. J. End-to-end multi-task learning with attention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1871–1880, 2019.

Liu, S., James, S., Davison, A., and Johns, E. Auto-lambda: Disentangling dynamic task relationships. *Transactions on Machine Learning Research*, 2022.

Ma, P., Du, T., and Matusik, W. Efficient continuous pareto exploration in multi-task learning. *International Conference on Machine Learning (ICML)*, 2020.

Mahapatra, D. and Rajan, V. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. *International Conference on Machine Learning (ICML)*, 2020.

Malkiel, I. and Wolf, L. Mtadam: Automatic balancing of multiple training loss terms. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10713–10729, 2021.

Maninis, K.-K., Radosavovic, I., and Kokkinos, I. Attentive single-tasking of multiple tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1851–1860, 2019.

Martinez, N., Bertran, M., and Sapiro, G. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning (ICML)*, pp. 6755–6764. PMLR, 2020.

Mertikopoulos, P., Lecouat, B., Zenati, H., Foo, C.-S., Chandrasekhar, V., and Piliouras, G. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *International Conference on Learning Representations (ICLR)*, pp. 1–23, 2019.

Miettinen, K. *Nonlinear Multiobjective Optimization*, volume 12. Springer Science & Business Media, 1999.

Mokhtari, A., Ozdaglar, A., and Pattathil, S. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1497–1507. PMLR, 2020.

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

Monteiro, R. D. and Svaiter, B. F. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20(6): 2755–2787, 2010.

Navon, A., Shamsian, A., Chechik, G., and Fetaya, E. Learning the pareto front with hypernetworks. In *International Conference on Learning Representations (ICLR)*, 2021.

Navon, A., Shamsian, A., Achituve, I., Maron, H., Kawaguchi, K., Chechik, G., and Fetaya, E. Multi-task learning as a bargaining game. In *International Conference on Machine Learning (ICML)*, pp. 16428–16446. PMLR, 2022.

Nemirovski, A. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1): 229–251, 2004.

Nesterov, Y. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

Nesterov, Y. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2):319–344, 2007.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, 1999.

Palaniappan, B. and Bach, F. Stochastic variance reduction methods for saddle-point problems. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016.

Parisi, S., Pirotta, M., Smacchia, N., Bascetta, L., and Restelli, M. Policy gradient approaches for multi-objective sequential deb. In *International Joint Conference on Neural Networks*, 2014.

Ramakrishnan, R., Dral, P. O., Rupp, M., and Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):1–7, 2014.

Ray, T. and Liew, K. A swarm metaphor for multiobjective design optimization. *Engineering Optimization*, 34(2): 141–153, 2002.

Royer, A., Blankevoort, T., and Ehteshami Bejnordi, B. Scalarization for multi-task and multi-domain learning at scale. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023.

Ruchte, M. and Grabocka, J. Scalable pareto front approximation for deep multi-objective learning. In *IEEE International Conference on Data Mining (ICDM)*, 2021.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pp. 213–226. Springer, 2010.

Schäffler, S., Schultz, R., and Weinzierl, K. Stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization Theory and Applications*, 114:209–222, 2002.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pp. 525–536, 2018.

Senushkin, D., Patakin, N., Kuznetsov, A., and Konushin, A. Independent component alignment for multi-task learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20083–20093, 2023.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, pp. 746–760. Springer, 2012.

Steuer, R. E. and Choo, E.-U. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344, 1983.

Tanabe, R. and Ishibuchi, H. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020.

Vaidyanathan, R., Tucker, K., Papila, N., and Shyy, W. Cfd-based design optimization for single element rocket injector. In *41st Aerospace Sciences Meeting and Exhibit*, pp. 296, 2003.

Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., and Van Gool, L. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(7):3614–3633, 2021.

Wang, R. and Zhang, C. Stochastic smoothing accelerated gradient method for nonsmooth convex composite optimization. *arXiv preprint arXiv:2308.01252*, 2023.

Wang, Z. and Tsvetkov, Y. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations (ICLR)*, 2021.

Xiao, P., Ban, H., and Ji, K. Direction-oriented multi-objective learning: Simple and provable stochastic algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Xie, Y., Shi, C., Zhou, H., Yang, Y., Zhang, W., Yu, Y., and Li, L. Mars: Markov molecular sampling for multi-objective drug discovery. In *International Conference on Learning Representations (ICLR)*, 2021.

Xin, D., Ghorbani, B., Gilmer, J., Garg, A., and Firat, O. Do current multi-task optimization methods in deep learning even help? *Advances in Neural Information Processing Systems*, 35:13597–13609, 2022.

Xu, J., Tian, Y., Ma, P., Rus, D., Sueda, S., and Matusik, W. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International Conference on Machine Learning*, pp. 10607–10616. PMLR, 2020.

Xu, Y., Yan, Y., Lin, Q., and Yang, T. Homotopy smoothing for non-smooth problems with lower complexity than $o(1/\epsilon)$. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016.

Xu, Z., Zhang, H., Xu, Y., and Lan, G. A unified single-loop alternating gradient projection algorithm for nonconvex–concave and convex–nonconcave minimax problems. *Mathematical Programming*, 201(1):635–706, 2023.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:5824–5836, 2020.

Zhang, J., Xiao, P., Sun, R., and Luo, Z. A single-loop smoothed gradient descent-ascent algorithm for nonconvex-concave min-max problems. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 7377–7389, 2020.

Zhang, Q. and Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

Zhang, X., Lin, X., Xue, B., Chen, Y., and Zhang, Q. Hypervolume maximization: A geometric view of pareto set learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zhou, S., Zhang, W., Jiang, J., Zhong, W., Gu, J., and Zhu, W. On the convergence of stochastic multi-objective gradient manipulation and beyond. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Zitzler, E., Brockhoff, D., and Thiele, L. The hypervolume indicator revisited: On the design of pareto-compliant

indicators via weighted integration. In *International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, 2007.

We provide more discussion, details of the proposed algorithm and problem, and extra experimental results in this appendix:

- Detailed proofs for the theoretical analysis are provided in Section A.
- The details of the practical implementation for STCH scalarization can be found in Section B.
- A discussion with related work on minimax optimization are provided in Section C.
- Details of application problems and experimental setting can be found in Section D.
- More experimental results and analyses are provided in Section E.

A. Detailed Proofs

A.1. Proof of Theorem 3.7

Theorem 3.7 (Pareto Optimality of the Solution). *The optimal solution of STCH scalarization (9) is weakly Pareto optimal. In addition, the solution is Pareto optimal if either*

1. *all preference coefficients are positive ($\lambda_i > 0\ \forall i$) or*
2. *the optimal solution is unique.*

Proof. **Weakly Pareto Optimality:** We first prove the optimal solution of STCH scalarization is weakly Pareto optimal by contradiction. Let $\boldsymbol{x}^*$ be an optimal solution for the STCH scalarization $g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$ with valid preference $\boldsymbol{\lambda}$, we have:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \mu \log\left(\sum_{i=1}^m e^{\lambda_i(f_i(\boldsymbol{x})-z_i^*)/\mu}\right). \tag{12}$$

Suppose that $\boldsymbol{x}^*$ is *not* weakly Pareto optimal for the multi-objective optimization problem (1). According to Definition 2.2 on the (weakly) Pareto optimality, there exists a valid solution $\hat{\boldsymbol{x}} \in \mathcal{X}$ such that $\boldsymbol{f}(\hat{\boldsymbol{x}}) \prec_{\text{strict}} \boldsymbol{f}(\boldsymbol{x}^*)$. In other words, we have:

$$f_i(\hat{\boldsymbol{x}}) < f_i(\boldsymbol{x}^*) \quad \forall i \in \{1, ..., m\}. \tag{13}$$

Based on the above inequalities, it is easy to see:

$$\mu \log\left(\sum_{i=1}^m e^{\lambda_i(f_i(\hat{\boldsymbol{x}})-z_i^*)/\mu}\right) < \mu \log\left(\sum_{i=1}^m e^{\lambda_i(f_i(\boldsymbol{x}^*)-z_i^*)/\mu}\right) \tag{14}$$

which contradicts the optimality of $\boldsymbol{x}^*$ for the STCH scalarization (12). Therefore, $\boldsymbol{x}^*$ should be weakly Pareto optimal for the original multi-objective optimization problem.

Then we provide proof of the two sufficient conditions for $\boldsymbol{x}^*$ to be Pareto optimal in a similar way.

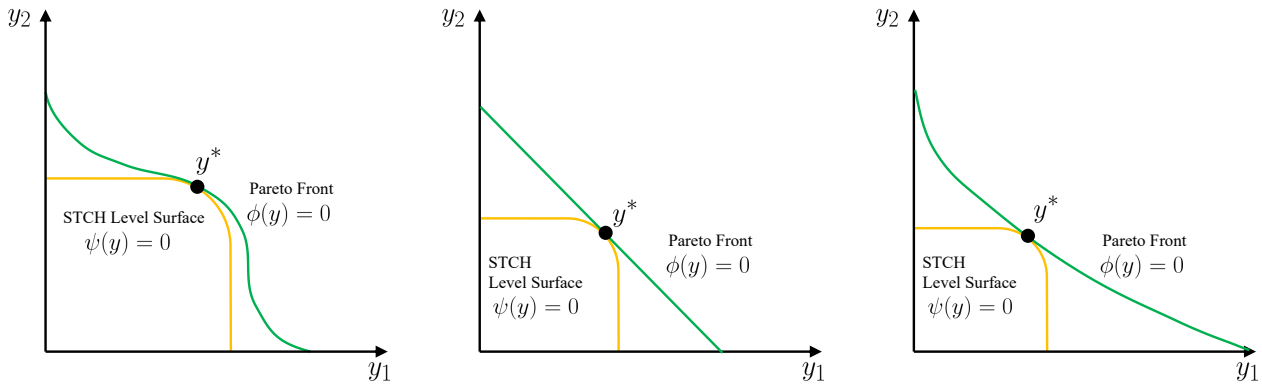
1. All Positive Preference Coefficients: Suppose that all preference coefficients $\boldsymbol{\lambda}_i$ are positive. If $\boldsymbol{x}^*$ is *not* Pareto optimal, there exists a valid solution $\hat{\boldsymbol{x}} \in \mathcal{X}$ such that $\boldsymbol{f}(\hat{\boldsymbol{x}}) \prec \boldsymbol{f}(\boldsymbol{x}^*)$. In other words, we have:

$$f_i(\hat{\boldsymbol{x}}) \leq f_i(\boldsymbol{x}^*), \forall i \in \{1, ..., m\} \text{ and } f_j(\hat{\boldsymbol{x}}) < f_j(\boldsymbol{x}^*), \exists j \in \{1, ..., m\}. \tag{15}$$

Based on the above inequalities and the condition $\boldsymbol{\lambda}_i > 0\ \forall i \in \{1, ..., m\}$, it is easy to see $\mu \log\left(\sum_{i=1}^m e^{\lambda_i(f_i(\hat{\boldsymbol{x}})-z_i^*)/\mu}\right) < \mu \log\left(\sum_{i=1}^m e^{\lambda_i(f_i(\boldsymbol{x}^*)-z_i^*)/\mu}\right)$ which contradicts the STCH optimality of $\boldsymbol{x}^*$ in (12). Therefore, $\boldsymbol{x}^*$ should be Pareto optimal for the original multi-objective optimization problem.

2. Uniqueness of the Solution: Suppose $\boldsymbol{x}^*$ is a unique optimal solution for the STCH scalarization. If $\boldsymbol{x}^*$ is *not* Pareto optimal, there exists a valid solution $\hat{\boldsymbol{x}} \in \mathcal{X}$ that satisfies the inequalities in (15). It is easy to check (it is different from (14)):

$$\mu \log\left(\sum_{i=1}^m e^{\lambda_i(f_i(\hat{\boldsymbol{x}})-z_i^*)/\mu}\right) \leq \mu \log\left(\sum_{i=1}^m e^{\lambda_i(f_i(\boldsymbol{x}^*)-z_i^*)/\mu}\right). \tag{16}$$



(a) General Pareto Front (b) Linear Pareto Front (c) Convex Pareto Front

Figure 5. The level surfaces of smooth Tchebycheff (STCH) scalarization and different Pareto fronts.

On the contrary, the condition of $\boldsymbol{x}^*$ being a unique optimal solution for STCH scalarization indicates that

$$\mu \log \left(\sum_{i=1}^{m} e^{\lambda_i (f_i(\hat{\boldsymbol{x}}) - z_i^*)/\mu} \right) > \mu \log \left(\sum_{i=1}^{m} e^{\lambda_i (f_i(\boldsymbol{x}^*) - z_i^*)/\mu} \right). \tag{17}$$

These two inequalities above are contradictory, and therefore the solution $\boldsymbol{x}^*$ should be Pareto optimal.

□

A.2. Proof of Theorem 3.8

Following the work on convexification of the Pareto front (Li, 1996; Goh & Yang, 1998), we prove this theorem by analyzing the relationship between the level surface of the STCH scalarization and the Pareto front as illustrated in Figure 5.

First, we express the Pareto front for a multi-objective optimization problem as the surface $\phi(\boldsymbol{y}) = 0$ where $\boldsymbol{y} \in \mathbb{R}^m$ is a vector in the objective space. Then we make the following assumption following the previous work:

Assumption A.1 (Li (1996))**.** For each point $\boldsymbol{y}$ on the Pareto front, we assume:

1. without loss of generality, $\nabla\phi(\boldsymbol{y}) \succ 0$, which means $\frac{\partial \phi(\boldsymbol{y})}{\partial y_i} > 0$ for all $i = 1, \ldots, m$; and
2. all second-order derivatives of $\phi(\boldsymbol{y})$ are bounded.

For the first assumption, we can easily reverse the sign of $\phi(\boldsymbol{y})$ if $\nabla\phi(\boldsymbol{y}) \prec 0$. This assumption also indicates that $\boldsymbol{f}(\boldsymbol{X}_w^*) \setminus \boldsymbol{f}(\boldsymbol{X}^*) = \emptyset$ which means no solution can be weakly Pareto optimal but not Pareto optimal in this problem. The second assumption means that the eigenvalues of the Hessian $\nabla^2\phi(\boldsymbol{y})$ are finite. With these two assumptions, we are ready to give a proof for the theorem.

Theorem 3.8 (Ability to Find All Pareto Solutions). *Under mild conditions, there exists a μ^* such that, for any $0 < \mu < \mu^*$, every Pareto solution of the original multi-objective optimization problem (1) is an optimal solution of the STCH scalarization problem (9) with some valid preference $\boldsymbol{\lambda}$.*

Proof. We denote the level surface of the STCH scalarization by $\psi(\boldsymbol{y})$ with:

$$\psi(\boldsymbol{y}) = \mu \log \left(\sum_{i=1}^{m} e^{\lambda_i (y_i - z_i^*)/\mu} \right) - c = 0. \tag{18}$$

For *any* valid Pareto solution $\boldsymbol{y}$ on the Pareto front, suppose that the STCH level surface $\psi(\boldsymbol{y}) = 0$ with some preference $\boldsymbol{\lambda}$ is tangential to the Pareto front $\phi(\boldsymbol{y}) = 0$. To prove this theorem, it is equivalent to show $\boldsymbol{y}$ must be an optimal solution of the STCH scalarization problem (9). In other words, the STCH level surface must be enclosed by the Pareto front toward the

origin. Since these two surfaces are tangential at $\boldsymbol{y}$, the curvature of the STCH level surface $\psi(\boldsymbol{y}) = 0$ should be larger than the curvature of the Pareto front $\phi(\boldsymbol{y}) = 0$. Some illustrations can be found in Figure 5. In the following, we give a sufficient condition for this case with a small enough μ^*.

For better exposition and without loss of generality, we further assume that the level surface of STCH scalarization with equal preference $\boldsymbol{\lambda} = (1, 1, \ldots, 1)$ and ideal point $\boldsymbol{z}^* = 0$ is tangential to the Pareto front at some point $\boldsymbol{y}^*$. This case can be easily generalized to all valid preference $\boldsymbol{\lambda}$ and ideal point $\boldsymbol{z}^*$.

Since the two surfaces $\psi(\boldsymbol{y}) = 0$ and $\phi(\boldsymbol{y}) = 0$ are tangential at $\boldsymbol{y}^*$, we have:

$$\psi(\boldsymbol{y}^*) = \phi(\boldsymbol{y}^*), \quad \nabla\psi(\boldsymbol{y}^*) = \nabla\phi(\boldsymbol{y}^*). \tag{19}$$

According to Assumption A.1, the Pareto front $\phi(\boldsymbol{y})$ is a smooth function with respect to $\boldsymbol{y}$. It is also easy to check that the STCH level surface $\psi(\boldsymbol{y})$ is also smooth. Therefore, we can expand both $\phi(\boldsymbol{y})$ and $\psi(\boldsymbol{y})$ near $\boldsymbol{y}^*$ with their Taylor series up to the second term:

$$\phi(\boldsymbol{y}) = \phi(\boldsymbol{y}^*) + \nabla\phi(\boldsymbol{y}^*)(\boldsymbol{y} - \boldsymbol{y}^*) + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{y}^*)^T\nabla^2\phi(\boldsymbol{y}^*)(\boldsymbol{y} - \boldsymbol{y}^*) + o(||\boldsymbol{y} - \boldsymbol{y}^*||^2), \tag{20}$$

$$\psi(\boldsymbol{y}) = \psi(\boldsymbol{y}^*) + \nabla\psi(\boldsymbol{y}^*)(\boldsymbol{y} - \boldsymbol{y}^*) + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{y}^*)^T\nabla^2\psi(\boldsymbol{y}^*)(\boldsymbol{y} - \boldsymbol{y}^*) + o(||\boldsymbol{y} - \boldsymbol{y}^*||^2). \tag{21}$$

Taking them together with (19), we have:

$$\psi(\boldsymbol{y}) - \phi(\boldsymbol{y}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{y}^*)^T(\nabla^2\psi(\boldsymbol{y}^*) - \nabla^2\phi(\boldsymbol{y}^*))(\boldsymbol{y} - \boldsymbol{y}^*) + o(||\boldsymbol{y} - \boldsymbol{y}^*||^2). \tag{22}$$

The STCH level surface $\psi(\boldsymbol{y}) = 0$ is enclosed by the Pareto front $\phi(\boldsymbol{y}) = 0$ towards the origin if $\psi(\boldsymbol{y}) - \phi(\boldsymbol{y}) \geq 0$ for all $\boldsymbol{y}$ in the neighborhood of $\boldsymbol{y}^*$. According to Assumption A.1, the term $\nabla^2\phi(\boldsymbol{y}^*)$ is bounded. Therefore, it is important to analyze the term $\nabla^2\psi(\boldsymbol{y}^*) = \nabla^2 g_\mu^{(\text{STCH})}(\boldsymbol{y}^*)$ which is the Hessian matrix of STCH scalarization with respect to the objective value $\boldsymbol{y}$.

For STCH scalarization with a equal preference and ideal point at $\mathbf{0}$, we have

$$\nabla_{\boldsymbol{y}} g_\mu^{(\text{STCH})} = \nabla_{\boldsymbol{y}}\mu\log\left(\sum_{i=1}^m e^{y_i/\mu}\right) = \frac{e^{\boldsymbol{y}/\mu}}{\sum_i e^{y_i/\mu}} \in \mathbb{R}^m. \tag{23}$$

It is easy to show the STCH scalarization is twice differentiable with

$$\nabla^2_{\boldsymbol{y}} g_\mu^{(\text{STCH})} = \frac{1}{\mu}\left(\frac{1}{\mathbf{1}^T\boldsymbol{z}}\operatorname{diag}(\boldsymbol{z}) - \frac{1}{(\mathbf{1}^T\boldsymbol{z})^2}\boldsymbol{z}\boldsymbol{z}^T\right), \text{where } \boldsymbol{z} = e^{\boldsymbol{y}}. \tag{24}$$

For any $\boldsymbol{v} \in \mathbb{R}^m$, we have

$$\boldsymbol{v}^T\nabla^2_{\boldsymbol{y}} g_\mu^{(\text{STCH})}\boldsymbol{v} = \boldsymbol{v}^T\left[\frac{1}{\mu}\left(\frac{1}{\mathbf{1}^T\boldsymbol{z}}\operatorname{diag}(\boldsymbol{z}) - \frac{1}{(\mathbf{1}^T\boldsymbol{z})^2}\boldsymbol{z}\boldsymbol{z}^T\right)\right]\boldsymbol{v} \tag{25}$$

$$= \frac{(\sum_{i=1}^m \boldsymbol{z}_i\boldsymbol{v}_i^2)(\sum_{i=1}^m \boldsymbol{z}_i) - (\sum_{i=1}^m \boldsymbol{v}_i\boldsymbol{z}_i)^2}{\mu(\sum_{i=1}^m \boldsymbol{z}_i)^2}. \tag{26}$$

According to the Cauchy-Schwarz inequality, we have $(\sum_{i=1}^m \boldsymbol{v}_i\boldsymbol{z}_i)^2 \leq (\sum_{i=1}^m \boldsymbol{z}_i\boldsymbol{v}_i^2)(\sum_{i=1}^m \boldsymbol{z}_i)$ and the two sides are equal if and only if $\boldsymbol{v} = \gamma\boldsymbol{I}_m$ with any $\gamma \in \mathbb{R}$. It means $\boldsymbol{v}^T\nabla^2_{\boldsymbol{y}} g_\mu^{(\text{STCH})}\boldsymbol{v} \geq 0$ for all $\boldsymbol{v} \in \mathbb{R}^m$. In other words, $\nabla^2_{\boldsymbol{y}} g_\mu^{(\text{STCH})}$ is positive semi-definite ($\nabla^2_{\boldsymbol{y}} g_\mu^{(\text{STCH})} \succeq 0$). Now we are ready to analyze the cases for different Pareto fronts.

Convex Pareto Front If the Pareto set is convex (here it means cone-convex), all the eigenvalues of its Hessian $\nabla^2\phi(\boldsymbol{y})$ are negative for any valid $\boldsymbol{y}$. Since $\nabla^2\psi(\boldsymbol{y}^*)$ is positive semi-definite, it is easy to check $\psi(\boldsymbol{y}) - \phi(\boldsymbol{y}) > 0$ for every $\boldsymbol{y}$, which is agnostic to the smoothing parameter μ. Therefore, for any $\mu > 0$, every Pareto solution of the original multi-objective optimization problem (1) is an optimal solution of the STCH scalarization problem (9) with some valid preference $\boldsymbol{\lambda}$ (Corollary 3.9).

General Pareto Front Let $\boldsymbol{v} = (\boldsymbol{y} - \boldsymbol{y}^*)$, we can combine (22) and (26) to have

$$\psi(\boldsymbol{y}) - \phi(\boldsymbol{y}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{y}^*)^T(\nabla^2\psi(\boldsymbol{y}^*) - \nabla^2\phi(\boldsymbol{y}^*))(\boldsymbol{y} - \boldsymbol{y}^*) + o(||\boldsymbol{y} - \boldsymbol{y}^*||^2) \tag{27}$$

$$= \frac{(\sum_{i=1}^m \boldsymbol{z}_i\boldsymbol{v}_i^2)(\sum_{i=1}^m \boldsymbol{z}_i) - (\sum_{i=1}^m \boldsymbol{v}_i\boldsymbol{z}_i)^2}{2\mu(\sum_{i=1}^m \boldsymbol{z}_i)^2} - \frac{1}{2}\boldsymbol{v}^T\nabla^2\phi(\boldsymbol{y}^*)\boldsymbol{v} + o(||\boldsymbol{y} - \boldsymbol{y}^*||^2) \tag{28}$$

$$\geq \frac{(\sum_{i=1}^m \boldsymbol{z}_i\boldsymbol{v}_i^2)(\sum_{i=1}^m \boldsymbol{z}_i) - (\sum_{i=1}^m \boldsymbol{v}_i\boldsymbol{z}_i)^2}{2\mu(\sum_{i=1}^m \boldsymbol{z}_i)^2} - K + o(||\boldsymbol{y} - \boldsymbol{y}^*||^2) \tag{29}$$

where K is some constant since $\nabla^2\phi(\boldsymbol{y}^*)$ is bounded according to Assumption A.1. According to (26) and the Cauchy-Schwarz inequality, if $\boldsymbol{v} = (\boldsymbol{y} - \boldsymbol{y}^*) \neq \gamma\boldsymbol{I}_m$ with any $\gamma \in \mathbb{R}$, the first term of (29) is always positive. Therefore, there exists a small enough μ^* such that

$$\psi(\boldsymbol{y}) - \phi(\boldsymbol{y}) > 0, \forall 0 < \mu < \mu^*. \tag{30}$$

Furthermore, since $\boldsymbol{y}^*$ is the optimal Pareto solution with a equal preference $(1, 1, \ldots, 1)$, we have $y_i^* = y_j^* \forall 0 \leq i, j \leq m$. If $\boldsymbol{v} = (\boldsymbol{y} - \boldsymbol{y}^*) = \gamma\boldsymbol{I}_m$, the point $\boldsymbol{y}$ should also satisfy $y_i = y_j \forall 0 \leq i, j \leq m$. However, according to the definition of Pareto front and the property of STCH scalarization, if $\boldsymbol{y} \neq \boldsymbol{y}^*$, $\boldsymbol{y}$ cannot be on the Pareto front or the STCH level surface. In other words, such $\boldsymbol{y}$ is not a valid neighbor point on the surface $\psi(\boldsymbol{y})$ or $\phi(\boldsymbol{y})$.

Therefore, this proof is completed. □

Corollary 3.9. *If the Pareto front is convex, then for any μ, every Pareto solution of the original multi-objective optimization problem (1) is an optimal solution of the STCH scalarization problem (9) with some valid preference $\boldsymbol{\lambda}$.*

A.3. Proof of Theorem 3.10

Theorem 3.10 (Convergence to Pareto Stationary Solution). *If there exists a solution $\hat{\boldsymbol{x}}$ such that $\nabla g_\mu^{(STCH)}(\hat{\boldsymbol{x}}|\boldsymbol{\lambda}) = 0$, then $\hat{\boldsymbol{x}}$ is a Pareto stationary solution of the original multi-objective optimization problem (1).*

Proof. Let $\boldsymbol{y} = \boldsymbol{\lambda}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{z}^*)$ be an m-dimensional vector, we can rewrite the STCH scalarization as:

$$g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \mu \log\left(\sum_{i=1}^m e^{\lambda_i(f_i(\boldsymbol{x}) - z_i^*)/\mu}\right) = \mu \log\left(\sum_{i=1}^m e^{y_i/\mu}\right). \tag{31}$$

Then take the gradient of $g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$ with respect to $\boldsymbol{y}$, we have:

$$\nabla_{\boldsymbol{y}} g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \nabla_{\boldsymbol{y}} \mu \log\left(\sum_{i=1}^m e^{y_i/\mu}\right) = \frac{e^{\boldsymbol{y}/\mu}}{\sum_i e^{y_i/\mu}} \in \mathbb{R}^m, \tag{32}$$

Therefore, according to the chain rule, the gradient of $g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda})$ with respect to the solution $\boldsymbol{x}$ is:

$$\nabla_{\boldsymbol{x}} g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \nabla_{\boldsymbol{y}} g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) \cdot \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$$

$$= \sum_{i=1}^m \frac{e^{y_i/\mu}}{\sum_i e^{y_i/\mu}} \nabla\lambda_i(f_i(\boldsymbol{x}) - z_i^*) \tag{33}$$

$$= \sum_{i=1}^m \frac{\lambda_i e^{y_i/\mu}}{\sum_i e^{y_i/\mu}} \nabla f_i(\boldsymbol{x}). \tag{34}$$

It is easy to check $w_i = \frac{\lambda_i e^{y_i/\mu}}{\sum_i e^{y_i/\mu}} \geq 0, \forall i$. Therefore, let $s = \sum_{i=1}^m w_i$ and $\bar{w}_i = w_i/s$, we have:

$$\nabla_{\boldsymbol{x}} g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \sum_{i=1}^m w_i \nabla f_i(\boldsymbol{x}) = s\sum_{i=1}^m \bar{w}_i \nabla f_i(\boldsymbol{x}), \tag{35}$$

where $s > 0$ and $\hat{\boldsymbol{w}} \in \boldsymbol{\Delta}^{m-1} = \{\hat{\boldsymbol{w}} | \sum_{i=1}^{m} \bar{w}_i = 1, \bar{w}_i \geq 0 \; \forall i\}$. For a solution $\hat{\boldsymbol{x}}$, if $\nabla g_{\mu}^{(\text{STCH})}(\hat{\boldsymbol{x}}|\boldsymbol{\lambda}) = 0$, we have:

$$\nabla_{\boldsymbol{x}} g_{\mu}^{(\text{STCH})}(\hat{\boldsymbol{x}}|\boldsymbol{\lambda}) = s \sum_{i=1}^{m} \bar{w}_i \nabla f_i(\hat{\boldsymbol{x}}) = 0 \tag{36}$$

$$\Longrightarrow_{\text{divided by } s} \sum_{i=1}^{m} \bar{w}_i \nabla f_i(\hat{\boldsymbol{x}}) = 0.$$

According to Definition 2.4 of Pareto stationarity, the solution $\hat{\boldsymbol{x}}$ is a Pareto stationary solution. □

B. Practical Implementation for STCH Scalarization

B.1. Computational Stability

For the STCH scalarization, we have

$$g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \mu \log\left(\sum_{i=1}^m e^{\frac{y_i}{\mu}}\right) \tag{37}$$

$$\nabla_{\boldsymbol{x}} g_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \sum_{i=1}^m \frac{\lambda_i e^{y_i/\mu}}{\sum_i e^{y_i/\mu}} \nabla f_i(\boldsymbol{x}), \tag{38}$$

where $\boldsymbol{y}_i = \boldsymbol{\lambda}_i(\boldsymbol{f}_i(\boldsymbol{x}) - z_i^*)$. These two terms might both have computational stability issues with a very small smoothing parameter μ.

Following the stable technique proposed in Nesterov (2005), we first calculate $\tilde{y} = \max_{\forall i} y_i$, and then set $\hat{y}_i = y_i - \tilde{y}$ for all i. Then we have a stabalized version of the STCH scalarization and its gradient:

$$\hat{g}_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \mu \log\left(\sum_{i=1}^m e^{\hat{y}_i/\mu}\right) \tag{39}$$

$$\nabla_{\boldsymbol{x}} \hat{g}_\mu^{(\text{STCH})}(\boldsymbol{x}|\boldsymbol{\lambda}) = \sum_{i=1}^m \frac{\lambda_i e^{\hat{y}_i/\mu}}{\sum_i e^{\hat{y}_i/\mu}} \nabla f_i(\boldsymbol{x}). \tag{40}$$

B.2. Objective Normalization

In many real-world application problems, the objective functions could be in very different scales. It could be very difficult for the user to assign a meaningful preference among the objectives. In this case, we can first normalize each objective

$$\hat{f}_i = \frac{f_i - f_{\text{i, min}}}{f_{\text{i, max}} - f_{\text{i, min}}} \tag{41}$$

with known or predicted lower bounds and upper bounds, and then apply STCH scalarization on the normalized objective $\hat{\boldsymbol{f}}(\boldsymbol{x}) = (\hat{f}_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$. For all multi-task learning problems, following the setting from Dai et al. (2023), we set $f_{\text{i, max}}$ to be the training loss for the i-th task at the end of the second epoch and $f_{\text{i, min}} = 0$.

B.3. Setting the Smoothing Parameter μ

For the STCH scalarization, there is an remained challenge on how to set the smoothing parameter. According to Nesterov (2005) and Beck & Teboulle (2012), the smoothing parameter μ for smooth optimization should be set properly based on the properties of the original function (e.g., Lipschitz constant L), which is usually unknown. Some practical algorithms have been proposed to continually solve a smoothed surrogate sequence with a schedule of large to small smoothing parameters (Xu et al., 2016). However, this approach could lead to extra computational overhead. In this work, we simply use a fixed smoothing parameter in all experiments, and find that the fixed μ work pretty well.

B.4. Constrains Handling

This work mainly focuses on the unconstrained multiobjective optimization problem, while the constrained problem could also be important for many real-world problems. Extra approaches could be needed for STCH to handle the constraints.

A constrained multiobjective optimization problem can be defined as

$$\min_{x \in \Omega} \; f(x) = (f_1(x), f_2(x), \cdots, f_m(x)) \tag{42}$$

with $\Omega = \{x \in R^n : g_j(x) \leq 0, j = 1, \ldots, p, \quad h_l(x) = 0, l = 1, \ldots, q\}$, where $\{g_j(x)\}$ and $\{h_l(x)\}$ are the inequality and equality constraints. For differentiable multiobjective optimization, we assume that all objectives $\{f_i(x)\}$ and constraints $\{g_j(x)\}, \{h_l(x)\}$ are continuously differentiable.

Following previous work on gradient-based multiobjective optimization (Fliege & Vaz, 2016), we can reformulate the constrained multiobjective optimization problem into an unconstrained problem with additional objectives:

$$\min_{x \in R^n} (f(x), g^+(x), |h(x)|) = (f_1(x), \cdots, f_m(x), g_1^+(x), \cdots, g_p^+(x), |h_1(x)|, \cdots, |h_q(x)|) \tag{43}$$

where $g_j^+(x) = \max\{0, g_j(x)\}$. A feasible solution x for the original constrained problem satisfies $\{g_j^+(x) = 0, j = 1, \ldots, p, \quad |h_l(x)| = 0, l = 1, \ldots, q\}$. Instead of the original problem with m objectives and $p + q$ constraints, we now have an unconstrained problem with $m + p + q$ objectives and 0 constraints. With our proposed STCH scalarization, we have the following problem:

$$g_\mu^{(\text{STCH})}(x|\lambda, \sigma) = \mu \log \left(\sum_{i=1}^{m} e^{\frac{\lambda_i (f_i(x) - z_i^*)}{\mu}} + \sum_{j=1}^{p} e^{\frac{\sigma g_j^+(x)}{\mu}} + \sum_{l=1}^{q} e^{\frac{\sigma |h_l(x)|}{\mu}} \right), \tag{44}$$

where $\sigma \geq 0$ is a penalty parameter. We can also use the smooth counterparts for $g_j^+(x)$ and $|h_l(x)|$ in the STCH scalarization above to achieve a better convergence rate. In addition to this soft constraint handling approach, there could be many other possible constraint handling methods that work for STCH, which is an important research direction in the future.

B.5. Potential Extension for Stochastic Multi-Objective Optimization

In this work, we mainly focus on deterministic multi-objective optimization setting, and leave the study for stochastic optimization as an important future work. We would like to make the following remarks on this concern:

- Stochastic multi-objective optimization itself is an important research topic that was not yet well explored in the past decades. Most existing methods only study the deterministic setting. Until recently, a few work have been proposed to investigate the stochastic adaptive gradient method (Liu & Vicente, 2021; Zhou et al., 2022; Fernando et al., 2023). In this work, instead of yet another adaptive gradient method, we focus on proposing a lightweight, efficient, and theoretically solid scalarization method.
- Although developed from the deterministic setting, our proposed STCH scalarization method can outperform MOCO (Fernando et al., 2023), a state-of-the-art stochastic adaptive gradient method on all multi-task learning datasets (for performance and run time), which are essentially stochastic optimization problems. These results validate that our proposed method can be used to tackle stochastic optimization problems.
- In addition, how to design an efficient algorithm for stochastic nonsmooth optimization problem is also an emerging research topic. A recent work (Wang & Zhang, 2023) investigates efficient accelerated gradient methods for the classic smoothing method proposed in Nesterov (2005) and Beck & Teboulle (2012). We might leverage this recent development to investigate efficient scalarization methods for stochastic multi-objective optimization.

C. Discussion with Related Work on Minimax Optimization

The Tchebycheff scalarization approach is naturally a minimax optimization problem. In this section, we briefly discuss the relation between our proposed STCH method and the related algorithms for minimax optimization.

Problem Formulation A general minimax optimization problem is in the form:

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} f(x,y),$$

where $\mathcal{X} \in R^n$ and $\mathcal{Y} \in R^m$ are both closed and bounded convex sets, while $f : \mathcal{X} \times \mathcal{Y} \to R$ is a smooth function. From the viewpoint of Tchebycheff scalarization, we are more interested in the problem with a specific linear structure on y:

$$\min_{x\in\mathcal{X}} \max_{y\in\Delta^{m-1}} F(x)^T y,$$

where $F(x) = (f_1(x), f_2(x), \cdots, f_m(x))^T$ and $\Delta^{m-1} = \{y | \sum_{i=1}^m y_i = 1, y_i \geq 0\ \forall i\}$ is a probability simplex. It is easy to check that the optimal solution of the above problem is also optimal for the Tchebycheff scalarization problem $\min_{x\in\mathcal{X}} \max_{1\leq i\leq m} f_i(x)$ since the linear optimization problem over the simplex is always optimized at one of its vertices (Boyd & Vandenberghe, 2004). Here we drop the preference λ and reference point z^* for better exposition.

Algorithms for Minimax Optimization There are many algorithms to solve the minimax optimization problems, such as some classic methods (Nemirovski, 2004; Nesterov, 2007; Monteiro & Svaiter, 2010). In recent years, different algorithms have been proposed to tackle the minimax problems in different settings (Palaniappan & Bach, 2016; Gidel et al., 2017; Mertikopoulos et al., 2019). Rather than those algorithms with a double-loop or even triple-loop structure, we are more interested in the lightweight single-loop algorithm for minimax optimization such as Gradient Descent Ascent (GDA) (Lin et al., 2020a). However, the simple GDA algorithm could oscillate during the optimization process (Mokhtari et al., 2020) and fail to converge even for a simple bilinear problem $\min_{x\in R^n} \max_{y\in R^n} x^T y$ (Letcher et al., 2019). Extra efforts, such as alternating gradient projection (Xu et al., 2023) or a smoothing method (Zhang et al., 2020), are needed for GDA.

Comparison and Discussion The smoothed gradient descent ascent (smoothed-GDA) (Zhang et al., 2020) algorithm is the most related minmax optimization algorithm to our proposed STCH method. The key approach in smoothed-GDA is to add an extra smoothed regularization term to the objective function:

$$K(x, z; y) = f(x, y) + \frac{p}{2}||x - z||^2. \tag{45}$$

To optimize the above function, in addition to the original x and y, smoothed-GDA maintains and updates an auxiliary sequence $\{z^t\}$ where $z^{t+1} = z^t + \beta(x^{t+1} - z^t)$ throughout the optimization process. The terms $p > 0$ and $\beta > 0$ are hyperparameters, and the smoothed-GDA reduces to the standard GDA when $\beta = 1$.

Compared to smoothed-GDA, our proposed STCH approach does not require maintaining an extra sequence $\{z^t\}$ which could lead to high memory cost (e.g., an additional deep neural network with millions of parameters). For Pareto set learning $x^*(\lambda) = h_\theta(\lambda) = argmin_{x\in\mathcal{X}} g(x|\lambda) \forall \lambda \in \Delta^{m-1}$, we need to randomly sample different trade-off preferences $\lambda \in \Delta^{m-1}$ (and hence different single-objective subproblems) at each iteration. In this case, it could be very hard to maintain extra sequences for each subproblem, since the trade-offs could be infinite and come in an online manner. Our proposed STCH method is simple, lightweight, and with good theoretical properties, and we hope that it can inspire more follow-up work on developing more efficient gradient-based methods (e.g., improved minmax optimization algorithms) for multi-objective optimization.

	μ	α	ϵ_{HOMO}	ϵ_{LUMO}	$< R^2 >$	ZPVE	U_0	U	H	G	c_v	$\Delta_p \uparrow$
STL	0.062	0.192	58.82	51.95	0.529	4.52	63.69	60.83	68.33	60.31	0.069	0.00
TCH	0.266	0.401	107.1	151.6	5.922	13.2	166.7	167.5	168.1	162.0	0.206	-252.2
Smoothed-GDA	0.252	0.424	105.2	157.1	4.561	11.8	162.4	159.3	149.9	152.2	0.198	-218.8
STCH (Ours)	0.166	0.260	94.48	101.2	1.850	4.88	58.34	58.68	58.70	58.27	0.104	-58.14

We also implemented and tested smoothed-GDA on the QM9 dataset. According to the results above, smoothed-GDA achieves better overall performance than the original TCH scalarization, but is significantly outperformed by our proposed STCH scalarization method. We have also tried to tune the hyperparameters for smoothed-GDA, but do not obtain any significant improvement.

D. Detailed Experiment Setting

D.1. Multi-Task Learning Problems

In this work, we implement our STCH scalarization method for MTL problems with the LibMTL library (Lin & Zhang, 2023), and follow the experiment settings of Lin et al. (2023).

NYUv2 (Silberman et al., 2012) is an indoor scene understanding dataset with 3 tasks on semantic segmentation, depth estimation, and surface normal prediction, with 795 training and 654 testing images. Following Lin et al. (2023), we compare all MTL methods using the SegNet model with a shared encoder and 3 task-specific decoders. The model is trained for 200 epochs with Adam (Kingma & Ba, 2015), of which the learning rate is initially set to 10^{-4} with 10^{-5} weight decay and will be halved to 5×10^{-5} after 100 epochs. The batch size is set to 2. The three optimization objectives are cross-entropy loss, L_1 loss and consine loss for semantic segmentation, depth estimation, and surface normal prediction.

Office-31 (Saenko et al., 2010) is an image classification dataset that contains $4,110$ images across 3 domains (Amazon, DSLR, and Webcam), of which each task has 31 classes. The data split from (Lin et al., 2022a) is utilized to split the data as 60%-20%-20% for training, validation, and testing. All MTL methods are compared on the same model with a shared ResNet-18 pretrained encoder and linear task-specific head for each task. All methods are trained by Adam (Kingma & Ba, 2015), of which the learning rate is 10^{-4} with 10^{-5} weight decay. The batch size is 64 and the training epoch is 100. The optimization objectives are three cross-entropy losses for each classification task.

QM9 (Ramakrishnan et al., 2014) is a molecular property prediction dataset with 11 tasks on different properties. The data split in Navon et al. (2022) is used to divide the dataset into $110,000$ for training, $10,000$ for validation, and $10,000$ for testing. All MTL methods are compared on the same model with a shared graph neural network encoder and 11 linear task-specific head as in Navon et al. (2022) and Lin et al. (2023). All methods are trained by Adam (Kingma & Ba, 2015), of which the learning rate is 10^{-3} with the ReduceLROnPlateau scheduler. The batch size is 128 and the number of training epochs is 300. The optimization objectives are 11 mean squared error (MSE) for each task.

D.2. Pareto Set Learning

Following other related works on Pareto set learning, we build a simple fully-connected multi-layered perceptron (MLP) as the Pareto set model:

$$\begin{aligned} h_{\boldsymbol{\theta}}(\boldsymbol{\lambda}) :&\text{Input } (\boldsymbol{\lambda}) \to \text{Linear}(m, 256) \to \text{ReLU} \to \text{Linear}(256, 256) \\ &\to \text{ReLU} \to \text{Linear}(256, 256) \to \text{ReLU} \\ &\to \text{Linear}(256, n) \to \text{Output } \boldsymbol{x}(\boldsymbol{\lambda}), \end{aligned} \tag{46}$$

where the input is the preference $\boldsymbol{\lambda} \in \Delta^{m-1}$ with size m. This model is a two-layer MLP that has 256 units in each hidden layer with ReLU activation, and the output is a solution $\boldsymbol{x}(\boldsymbol{\lambda}) \in \mathbb{R}^n$. For Pareto set learning, the goal is to find the optimal model parameter $\boldsymbol{\theta}^*$ such that $\boldsymbol{x}^*(\boldsymbol{\lambda}) = h_{\boldsymbol{\theta}^*}(\boldsymbol{\lambda})$ is the corresponding optimal solution for any given preference $\boldsymbol{\lambda}$. In all experiments, we train each set model with $2,000$ iterations, of which 10 different preferences are uniformly sampled from Δ^{m-1} at each iteration. In other words, the total evaluation budget is $20,000$ for each method.

D.2.1. Synthetic Benchmark Problems

We first compare different scalarization methods for learning the Pareto set on 6 synthetic benchmark problems shown on the next page. The problems F1-F3 have convex Pareto fronts, and the rest problems F4-F6 have concave Pareto fronts.

F1	$f_1(\boldsymbol{x}) = (1+\frac{s_1}{\lvert J_1\rvert})\boldsymbol{x}_1, \quad f_2(\boldsymbol{x}) = (1+\frac{s_2}{\lvert J_2\rvert})\left(1-\sqrt{\frac{\boldsymbol{x}_1}{1+\frac{s_2}{\lvert J_2\rvert}}}\right)$ where $s_1 = \sum_{j\in J_1}(\boldsymbol{x}_j-(2\boldsymbol{x}_1-1)^2)^2$ and $s_2 = \sum_{j\in J_2}(\boldsymbol{x}_j-(2\boldsymbol{x}_1-1)^2)^2$, $J_1 = \{j \mid j \text{ is odd and } 2\le j\le n\}$ and $J_2 = \{j \mid j \text{ is even and } 2\le j\le n\}$ (47)
F2	$f_1(\boldsymbol{x}) = (1+\frac{s_1}{\lvert J_1\rvert})\boldsymbol{x}_1, \quad f_2(\boldsymbol{x}) = (1+\frac{s_2}{\lvert J_2\rvert})\left(1-\sqrt{\frac{\boldsymbol{x}_1}{1+\frac{s_2}{\lvert J_2\rvert}}}\right)$ where $s_1 = \sum_{j\in J_1}(\boldsymbol{x}_j-\boldsymbol{x}_1^{0.5(1.0+\frac{3(j-2)}{n-2})})^2$ and $s_2 = \sum_{j\in J_2}(\boldsymbol{x}_j-\boldsymbol{x}_1^{0.5(1.0+\frac{3(j-2)}{n-2})})^2$, $J_1 = \{j \mid j \text{ is odd and } 2\le j\le n\}$ and $J_2 = \{j \mid j \text{ is even and } 2\le j\le n\}$ (48)
F3	$f_1(\boldsymbol{x}) = (1+\frac{s_1}{\lvert J_1\rvert})\boldsymbol{x}_1, \quad f_2(\boldsymbol{x}) = (1+\frac{s_2}{\lvert J_2\rvert})\left(1-\sqrt{\frac{\boldsymbol{x}_1}{1+\frac{s_2}{\lvert J_2\rvert}}}\right)$ where $s_1 = \sum_{j\in J_1}(\boldsymbol{x}_j-\sin(4\pi\boldsymbol{x}_1+\frac{j\pi}{n}))^2$ and $s_2 = \sum_{j\in J_2}(\boldsymbol{x}_j-\sin(4\pi\boldsymbol{x}_1+\frac{j\pi}{n}))^2$, $J_1 = \{j \mid j \text{ is odd and } 2\le j\le n\}$ and $J_2 = \{j \mid j \text{ is even and } 2\le j\le n\}$ (49)
F4	$f_1(\boldsymbol{x}) = (1+\frac{s_1}{\lvert J_1\rvert})\boldsymbol{x}_1, \quad f_2(\boldsymbol{x}) = (1+\frac{s_2}{\lvert J_2\rvert})\left(1-\left(\frac{\boldsymbol{x}_1}{1+\frac{s_2}{\lvert J_2\rvert}}\right)^2\right)$ where $s_1 = \sum_{j\in J_1}(\boldsymbol{x}_j-(2\boldsymbol{x}_1-1)^2)^2$ and $s_2 = \sum_{j\in J_2}(\boldsymbol{x}_j-(2\boldsymbol{x}_1-1)^2)^2$, $J_1 = \{j \mid j \text{ is odd and } 2\le j\le n\}$ and $J_2 = \{j \mid j \text{ is even and } 2\le j\le n\}$ (50)
F5	$f_1(\boldsymbol{x}) = (1+\frac{s_1}{\lvert J_1\rvert})\boldsymbol{x}_1, \quad f_2(\boldsymbol{x}) = (1+\frac{s_2}{\lvert J_2\rvert})\left(1-\left(\frac{\boldsymbol{x}_1}{1+\frac{s_2}{\lvert J_2\rvert}}\right)^2\right)$ where $s_1 = \sum_{j\in J_1}(\boldsymbol{x}_j-\boldsymbol{x}_1^{0.5(1.0+\frac{3(j-2)}{n-2})})^2$ and $s_2 = \sum_{j\in J_2}(\boldsymbol{x}_j-\boldsymbol{x}_1^{0.5(1.0+\frac{3(j-2)}{n-2})})^2$, $J_1 = \{j \mid j \text{ is odd and } 2\le j\le n\}$ and $J_2 = \{j \mid j \text{ is even and } 2\le j\le n\}$ (51)
F6	$f_1(\boldsymbol{x}) = (1+\frac{s_1}{\lvert J_1\rvert})\boldsymbol{x}_1, \quad f_2(\boldsymbol{x}) = (1+\frac{s_2}{\lvert J_2\rvert})\left(1-\left(\frac{\boldsymbol{x}_1}{1+\frac{s_2}{\lvert J_2\rvert}}\right)^2\right)$ where $s_1 = \sum_{j\in J_1}(\boldsymbol{x}_j-\sin(4\pi\boldsymbol{x}_1+\frac{j\pi}{n}))^2$ and $s_2 = \sum_{j\in J_2}(\boldsymbol{x}_j-\sin(4\pi\boldsymbol{x}_1+\frac{j\pi}{n}))^2$, $J_1 = \{j \mid j \text{ is odd and } 2\le j\le n\}$ and $J_2 = \{j \mid j \text{ is even and } 2\le j\le n\}$ (52)

D.2.2. Real-World Engineering Design Problems

In addition to synthetic benchmark problems, we also compare the STCH scalarization with different methods on the following five real-world engineering design problems summarized in Tanabe & Ishibuchi (2020):

Four Bar Truss Design It is a multi-objective engineering design problem for a four-bar truss system proposed in Cheng & Li (1999):

$$\begin{aligned} f_1(\boldsymbol{x}) &= L(2\boldsymbol{x}_1 + \sqrt{2}\boldsymbol{x}_2 + \sqrt{\boldsymbol{x}_3} + \boldsymbol{x}_4) \\ f_2(\boldsymbol{x}) &= \frac{FL}{E}(\frac{2}{\boldsymbol{x}_1} + \frac{2\sqrt{2}}{\boldsymbol{x}_2} - \frac{2\sqrt{2}}{\boldsymbol{x}_3} + \frac{2}{\boldsymbol{x}_4}) \end{aligned} \tag{53}$$

where the first objective $f_1(\boldsymbol{x})$ is to minimize the structural volume and the second objective $f_2(\boldsymbol{x})$ is to minimize the joint displacement. The four decision variables $\boldsymbol{x}_1, \boldsymbol{x}_4 \in [a, 3a]$, $\boldsymbol{x}_2, \boldsymbol{x}_3 \in [\sqrt{2}a, 3a]$ are the lengths of four bars with $a = F/\sigma$, and we have $F = 10$, $E = 2 \times 10^5$, $L = 200$, and $\sigma = 10$.

Hatch Cover Design It is a multi-objective engineering design problem for a hatch cover proposed in Amir & Hasegawa (1989):

$$\begin{aligned} f_1(\boldsymbol{x}) &= \boldsymbol{x}_1 + 120\boldsymbol{x}_2 \\ f_2(\boldsymbol{x}) &= \sum_{1=1}^{4} \max\{-g_i(\boldsymbol{x}), 0\} \text{ where} \\ g_1(\boldsymbol{x}) &= 1.0 - \frac{\sigma_b}{\sigma_{b,\max}}, \\ g_2(\boldsymbol{x}) &= 1.0 - \frac{\tau}{\tau_{\max}}, \\ g_3(\boldsymbol{x}) &= 1.0 - \frac{\delta}{\delta_{\max}}, \\ g_4(\boldsymbol{x}) &= 1.0 - \frac{\sigma_b}{\sigma_k} \end{aligned} \tag{54}$$

where the first objective $f_1(\boldsymbol{x})$ is to minimize the weight of the hatch cover and the second objective $f_2(\boldsymbol{x})$ is the sum of four constraint violations $g_1(\boldsymbol{x})$, $g_2(\boldsymbol{x})$, $g_3(\boldsymbol{x})$ and $g_4(\boldsymbol{x})$. The two decision variables are the flange thickness $\boldsymbol{x}_1 \in [0.5, 4]$ and beam height $\boldsymbol{x}_2 \in [0.5, 50]$ of the batch cover. The parameters in the constraints are defined as follows: $\sigma_{b,\max} = 700\text{kg/cm}^2$, $\tau_{\max} = 450\text{kg/cm}$, $\delta_{\max} = 1.5\text{cm}$, $\sigma_k = E\boldsymbol{x}_1^2/100\text{kg/cm}^2$, $\sigma_b = 4500/(\boldsymbol{x}_1\boldsymbol{x}_2)\text{kg/cm}^2$, $\tau = 1800/\boldsymbol{x}_2\text{kg/cm}^2$, $\delta = 56.2 \times 10^4/(E\boldsymbol{x}_1\boldsymbol{x}_2^2)$, where $E = 700,000\text{kg/cm}^2$.

Disk Brake Design It is a multi-objective engineering design problem for a disk brake proposed in Ray & Liew (2002):

$$\begin{aligned} f_1(\boldsymbol{x}) &= 4.9 \times 10^{-5}(\boldsymbol{x}_2^2 - \boldsymbol{x}_1^2)(\boldsymbol{x}_4 - 1) \\ f_2(\boldsymbol{x}) &= 9.82 \times 10^6(\frac{\boldsymbol{x}_2^2 - \boldsymbol{x}_1^2}{\boldsymbol{x}_3\boldsymbol{x}_4(\boldsymbol{x}_2^3 - \boldsymbol{x}_1^3)}) \\ f_3(\boldsymbol{x}) &= \sum_{1=1}^{4} \max\{-g_i(\boldsymbol{x}), 0\} \text{ where} \\ g_1(\boldsymbol{x}) &= (\boldsymbol{x}_2 - \boldsymbol{x}_1) - 20, \\ g_2(\boldsymbol{x}) &= 0.4 - \frac{\boldsymbol{x}_3}{3.14(\boldsymbol{x}_2^2 - \boldsymbol{x}_1^2)}, \\ g_3(\boldsymbol{x}) &= 1 - \frac{2.22 \times 10^{-3}\boldsymbol{x}_3(\boldsymbol{x}_2^3 - \boldsymbol{x}_1^3)}{(\boldsymbol{x}_2^2 - \boldsymbol{x}_1^2)^2}, \\ g_4(\boldsymbol{x}) &= \frac{2.66 \times 10^{-2}\boldsymbol{x}_3\boldsymbol{x}_4(\boldsymbol{x}_2^3 - \boldsymbol{x}_1^3)}{(\boldsymbol{x}_2^2 - \boldsymbol{x}_1^2)} - 900 \end{aligned} \tag{55}$$

where the first objective $f_1(\boldsymbol{x})$ is the mass of the brake, the second objective $f_2(\boldsymbol{x})$ is the minimum stopping time of the disc brake, and the third objective is the sum of four constraint violations $g_1(\boldsymbol{x})$, $g_2(\boldsymbol{x})$, $g_3(\boldsymbol{x})$ and $g_4(\boldsymbol{x})$. For the decision

variables, $\boldsymbol{x}_1 \in [55, 80]$ is the inner radius of the discs, $\boldsymbol{x}_2 \in [75, 110]$ is the outer radius of the discs, $\boldsymbol{x}_3 \in [1000, 3000]$ is the engaging force, and $\boldsymbol{x}_4 \in [11, 20]$ is the number of friction surfaces.

Gear Train Design It is a multi-objective engineering design problem for a gear train proposed in Deb & Srinivasan (2006):

$$\begin{aligned} f_1(\boldsymbol{x}) &= |6.931 - \frac{\boldsymbol{x}_3\boldsymbol{x}_4}{\boldsymbol{x}_1\boldsymbol{x}_2}| \\ f_2(\boldsymbol{x}) &= \max\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\} \\ f_3(\boldsymbol{x}) &= \max\{-g_1(\boldsymbol{x}), 0\} \text{ where} \\ g_1(\boldsymbol{x}) &= 0.5 - \frac{f_1(\boldsymbol{x})}{6.931}. \end{aligned} \tag{56}$$

The first objective $f_1(\boldsymbol{x})$ is the difference between the realized gear ration and a given specific gear ration, the second objective $f_2(\boldsymbol{x})$ is the maximum size of the four gears, and the third objective $f_3(\boldsymbol{x})$ is the constraint violation of $g_1(\boldsymbol{x})$. The four integer decision variables $\boldsymbol{x}_i \in \{12, \dots, 60\}$ are the number of teeth in each of the four gears.

Rocket Injector Design It is a multi-objective engineering design problem for a rocket injector proposed in Vaidyanathan et al. (2003):

$$\begin{aligned} f_1(\boldsymbol{x}) =& 0.692 + 0.477\boldsymbol{x}_1 - 0.687\boldsymbol{x}_2 - 0.080\boldsymbol{x}_3 - 0.0650\boldsymbol{x}_4 \\ & - 0.167\boldsymbol{x}_1\boldsymbol{x}_1 - 0.0129\boldsymbol{x}_2\boldsymbol{x}_1 + 0.0796\boldsymbol{x}_2\boldsymbol{x}_2 - 0.00634\boldsymbol{x}_3\boldsymbol{x}_1 - 0.0257\boldsymbol{x}_3\boldsymbol{x}_2 \\ & + 0.0877\boldsymbol{x}_3\boldsymbol{x}_3 - 0.0521\boldsymbol{x}_4\boldsymbol{x}_1 + 0.00156\boldsymbol{x}_4\boldsymbol{x}_2 + 0.00198\boldsymbol{x}_4\boldsymbol{x}_3 + 0.0184\boldsymbol{x}_4\boldsymbol{x}_4 \\ f_2(\boldsymbol{x}) =& 0.153 + 0.322\boldsymbol{x}_1 - 0.396\boldsymbol{x}_2 - 0.424\boldsymbol{x}_3 - 0.0226\boldsymbol{x}_4 \\ & - 0.175\boldsymbol{x}_1\boldsymbol{x}_1 - 0.0185\boldsymbol{x}_2\boldsymbol{x}_1 + 0.0701\boldsymbol{x}_2\boldsymbol{x}_2 - 0.251\boldsymbol{x}_3\boldsymbol{x}_1 - 0.179\boldsymbol{x}_3\boldsymbol{x}_2 \\ & + 0.0150\boldsymbol{x}_3\boldsymbol{x}_3 - 0.0134\boldsymbol{x}_4\boldsymbol{x}_1 + 0.0296\boldsymbol{x}_4\boldsymbol{x}_2 + 0.0752\boldsymbol{x}_4\boldsymbol{x}_3 + 0.0192\boldsymbol{x}_4\boldsymbol{x}_4 \\ f_3(\boldsymbol{x}) =& 0.370 + 0.205\boldsymbol{x}_1 - 0.0307\boldsymbol{x}_2 - 0.108\boldsymbol{x}_3 - 1.019\boldsymbol{x}_4 \\ & - 0.135\boldsymbol{x}_1\boldsymbol{x}_1 - 0.0141\boldsymbol{x}_2\boldsymbol{x}_1 + 0.0998\boldsymbol{x}_2\boldsymbol{x}_2 - 0.208\boldsymbol{x}_3\boldsymbol{x}_1 - 0.0301\boldsymbol{x}_3\boldsymbol{x}_2 \\ & + 0.226\boldsymbol{x}_3\boldsymbol{x}_3 - 0.353\boldsymbol{x}_4\boldsymbol{x}_1 + 0.0497\boldsymbol{x}_4\boldsymbol{x}_3 + 0.423\boldsymbol{x}_4\boldsymbol{x}_4 + 0.202\boldsymbol{x}_2\boldsymbol{x}_1\boldsymbol{x}_1 \\ & - 0.281\boldsymbol{x}_3\boldsymbol{x}_1\boldsymbol{x}_1 - 0.342\boldsymbol{x}_2\boldsymbol{x}_2\boldsymbol{x}_1 - 0.245\boldsymbol{x}_2\boldsymbol{x}_2\boldsymbol{x}_3 + 0.281\boldsymbol{x}_3\boldsymbol{x}_3\boldsymbol{x}_2 \\ & - 0.184\boldsymbol{x}_4\boldsymbol{x}_4\boldsymbol{x}_1 - 0.281\boldsymbol{x}_2\boldsymbol{x}_1\boldsymbol{x}_3 \end{aligned} \tag{57}$$

where the first objective $f_1(\boldsymbol{x})$ is the maximum temperature of the injector face, the second objective $f_2(\boldsymbol{x})$ is the distance from the inlet, and the third objective $f_3(\boldsymbol{x})$ is the maximum temperature at the post tip. The decision variables $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4] \in [0, 1]^4$ are the hydrogen flow angle (α), the hydrogen area (ΔHA), the oxygen area (ΔOA), and the oxidizer post tip thickness (OPTT), respectively.

D.2.3. Hypervolume Definition

Following the related works on multi-objective optimization, we use the hypervolume metrics (Zitzler et al., 2007) to compare the qualities of approximate Pareto sets obtained by different methods. For a solution set P, its hypervolume $\text{HV}(P)$ in the objective space can be defined as the volume of the following dominated set:

$$S = \{\boldsymbol{r} \in \mathbb{R}^m \mid \exists \boldsymbol{y} \in P \text{ such that } \boldsymbol{y} \prec \boldsymbol{r} \prec \boldsymbol{r}^*\}, \tag{58}$$

where $\boldsymbol{r}^*$ is the reference point that is dominated by all solutions in P, every point $\boldsymbol{r} \in S$ will be dominated by at least one $\boldsymbol{y} \in P$, and we have $\text{HV}(P) = \textbf{Vol}(S)$. According to this definition, if a solution set A dominates another solution set B (e.g., every $\boldsymbol{b} \in B$ is dominated by at least one $\boldsymbol{a} \in A$), we will always have $\text{HV}(A) > \text{HV}(B)$. The ground-truth Pareto set P^* will always have the largest hypervolume $\text{HV}(P^*)$. In the experiments, we report the hypervolume difference to the optimal Pareto set for each algorithm:

$$\Delta\text{HV}(P) = \text{HV}(P^*) - \text{HV}(P) \tag{59}$$

where a solution set P with better quality should have smaller $\Delta\text{HV}(P)$ and $\Delta\text{HV}(P^*) = 0$. We report the average $\Delta\text{HV}(P)$ over 30 independent runs for each method.

E. Additional Experimental Studies

E.1. Office-31

Table 6. Results on the Office-31 dataset.

	Amazon	DSLR	Webcam	Avg↑	$\Delta_p \uparrow$
		Single-Task Baseline			
STL	86.61	95.63	96.85	93.03	0.00
		Adaptive Gradient Method			
MGDA	85.47	95.90	97.03	92.80±0.14	-0.27±0.15
GradNorm	83.58	97.26	96.85	92.56±0.87	-0.59±0.94
PCGrad	83.59	96.99	96.85	92.48±0.53	-0.68±0.57
GradDrop	84.33	96.99	96.30	92.54±0.42	-0.59±0.46
GradVac	83.76	97.27	96.67	92.57±0.73	-0.58±0.78
IMTL-G	83.41	96.72	96.48	92.20±0.89	-0.97±0.95
CAGrad	83.65	95.63	96.85	92.04±0.79	-1.14±0.85
MTAdam	85.52	95.62	96.29	92.48±0.87	-0.60±0.93
Nash-MTL	85.01	97.54	97.41	93.32±0.82	+0.24±0.89
MetaBalance	84.21	95.90	97.40	92.50±0.28	-0.63±0.30
MoCo	85.64	97.78	98.33	93.91±0.31	+0.89±0.26
Aligned-MTL	83.36	96.45	97.04	92.28±0.46	-0.90±0.48
IMTL	83.70	96.44	96.29	92.14±0.85	-1.02±0.92
DB-MTL	85.12	**98.63**	**98.51**	94.09±0.19	+1.05±0.20
		Adaptive Loss Method			
UW	83.82	97.27	96.67	92.58±0.84	-0.56±0.90
DWA	83.87	96.99	96.48	92.45±0.56	-0.70±0.62
IMTL-L	84.04	96.99	96.48	92.50±0.52	-0.63±0.58
IGBv2	84.52	98.36	98.05	93.64±0.26	+0.56±0.25
		Scalarization Method			
EW	83.53	97.27	96.85	92.55±0.62	-0.61±0.67
GLS	82.84	95.62	96.29	91.59±0.58	-1.63±0.61
RLW	83.82	96.99	96.85	92.55±0.89	-0.59±0.95
TCH	84.45	96.72	96.11	92.43±0.46	-0.71±0.56
STCH (Ours)	**86.66**	98.36	98.33	**94.45±0.23**	**+1.48±0.31**

Full results on the Office-31 dataset are shown in Table 6. We run the experiments with MoCO, TCH, and STCH methods by ourselves, and the rest results are from Lin et al. (2023) with the same setting. Our proposed STCH scalarization method performs the best on the Amazon task, and achieves the second best performance on DSLR and Webcam, which leads to the best average performance and the best Δ_p. In fact, it is the only method that can dominate the STL baseline on all tasks.

E.2. QM9

Table 7. Results on the QM9 dataset.

	μ	α	ϵ_{HOMO}	ϵ_{LUMO}	$\langle R^2 \rangle$	**ZPVE**	U_0	U	H	G	c_v	$\Delta_p \uparrow$
Single-Task Baseline												
STL	**0.062**	**0.192**	**58.82**	**51.95**	**0.529**	4.52	63.69	60.83	68.33	60.31	**0.069**	**0.00**
Adaptive Gradient Method												
MGDA	0.181	0.325	118.6	92.45	2.411	5.55	103.7	104.2	104.4	103.7	0.110	-103.0±8.62
GradNorm	0.114	0.341	67.17	84.66	7.079	14.6	173.2	173.8	174.4	168.9	0.147	-227.5±1.85
PCGrad	0.104	0.293	75.29	88.99	3.695	8.67	115.6	116.0	116.2	113.8	0.109	-117.8±3.97
GradDrop	0.114	0.349	75.94	94.62	5.315	15.8	155.2	156.1	156.6	151.9	0.136	-191.4±9.62
GradVac	0.100	0.299	68.94	84.14	4.833	12.5	127.3	127.8	128.1	124.7	0.117	-150.7±7.41
IMTL-G	0.670	0.978	220.7	249.7	19.48	55.6	1109	1117	1123	1043	0.392	-1250±90.9
CAGrad	0.107	0.296	75.43	88.59	2.944	6.12	93.09	93.68	93.85	92.32	0.106	-87.25±1.51
MTAdam	0.593	1.352	232.3	419.0	24.31	69.7	1060	1067	1070	1007	0.627	-1403±203
Nash-MTL	0.115	0.263	85.54	86.62	2.549	5.85	83.49	83.88	84.05	82.96	0.097	-73.92±2.12
MetaBalance	0.090	0.277	70.50	78.43	4.192	11.2	113.7	114.2	114.5	111.7	0.110	-125.1±7.98
MoCo	0.489	1.096	189.5	247.3	34.33	64.5	754.6	760.1	761.6	720.3	0.522	-1314±65.2
Aligned-MTL	0.123	0.295	98.07	94.56	2.397	5.90	86.42	87.42	87.19	86.75	0.106	-80.58±4.18
IMTL	0.138	0.344	106.1	102.9	2.595	7.84	102.5	103.0	103.2	100.8	0.110	-104.3±11.7
DB-MTL	0.112	0.264	89.26	86.59	2.429	5.41	60.33	60.78	60.80	60.59	0.098	**-58.10±3.89**
Adaptive Loss Method												
UW	0.336	0.382	155.1	144.3	0.965	4.58	61.41	61.79	61.83	61.40	0.116	-92.35±13.9
DWA	0.103	0.311	71.55	87.21	4.954	13.1	134.9	135.8	136.3	132.0	0.121	-160.9±16.7
IMTL-L	0.277	0.355	150.1	135.2	0.946	4.46	58.08	58.43	58.46	58.06	0.110	-77.06±11.1
IGBv2	0.235	0.377	132.3	139.9	2.214	5.90	64.55	65.06	65.12	64.28	0.121	-99.86±10.4
Scalarization Method												
EW	0.096	0.286	67.46	82.80	4.655	12.4	128.3	128.8	129.2	125.6	0.116	-146.3± 7.86
GLS	0.332	0.340	143.1	131.5	1.023	**4.45**	**53.35**	**53.79**	**53.78**	**53.34**	0.111	-81.16±15.5
RLW	0.112	0.331	74.59	90.48	6.015	15.6	156.0	156.8	157.3	151.6	0.133	-200.9±13.4
TCH	0.266	0.401	107.1	151.6	5.922	13.2	166.7	167.5	168.1	162.0	0.206	-252.2±16.6
STCH (Ours)	0.166	0.260	94.48	101.2	1.850	4.88	58.34	58.68	58.70	58.27	0.104	-58.14±4.18

Full results on the QM9 dataset are shown in Table 7. We run the experiments with TCH and STCH by ourselves, and the other results are from Lin et al. (2023). Our STCH scalarization has a promising second best overall performance, while the gap to the best DB-MTL (Lin et al., 2023) is very tight. It should be noticed that DB-MTL is an adaptive gradient method that requires a much longer runtime as reported in Table 4. Therefore, our proposed STCH scalarization method can serve as a very promising alternative for solving MTL problems.

E.3. More Comparison to Linear Scalarization

Table 8. Results on the CelebA dataset.

	Average Task Accuracy
Unit. Scal.	9.090e-01±7.568e-04
IMTL	9.093e-01±7.631e-04
MGDA	9.022e-01±9.687e-04
GradDrop	9.098e-01±3.383e-04
PCGrad	9.093e-01±1.108e-03
RLW Diri.	9.099e-01±7.845e-04
RLW Norm.	9.095e-01±1.012e-03
STCH (Ours)	9.098e-01±4.692e-04

We also test our proposed STCH scalarization on the CelebA dataset with the same setting as in Kurin et al. (2022). We independently run STCH 3 times and report the mean of average task accuracy and standard deviation, while all other results are directly from Kurin et al. (2022). STCH has a simple equal weight for all tasks which is not tuned. According to the results in Table 8, STCH achieves similar performance with most other methods, and significantly outperforms MGDA. We find that STCH achieves almost zero training loss for all tasks, which is consistent with most other methods as reported in Kurin et al. (2022). In this case, the testing accuracy mainly depends on the generalization performance rather than the optimization performance (on training loss). Since the STCH scalarization is developed from the viewpoint of multi-objective optimization, we currently do not have a straightforward way to analyze its generalization performance. Recently, Chen et al. (2023) have investigated the generalization gap for the stochastic adaptive gradient method. We leave the generalization analysis for STCH in the stochastic setting to future work.

On the other hand, our proposed STCH scalarization is lightweight, with a computational overhead similar to that of linear scalarization. It is also interesting to further investigate its performance with tuned weights and/or regularization for multi-task learning as in Kurin et al. (2022) and Xin et al. (2022).

E.4. Multi-Objective Bayesian Optimization

Table 9. Results (hypervolume difference ΔHV $\downarrow$) for multi-objective Bayesian optimization on 6 synthetic benchmark problems and 5 real-world engineering design problems.

	F1	F2	F3	F4	F5	F6	BarTruss	PressureVessel	DiskBrake	GearTrain	RocketInjector
TCH	0.0183	0.0185	0.0413	0.0891	0.0602	0.0917	0.0429	0.0441	0.0262	0.0262	0.0449
STCH	**0.0147**	**0.0110**	**0.0304**	**0.0476**	**0.0455**	**0.0575**	**0.0233**	**0.0194**	**0.0214**	**0.0198**	**0.0332**

In this subsection, we test our proposed STCH on multi-objective Bayesian optimization. We follow the setting proposed in Lin et al. (2022c), where a Pareto set learning approach with TCH scalarization (PSL-TCH) has been proposed for multiobjective Bayesian optimization. We simply replace the TCH scalarization in PSL-TCH with the proposed STCH scalarization to obtain a new method PSL-STCH. We then compare the new PSL-STCH with PSL-TCH on different synthetic and multiobjective engendering design problems with a small evaluation budget (i.e. 100 evaluations) as in Lin et al. (2022c). According to Table 9 with the hypervolume difference (ΔHV) metrics, PSL-STCH can outperform PSL-TCH on all problems. This result confirms that the proposed STCH scalarization can serve as an off-the-shelf approach for different multiobjective optimization applications, including Bayesian optimization.

E.5. Efficient Pareto Set Learning

Table 10. Results (hypervolume HV ↑) on 6 synthetic benchmark problems and 5 real-world engineering design problems.

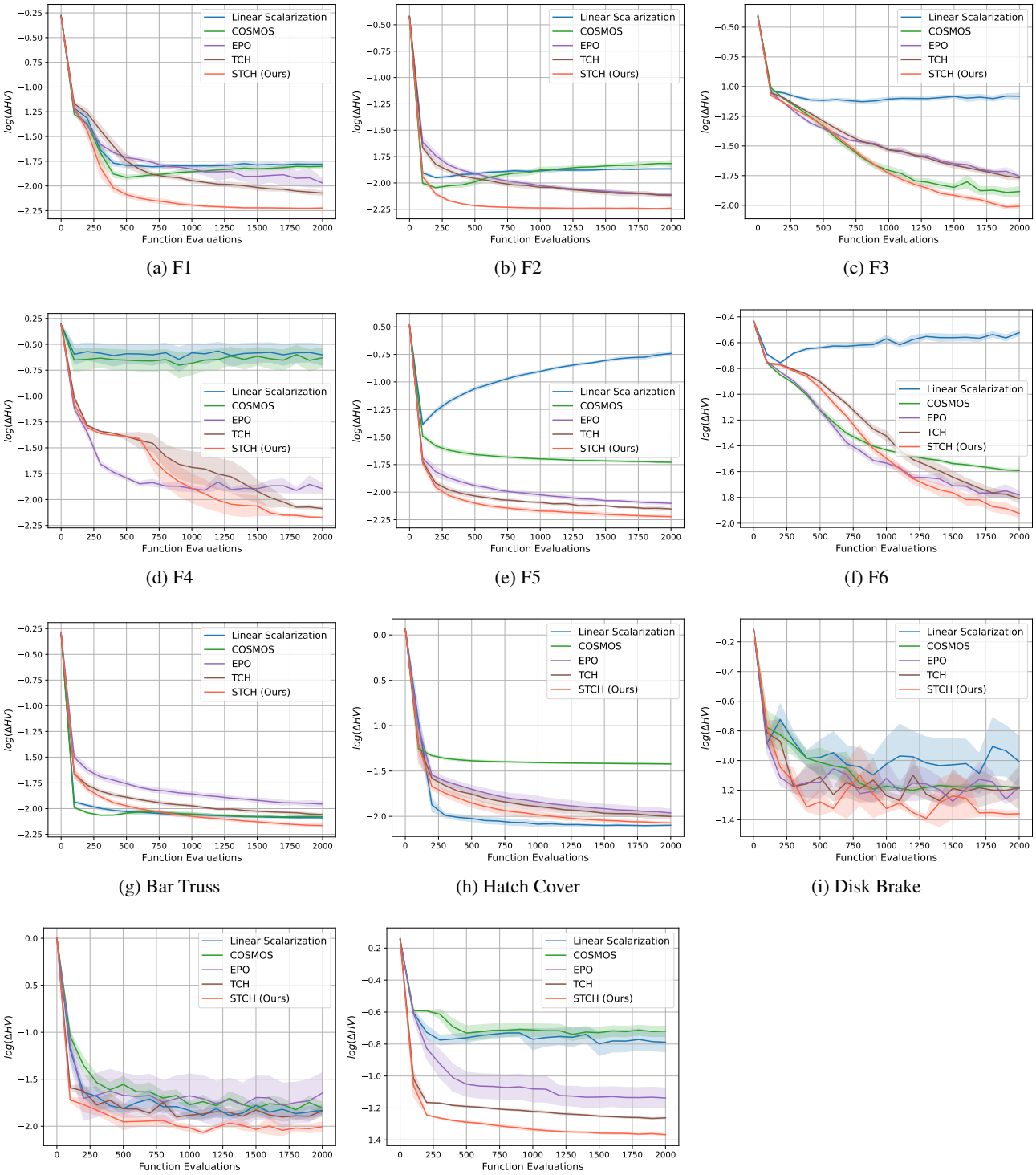
	F1	F2	F3	F4	F5	F6	BarTruss	HatchCover	DiskBrake	GearTrain	RocketInjector
LS	0.8602	0.8629	0.7826	0.3172	0.3712	0.2892	0.8805	**1.1633**	0.9738	1.0169	0.7051
COSMOS	0.8608	0.8614	0.8638	0.5283	0.5300	0.5242	0.8803	1.1425	0.9710	1.0174	0.8091
EPO	0.8653	0.8689	0.8564	0.5324	0.5349	0.5236	0.8772	1.1592	0.9805	1.0175	0.7889
TCH	0.8676	0.8686	0.8582	0.5345	0.5364	0.5287	0.8795	1.1611	0.9765	1.0170	0.8198
STCH	**0.8707**	**0.8709**	**0.8670**	**0.5365**	**0.5372**	**0.5316**	**0.8829**	1.1632	**0.9864**	**1.0177**	**0.8363**

Hypervolume Metrics For the efficient Pareto set learning problem, we report the hypervolume for all problems in Table 10. For each problem, we first normalize the objective value of all solutions for all methods into $[0, 1]$ with the same nadir point and ideal point, then calculate the hypervolume with reference point $[1.1, 1.1]^m$ where m is the number of objectives. Our proposed STCH achieves the best overall performance on these problems.

Table 11. The run time for learning the Pareto set of 5 real-world enginnering design problems with different methods.

Method	LS	COSMOS	EPO	TCH	STCH
Bar Truss	6s (1x)	6s (1x)	268s (45x)	6s (1x)	6s (1x)
Hatch Cover	6s (1x)	6s (1x)	277s (46x)	6s (1x)	6s (1x)
Disk Brake	9s (1x)	9s (1x)	329s (37x)	9s (1x)	9s (1x)
Gear Train	9s (1x)	9s (1x)	337s (38x)	9s (1x)	9s (1x)
Rocket Injector	9s (1x)	9s (1x)	324s (36x)	9s (1x)	9s (1x)

Run time In addition, we also report the runtimes of different methods to learn the Pareto set for five multi-objective engineering design problems in Table 11. According to the results, our proposed STCH scalarization method share a similar runtime with other scalarization methods (e.g., LS, COSMOS, and TCH), while the gradient-based EPO method takes significantly longer runtime (36x to 44x) to learn the Pareto set.



(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

(g) Bar Truss (h) Hatch Cover (i) Disk Brake

(j) Gear Train (k) Rocket Injector

Figure 6. The log hypervolume difference ($\log(\Delta HV)$) of different methods during the optimization process for 11 different multi-objective optimization problems.