ROWA: A LOW-COMPLEXITY ADAPTIVE WEIGHT FRAMEWORK FOR MULTI-TASK LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Gradient manipulation methods in Multi-Task Learning (MTL) are effective for balancing competing objectives, yet they often suffer from severe computational bottlenecks. Recent α-aware utility optimization approaches unify various task-level fairness criteria, but require expensive $\mathcal{O}(K)$ task-specific backward passes at each training step, where K denotes the number of tasks. To overcome this scalability barrier, we propose ROWA (GeneRic Online Weight Adaptation), which is derived from an online optimization formulation incorporating the loss decrease and α-aware targets. ROWA is implementation-friendly governed by a single mirror descent step at each iteration, and completely obviates the need for any task-specific gradient computation during task weighting. Remarkably, ROWA can be generalized to arbitrary α-configurations and readily incorporate the Weighted-Chebyshev strategy to enable exploration of user-preferred regions on the Pareto front. We establish an $\mathcal{O}(1/\sqrt{T})$ iteration complexity for the convergence of ROWA to Pareto stationarity in the nonconvex setting. Extensive experiments on four benchmarks with two to 40 tasks show that ROWA outperforms a wide range of strong baselines at negligible computational overhead. Ablation studies further confirm its flexibility across fairness criteria and Pareto-front preferences.

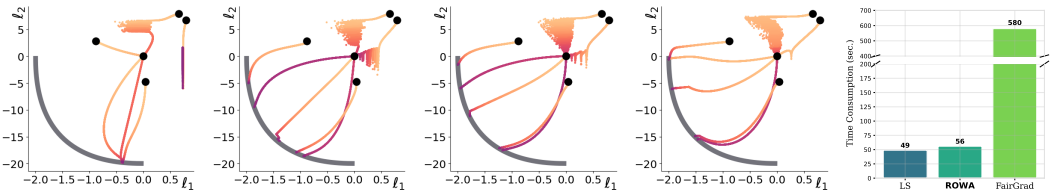
1 INTRODUCTION

Multi-Task Learning (MTL) has emerged as a highly effective paradigm for predicting multiple tasks simultaneously through training a single model. Compared to training each task individually, MTL significantly reduces computational overhead while simultaneously improving data efficiency and model generalization. The adoption of MTL has spanned natural language processing (Zhang et al., 2023), computer vision (Vandenhende et al., 2021), autonomous driving (Chen et al., 2018) and recommendation systems (Wang et al., 2020). A key challenge in MTL stems from resolving task imbalance, where heterogeneous losses and conflicting gradients severely under-optimize certain tasks, resulting in degraded overall performance and causing disparate convergence behavior.

In the literature, a series of gradient manipulation MTL methods have been proposed to address the imbalance issue, such as (Désidéri, 2012; Yu et al., 2020a; Liu et al., 2021; Xiao et al., 2024; Liu et al., 2024; Ban & Ji, 2024). Recently, Ban & Ji (2024) established a generic α-aware utility optimization problem that unifies a broad range of MTL methods by varying the hyperparameter α: setting $\alpha = 0$ recovers linear scalarization (LS); $\alpha = 1$ encodes a bargaining game which derives Nash-MTL (Navon et al., 2022) and its follow-ups (Zhou et al., 2025); $\alpha = 2$ implies minimizing the average steps required for unit progress (Qin et al., 2025b;a; Wang et al., 2026); the extreme case of $\alpha = +\infty$ is equivalent to a min-max optimization problem that corresponds to MGDA (Navon et al., 2022) and its subsequent variants (Liu et al., 2021; Fernando et al., 2023; Xiao et al., 2024; Zhang et al., 2025; Xu et al., 2025).

However, the family of α-fair MTL methods with different choices of α often require all K task-specific backward passes at every training step. In contrast to the standard $\mathcal{O}(1)$ gradient computation for updating shared parameters, this additional task-weighting cost can become a major bottleneck in MTL with deep architectures and large-scale datasets. Although Liu et al. (2024) improves the scalability of MGDA by reducing the gradient complexity to $\mathcal{O}(1)$, it is limited to the simplified min-max optimization setting (corresponding to $\alpha = +\infty$). Thus, a natural question arises:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) $\alpha = 0.01$ (b) $\alpha = 1$ (c) $\alpha = 2$ (d) $\alpha = +\infty$ (e) runtime

Figure 1: An illustrative 2-task example from (Ban & Ji, 2024) to show the convergence of ROWA to the Pareto front from different initialization points, and the runtime comparison of different MTL methods for 50000 steps: (a)-(d) demonstrate that ROWA embodies different α criterias. (e) shows that ROWA incurs a time cost comparable to LS, yet significantly more time-efficient compared to FairGrad. More details and analysis are provided in Section A.

Is it possible to develop an $\mathcal{O}(1)$ MTL framework that can easily scale to different α settings without any task-specific gradient computation?

This work provides an affirmative answer by presenting ROWA (GeneRic Online Weight Adaptation). The key idea of ROWA is to construct an α-aware dynamic loss-tracking process and frame the resulting weight adaptation through the lens of online optimization, driving the weight update via a single mirror descent step. Since the update relies only on observed task losses, it adds negligible cost on top of standard LS. We further demonstrate the flexibility of the framework by incorporating the Weighted-Chebyshev scalarization (Ehrgott, 2005), which steers training toward user-specified regions of the Pareto front. Illustrative examples are given in Figures 1 and 2. Our main contributions are summarized as follows:

- We develop ROWA, a low-complexity weight adaptation framework driven by a mirror descent step. The framework is generic: a single parameter α interpolates across task-level fairness criteria, and a preference vector enables broad exploration of the Pareto front. Theoretically, we establish an $\mathcal{O}(1/\sqrt{T})$ convergence rate to a Pareto stationary point for nonconvex objectives.
- ROWA incurs only $\mathcal{O}(1)$ overhead per iteration, requiring neither per-task gradient computation nor an auxiliary solver. Extensive experiments on four benchmarks with 2 to 40 tasks show that ROWA outperforms a wide range of baselines while matching the efficiency of LS. Ablation studies further show that varying α controls the fairness–efficiency trade-off and that the preference vector reliably guides solutions to the desired region of the Pareto front.

2 Related Works

To address the task imbalance, a highly related line of research frames MTL as a multi-objective optimization problem, where existing methods can be categorized into two classes: gradient-oriented and loss-oriented methods.

Gradient-oriented methods. Explicitly accessing to the task-specific gradient information, gradient-oriented methods leverage strategies such as Pareto optimization (Sener & Koltun, 2018), gradient projection (Yu et al., 2020a) or conflict mitigation (Désidéri, 2012; Liu et al., 2021; Navon et al., 2022; Xiao et al., 2024; Zhang et al., 2024; Ban & Ji, 2024; Qin et al., 2025b;a; Wang et al., 2026) to avoid negative transfer and enhance performance. However, the computational overhead of gradient-oriented methods becomes prohibitive when scaling to a large number of tasks.

Loss-oriented methods. By operating on task-specific losses, loss-oriented methods effectively avoiding the computational burden of per-task gradients. One straightforward line of approaches relies on scalarization methods like (Lin et al., 2022; 2024). Weighted-Chebyshev (WC) is a widely used scalarization technique (Ehrgott, 2005), which aims to minimize the worst-performing objective under the guidance of user preferences (Momma et al., 2022; Qiu et al., 2024). Recent advances (Lin et al., 2024; Zhang et al., 2026b) have rendered the WC strategy applicable to modern machine learning. Another line focus on adaptive loss weighting methods where task weights are adjusted

throughout the training process, via strategies like uncertainty modeling (Kendall et al., 2018), rates of loss changes (Liu et al., 2019), task grouping (Shen et al., 2024), decreasing losses at an equal rate (Liu et al., 2024) and loss discrepancy control (Xiao et al., 2026).

Notations. We use a, $\boldsymbol{a}$, $\boldsymbol{A}$ and $\mathcal{A}$ to represent a scalar, vector, matrix and set correspondingly. A vector $\boldsymbol{a} = (a_1, \cdots, a_d) \in \mathbb{R}^d$ is viewed as a column vector, unless clearly stated. A matrix $\boldsymbol{A} = (\boldsymbol{a}_1^\top, \cdots, \boldsymbol{a}_K^\top)$ is viewed as $\mathbb{R}^{K\times d}$ dimension. $\odot$ denotes the element-wise product between two vectors of the same dimension. $\|\cdot\|$ refers to ℓ_2-norm. We denote $\texttt{softmax}(\boldsymbol{a}) := \frac{\exp(\boldsymbol{a})}{\sum_{j=1}^K \exp(a_j)}$.

3 PRELIMINARIES

3.1 A GENERIC α-AWARE UTILITY FORMULATION

MTL aims to optimize a single model with parameter $\theta \in \mathbb{R}^d$, learning a shared generalized representation to solve $K \geq 2$ distinct tasks simultaneously. MTL can be formulated as $\min_{\theta\in\mathbb{R}^d} \boldsymbol{L}(\theta) := (l_1(\theta), \cdots, l_K(\theta))$, where each task i among K tasks is associated with a loss function $l_i(\theta) : \mathbb{R}^d \to \mathbb{R}_+$. Denote the gradient of $l_i(\theta)$ as $\boldsymbol{g}_i(\theta) = \nabla l_i(\theta) \in \mathbb{R}^d$. During the MTL optimization process, the update of θ at each iteration t proceeds as $\theta_{t+1} = \theta_t - \eta\boldsymbol{d}$, where η is the learning rate and $\boldsymbol{d}$ is the update direction that effectively balances the task-specific gradients. Recently, Ban & Ji (2024) identifies a common improving direction $\boldsymbol{d}$ by maximizing the α-fair aggregation of task-specific utilities in the interior of a l_2-norm ball B_ϵ ($\|\boldsymbol{d}\|^2 \leq d_\epsilon^2$), i.e.,

$$\arg\max_{\boldsymbol{d}\in B_\epsilon} \sum_{i=1}^K \frac{(\boldsymbol{g}_i(\theta)^\top \boldsymbol{d})^{1-\alpha}}{1-\alpha}, \text{ s.t. } \boldsymbol{g}_i(\theta)^\top \boldsymbol{d} \geq 0 \tag{1}$$

This utility optimization problem unifies a wide range of MTL methods corresponding to $\alpha = 0, 1, 2$ and $+\infty$, as mentioned in the introduction.

3.2 THE SCALABILITY BARRIER IN TASK WEIGHTING

Notably, Eq. (1) admits a general solution for $\boldsymbol{d}$ across all α settings. To be specific, by introducing Lagrange multipliers λ and μ_i ($i \in [K]$), we can construct the Lagrangian function on θ_t as $\mathcal{L}(\boldsymbol{d}) = \sum_{i=1}^K \frac{(\boldsymbol{g}_i(\theta_t)^\top \boldsymbol{d})^{1-\alpha}}{1-\alpha} - \frac{\lambda}{2}(\|\boldsymbol{d}\|^2 - d_\epsilon^2) + \sum_{i=1}^K \mu_i(\boldsymbol{g}_i(\theta_t)^\top \boldsymbol{d})$, which yields the stationary condition $\sum_{i=1}^K \left[(\boldsymbol{g}_i(\theta_t)^\top \boldsymbol{d})^{-\alpha} + \mu_i\right] \boldsymbol{g}_i(\theta_t) = \lambda\boldsymbol{d}$, where $\mu_i = 0$ due to the complementary slackness. As commonly formulated in many MTL works, $\boldsymbol{d}$ is represented as a linear combination of task gradients, i.e., $\boldsymbol{d} = \boldsymbol{G}_t\boldsymbol{w}$, where $\boldsymbol{G}_t = (\boldsymbol{g}_1(\theta_t)^\top, \cdots, \boldsymbol{g}_K(\theta_t)^\top) \in \mathbb{R}^{K\times d}$, and $\boldsymbol{w} = (w_1, \cdots, w_K) \in \mathbb{R}_+^K$ denotes the nonnegative weights corresponding to K tasks. Then, determining $\boldsymbol{d}$ translates to selecting an appropriate $\boldsymbol{w}$. By plugging $\boldsymbol{d} = \boldsymbol{G}_t\boldsymbol{w}$ into the stationary condition and denoting $\xi := \lambda^{-1/\alpha}$, an explicit equation for $\boldsymbol{w}$ is established as

$$\boldsymbol{G}_t^\top \boldsymbol{G}_t \boldsymbol{w} = \xi \boldsymbol{w}^{-1/\alpha}. \tag{2}$$

Despite the explicit formulation in Eq. (2), solving for $\boldsymbol{w}$ is highly inefficient. The presence of the Gram matrix $\boldsymbol{G}_t^\top \boldsymbol{G}_t$ term necessitates the explicit computation of K task-specific gradients at each iteration t. Although the recent work (Liu et al., 2024) circumvents the explicit computation of the Gram matrix by leveraging the loss history, it is tailored for the $\alpha = +\infty$ scenario, thereby precluding its scalability to broader α criteria. In light of these limitations, this work develops an α-aware task-weighting adaptation to fundamentally enhance its scalability.

4 THE PROPOSED ROWA FRAMEWORK

In this section, we will first present the algorithmic design of ROWA in Section 4.1, which is followed by theoretical analysis in Section 4.2 and discussions on practical implementation in Section 4.3.

4.1 ALGORITHMIC DESIGN

The development of our proposed ROWA framework consists of two key steps:

Step 1) Low-Complexity Online Weight Adaptation: Firstly, we adopt the first-order Taylor expansion of $\log l_i(\cdot)$ around θ_t, i.e., $\log l_i(\theta_{t+1}) - \log l_i(\theta_t) \approx \frac{\boldsymbol{g}_i(\theta_t)^\top}{l_i(\theta_t)}(\theta_{t+1} - \theta_t)$ for any $i \in [K]$, which yields

$$\eta^{-1} \cdot l_i(\theta_t)(\log l_i(\theta_t) - \log l_i(\theta_{t+1})) \approx \boldsymbol{g}_i(\theta_t)^\top \boldsymbol{G}_t \boldsymbol{w}. \tag{3}$$

As observed from Eq. (3), the right-hand side term that requires task-specific gradient computation, can be effectively replaced by the step-wise log-loss decrease scaled by the current loss magnitude. Whereas, by substituting this gradient-free surrogate back into Eq. (2), the original algebraic balance of $\boldsymbol{w}$ is broken. This structural imbalance precludes any straightforward static equation-solving, but transforming it into a time-varying loss tracking process driven by an α-aware weight regulation. An intuitive way is to find an appropriate $\boldsymbol{w}$ to drive $\frac{\boldsymbol{L}(\theta_t) \odot (\log \boldsymbol{L}(\theta_t) - \log \boldsymbol{L}(\theta_{t+1}))}{\eta} - \xi \boldsymbol{w}^{-1/\alpha}$ to $\mathbf{0}$. Nevertheless, the loss decrease can be highly volatile and occasionally negative at each practical training iteration t, making it infeasible to enforce this condition.

Moving beyond the root-finding paradigm, we turn to focus on $\boldsymbol{q}_t := \frac{\boldsymbol{L}(\theta_t) \odot (\log \boldsymbol{L}(\theta_t) - \log \boldsymbol{L}(\theta_{t+1}))}{\eta} - \xi \boldsymbol{w}_t^{-1/\alpha}$ and treat it through the lens of online optimization. $\boldsymbol{q}_t$ incorporates $\boldsymbol{L}(\theta_{t+1})$ which serves as the environment feedback to the weight $\boldsymbol{w}_t$ decided at iteration t. Structuring as the actual loss progress minus the expected target, $\boldsymbol{q}_t$ can be viewed as a cost vector revealed sequentially along the optimization trajectory. Driven by this insight, our goal is to minimize the *cumulative regret* over the entire T iterations, defined as

$$R_T := \sum_{t=1}^{T} \langle \boldsymbol{q}_t, \boldsymbol{w}_t \rangle - \min_{\boldsymbol{w} \in \mathcal{W}} \sum_{t=1}^{T} \langle \boldsymbol{q}_t, \boldsymbol{w} \rangle. \tag{4}$$

We project $\boldsymbol{w}$ on the constraint $\mathcal{W} = \{\boldsymbol{w} \in \mathbb{R}_+^K \mid \sum_{i=1}^K w_i = K\}$, so as to enforce a relative weight redistribution among tasks while preserving the scale equivalence to the unweighted sum. Since $\boldsymbol{q}_t$ is revealed sequentially during training, we use the classical Mirror Descent method to update $\boldsymbol{w}_t$: For any vectors $\boldsymbol{u}$ and $\boldsymbol{w}$, define the KL distance $D_{KL}(\boldsymbol{u} \parallel \boldsymbol{w}) = \sum_{i=1}^K u_i \log \frac{u_i}{w_i}$, then

$$\boldsymbol{w}_{t+1} \leftarrow \arg\min_{\boldsymbol{w} \in \mathcal{W}} \left\{ \langle \boldsymbol{q}_t, \boldsymbol{w} \rangle + \beta^{-1} D_{KL}(\boldsymbol{w} \parallel \boldsymbol{w}_t) \right\}. \tag{5}$$

Solving Eq. (5) yields $w_{i,t+1} \propto w_{i,t} \exp(-\beta q_{i,t})$ for each task i, such that the optimization trajectory can adapt to time-varying learning paces across tasks: if task i's loss decrease shrinks or even turns negative, falling behind its target $\xi w_{i,t}^{-1/\alpha}$, the suppression on its weight $w_{i,t}$ is relatively alleviated, thereby driving a compensatory increase; if task i's loss decrease surges, substantially exceeding its target, the suppression on its weight $w_{i,t}$ will be enforced, thereby driving a penalizing decay.

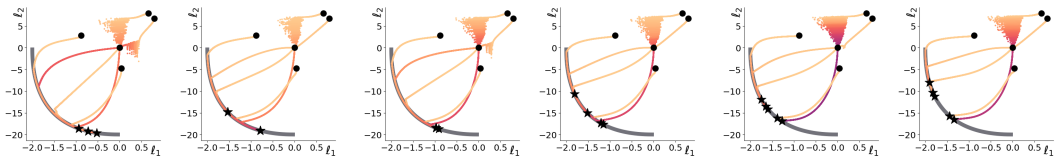
Remark 1. For tasks with small loss magnitudes, their absolute loss decrease can become negligibly small, causing the update of their corresponding $w_{i,t}$ to stagnate and lose regulating capability. Thus, the log-based approximation is adopted in Eq. (3), so as to amplify the competitiveness of small-loss tasks during weight allocation, without losing the approximation error control. The rationale is that, by further applying the second-order Taylor expansion $-\log(1-a) = a + \frac{a^2}{2} + o(a^3)$ where $a = \frac{l_i(\theta_t) - l_i(\theta_{t+1})}{l_i(\theta_t)}$, it can be deduced that

$$l_i(\theta_t)(\log l_i(\theta_t) - \log l_i(\theta_{t+1})) \approx l_i(\theta_t) - l_i(\theta_{t+1}) + (2l_i(\theta_t))^{-1}(l_i(\theta_t) - l_i(\theta_{t+1}))^2. \tag{6}$$

The log-based approximation incorporates the impact of the intrinsic loss magnitude $l_i(\theta_t)$ on the scale of the loss decrease, which amplifies the relative progress of small-loss tasks.

Step 2) Weighted-Chebyshev Augmentation for Pareto Front Exploration: Next, we will leverage the online weight adaptation technique derived in Step 1) and augment it with the weighted Chebyshev scalarization technique to facilitate Pareto front exploration. Specifically, by introducing weights $\boldsymbol{c} := (c_1, \cdots, c_K) \in \mathbb{R}_+^K$, the utility problem can be further generalized to the weighted version $\arg\max_{\boldsymbol{d} \in B_\epsilon} \sum_{i=1}^K c_i \frac{(\boldsymbol{g}_i(\theta_t)^\top \boldsymbol{d})^{1-\alpha}}{1-\alpha}$, s.t. $\boldsymbol{g}_i(\theta_t)^\top \boldsymbol{d} \geq 0$. Following similar derivation steps, we can obtain $\boldsymbol{G}_t^\top \boldsymbol{G}_t \boldsymbol{w} = \xi \boldsymbol{c}^{1/\alpha} \odot \boldsymbol{w}^{-1/\alpha}$, which is a weighted version of Eq. (2). Building upon the core rationale of our framework (i.e., Eq. (3)-Eq. (5)), the scalable formulation via the mirror descent step is achieved by

$$\boldsymbol{w}_{t+1} \leftarrow \arg\min_{\boldsymbol{w} \in \mathcal{W}} \left\langle \frac{\boldsymbol{L}(\theta_t) \odot (\log \boldsymbol{L}(\theta_t) - \log \boldsymbol{L}(\theta_{t+1}))}{\eta} - \xi \boldsymbol{c}^{1/\alpha} \odot \boldsymbol{w}_t^{-1/\alpha}, \boldsymbol{w} \right\rangle + \frac{D_{KL}(\boldsymbol{w} \parallel \boldsymbol{w}_t)}{\beta}.$$



(a) $\boldsymbol{r}$=[0.5, 0.5] (b) $\boldsymbol{r}$=[0.7, 0.3] (c) $\boldsymbol{r}$=[0.9, 0.1]

Figure 2: Figures (a)-(c) show the Pareto front evolution across different α and different preference $\boldsymbol{r} = [r_1, r_2]$ assigned to task 1 and 2 (corresponding to loss ℓ_1 and ℓ_2), respectively. The **left** and **right columns** correspond to $\alpha = 1$ and 2. Figures show that ROWA can cover a broader range of Pareto front (represented by the black stars). More cases are provided in Appendix A.

To equip our framework with the capability of systematic Pareto front exploration, we integrate the Weighted-Chebyshev (WC) technique. Specifically, we design $\boldsymbol{c}$ by solving the WC scalarization problem $\min_\theta \max_{i\in[K]}\{r_i l_i(\theta)\}$, s.t. $r_i > 0, \sum_{i=1}^K r_i = 1$ (Momma et al., 2022; Lin et al., 2024; Zhang et al., 2026a), where $\boldsymbol{r} = (r_1, \cdots, r_K)$ represents the user-decided preference weight assigned to K tasks, with a larger r_i-value indicating a stronger preference to task i.

To address the non-smooth challenge due to the min-max nature of WC, we consider a smoothed version of the WC scalarization following (Lin et al., 2024), i.e., $\min_\theta \tau \log \sum_{i=1}^K \exp\left(r_i l_i(\theta)/\tau\right)$, where τ is a smoothing parameter. Then, we obtain $\boldsymbol{c}$ as follows:

$$\nabla_\theta \left(\tau \log \sum_{i=1}^K \exp\left(\frac{r_i l_i(\theta)}{\tau}\right)\right) = \sum_{i=1}^K \underbrace{\left[\texttt{softmax}\left(\boldsymbol{r} \odot \boldsymbol{L}(\theta)/\tau\right)\right]_i \cdot r_i}_{:=c_i(\theta),\ \forall i\in[K]} \cdot \boldsymbol{g}_i(\theta), \tag{7}$$

which adaptively accounts for both the losses $\boldsymbol{L}(\theta)$ and the $\boldsymbol{r}$-preference. When setting $\alpha = 0$ in the weighted utility problem, it can be checked that $\boldsymbol{d} = \sum_{i=1}^K c_i(\theta_t)\boldsymbol{g}_i(\theta_t)$ reduces to Eq. (7), which implies that the update direction aligns with the steepest descent trajectory of the smoothed WC objective. Integrated with the WC-based $\boldsymbol{c}(\theta_t)$, our framework is able to balance the trade-off between the α fairness and the user-specified preferences, as illustrated in Figure 2. To conclude the discussions on our algorithmic design, we summarize the ROWA framework in Algorithm 1.

4.2 Finite-Time Pareto Stationarity Convergence Analysis

To analyze the finite-time convergence rate of ROWA to a Pareto stationary solution, we adopt an assumption widely adopted in the online optimization literature:

Assumption 1 (Boundedness of L-Smoothness of the Loss Function)**.** For each $i \in [K]$, there exists a constant $C < \infty$ such that $l_i(\theta) < C$, and $l_i(\theta)$ is L-gradient lipschitz continuous.

Based on Assumption 1, we can establish the following main theoretical result (the proof is relegated to Appendix D due to space limitation):

Theorem 1. *Let $\delta_1, \delta_2 > 0$ be two constants that can be chosen arbitrarily small. Under Assumption 1, by setting the learning rate $\eta \le \frac{1}{LK(3+4\delta_1^{-1}(2K^2/9+4))}$, $\beta = \mathcal{O}(1/K\sqrt{T})$ and denoting $\Delta_{D_{KL}} := D_{KL}(\boldsymbol{u} \parallel \boldsymbol{w}_1) - D_{KL}(\boldsymbol{u} \parallel \boldsymbol{w}_{T+1})$ for any $\boldsymbol{u} \in \mathcal{W}$, it holds that $\frac{1}{T}\sum_{t=1}^T \|\boldsymbol{G}_t\boldsymbol{w}_t\|^2 \le \frac{1}{T}\sum_{t=1}^T 2\xi \mathbf{1}^\top(\boldsymbol{c}(\theta_t)^{1/\alpha} \odot \boldsymbol{w}_t^{1-1/\alpha}) + \mathcal{O}\left(\frac{K\Delta_{D_{KL}}+\xi^2+K^2L^2C^2}{\sqrt{T}}\right)$. Furthermore, by letting $\xi = \mathcal{O}(1/\sqrt{T})$, denoting $\Phi_{\alpha,\|\mathbf{r}\|_\infty} := \mathbb{I}(\alpha > 1)\|\mathbf{r}\|_\infty^{1/\alpha} K^{1-1/\alpha} + \mathbb{I}(\alpha = 1)\|\mathbf{r}\|_\infty + \mathbb{I}(0 < \alpha < 1)\delta_2^{1-1/\alpha}\|\mathbf{r}\|_\infty^{1/\alpha}$, and defining the normalized simplex weight $\boldsymbol{v}_t = \frac{\boldsymbol{w}_t}{K}$, it follows that:*

$$\frac{1}{T}\sum_{t=1}^T \|\boldsymbol{G}_t\boldsymbol{v}_t\|^2 = \mathcal{O}\left(\frac{\Phi_{\alpha,\|\boldsymbol{r}\|_\infty}}{K^2\sqrt{T}} + \frac{\Delta_{D_{KL}}}{K\sqrt{T}} + \frac{L^2C^2}{\sqrt{T}}\right). \tag{8}$$

Algorithm 1 The ROWA algorithm.

1: **Input:** Initial model θ_1 and weight $\boldsymbol{w}_1$. Stepsize η and β. Parameters α, $\boldsymbol{r}$, τ and γ.
2: **for** $t = 1$ to T **do**
3: Update the shared model parameter:

$$\theta_{t+1} \leftarrow \theta_t - \mathrm{Adam}(\eta \mathbf{G}_t \boldsymbol{w}_t)$$

4: Compute the weight from the smoothed Weighted-Chebyshev scalarization:

$$\boldsymbol{c}(\theta_t) \leftarrow \texttt{softmax}(\boldsymbol{r} \odot \boldsymbol{L}(\theta_t)/\tau) \odot \boldsymbol{r}$$

5: Compute the online cost vector:

$$\boldsymbol{u}_t \leftarrow \boldsymbol{L}(\theta_t) \odot (\log \boldsymbol{L}(\theta_t) - \log \boldsymbol{L}(\theta_{t+1}))\,,\ \boldsymbol{v}_t \leftarrow \boldsymbol{c}(\theta_t)^{1/\alpha} \odot \boldsymbol{w}_t^{-1/\alpha}$$
$$\boldsymbol{h}_t \leftarrow \frac{\boldsymbol{u}_t}{\boldsymbol{u}_t.\mathrm{abs}().\mathrm{mean}()} - \frac{\boldsymbol{v}_t}{\boldsymbol{v}_t.\mathrm{mean}()} + \gamma \log \boldsymbol{w}_t$$

6: Update task weights via the mirror descent:

$$\boldsymbol{w}_{t+1} \leftarrow \frac{\boldsymbol{w}_t \odot \exp(-\beta \cdot \boldsymbol{h}_t)}{(\boldsymbol{w}_t \odot \exp(-\beta \cdot \boldsymbol{h}_t)).\mathrm{mean}()}$$

7: **end for**
8: **Output:** θ_T

Theorem 1 establishes that for a sufficiently large T, $\frac{1}{T}\sum_{t=1}^{T} \|\boldsymbol{G}_t \boldsymbol{w}_t\|^2 \lesssim \frac{1}{T}\sum_{t=1}^{T} 2\xi \mathbf{1}^\top (\boldsymbol{c}(\theta_t)^{1/\alpha} \odot \boldsymbol{w}_t^{1-1/\alpha})$ is achieved. Furthermore, Eq. (8) implies the ROWA method converges to a Pareto stationary solution with an iteration complexity of $\mathcal{O}(1/\sqrt{T})$ in the non-convex settings.

4.3 Practical Implementation Considerations for ROWA

As in most optimization methods for machine learning, one could further adopt various acceleration techniques in ROWA to update the model θ (e.g., the Adam-type adaptive learning rate selection). However, adaptive learning rates render the learning rate η in Eq. (3) unpredictable and intractable to manual tuning. Here, we present a practical version of ROWA, whose pseudo-code is shown in Algorithm 1. In Line 5, we rescale the loss reduction term and the $\xi \boldsymbol{c}(\theta_t)^{1/\alpha} \odot \boldsymbol{w}_t^{-1/\alpha}$ term by their respective mean magnitudes. Such a normalization strategy facilitates a certain level of magnitude alignment, and mitigate the influence of hand tuned parameters on the weight assignment. The term $\gamma \log \boldsymbol{w}_t$ acts as a decay mechanism on $\boldsymbol{w}$ inspired by (Liu et al., 2024). Line 6 presents an equivalent analytical form of the mirror descent step.

5 Experimental Results

5.1 Performance Evaluation of ROWA

In this section, we first empirically validate the effectiveness of our ROWA method. We focus on four benchmarks commonly used in multi-task optimization literature, which are varying in both the number of tasks and types: **CelebA** (Liu et al., 2015) contains over 200K celebrity images, with 40 attribute annotations per image. Using such a 40-task classification benchmarkdemonstrates the scalability of our ROWA method across a large number of tasks. **QM9** (Ramakrishnan et al., 2014) is a graph-based molecular dataset, with an 11-task regression benchmark for predicting 11 properties under diverse measurement scales. It can evaluate an algorithm's ability to balance the inherent variations across these tasks. **NYU-v2** (Silberman et al., 2012) consists of 1,449 indoor-scene images and presents a 3-task prediction problem. **Cityscapes** (Cordts et al., 2016) comprises 5,000 street-scene images used to evaluate 2 pixel-level tasks.

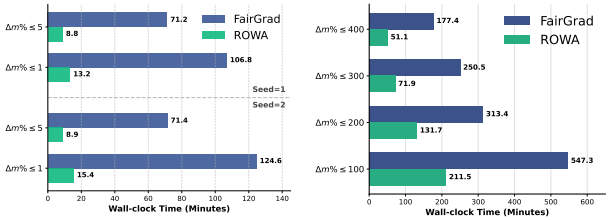
1) Baselines. We compare ROWA with a single-task learning baseline (STL) and 8 loss-oriented methods, as well as 5 representatively gradient-oriented methods. STL trains an independent deep

Table 1: Results on CelebA (40-task), Cityscapes (2-task), and NYU-v2 (3-task) datasets. Each experiment is repeated 3 times with random seeds and the average is reported. The best scores are reported in bold and the second best are underlined.

METHOD	CELEBA	CITYSCAPES		NYU-V2	
	$\Delta m\% \downarrow$	MR $\downarrow$	$\Delta m\% \downarrow$	MR $\downarrow$	$\Delta m\% \downarrow$
MGDA	14.85	12.25	44.14	8.33	1.38
PCGRAD	3.17	7.63	18.29	10.11	3.97
CAGRAD	2.48	6.88	11.64	6.61	0.20
NASH-MTL	2.84	**3.50**	6.82	**3.94**	-4.04
FAIRGRAD	0.37	7.00	3.90	4.44	**-4.96**
LS	4.15	7.25	22.60	10.11	5.59
SI	7.20	11.75	14.11	9.44	4.39
RLW	1.46	10.00	24.38	13.22	7.78
DWA	3.20	7.00	21.45	9.00	3.57
UW	3.23	8.50	5.89	9.78	4.05
FAMO	-1.00	<u>5.75</u>	1.53	<u>4.33</u>	<u>-4.72</u>
GO4ALIGN	1.21	8.00	2.23	5.33	-4.25
LDC-MTL	<u>-1.67</u>	**3.50**	<u>0.19</u>	5.00	-2.90
ROWA	**-1.99**	6.00	**-0.11**	5.33	-4.21

Table 2: Results on QM9 (11-task). Each experiment is repeated 3 times with 120 batch size. The best scores are reported in bold.

METHOD	MR $\downarrow$	$\Delta m\% \downarrow$
MGDA	9.82	120.5
PCGRAD	7.59	125.7
CAGRAD	8.64	112.8
NASH-MTL	4.82	62.0
FAIRGRAD	5.09	57.9
LS	9.55	177.6
SI	6.18	77.8
RLW	11.09	203.8
DWA	9.32	175.3
UW	7.95	108.0
GO4ALIGN	8.36	139.2
FAMO	7.23	64.8
LDC-MTL	4.95	55.6
ROWA	**4.41**	**52.0**



(a) CelebA (40-task) (b) QM9 (11-task)

Figure 4: Training time comparison between ROWA and FairGrad to reach the target $\Delta m\%$ on CelebA and QM9 dataset under the same seed.

Table 3: Results on QM9 (11-task) dataset. Each experiment is repeated 3 times with 60 batch size and different random seeds. The average is reported and the best score is reported in bold.

METHOD	MR $\downarrow$	$\Delta m\% \downarrow$
GO4ALIGN	3.36	124.7
FAMO	2.45	35.8
LDC-MTL	3.36	46.1
ROWA	**2.00**	**32.5**

network for each task. The loss-oriented methods include LS, Scale-invariant (SI), UW (Kendall et al., 2018), DWA (Liu et al., 2019), RLW (Lin et al., 2022), FAMO (Liu et al., 2024), GO4Align (Shen et al., 2024) and LDC-MTL (Xiao et al., 2026). The gradient-oriented methods include MGDA (Désidéri, 2012), PCGrad (Yu et al., 2020a), CAGrad (Liu et al., 2021), NASH-MTL (Navon et al., 2022) and FairGrad (Ban & Ji, 2024). Since this work focuses on the foundational framework, engineered variants of FairGrad are excluded from our main comparison. To ensure a fair comparison, we have reproduced three state-of-the-art and highly competitive loss-oriented methods FAMO, Go4Align, and LDC-MTL strictly according to their public repositories. The results for the remaining algorithms are taken directly from their respective original papers. Detailed experimental setup and results are provided in Appendix B.

2) Evaluations. We report the performance of each individual task, and employ two metrics to quantify the overall performance: **(1)** $\boldsymbol{\Delta m\%}$: the average per-task performance drop of a MTL method m against the STL baseline b: $\Delta m\% = \frac{1}{K}\sum_{i=1}^{K}(-1)^{\delta_k}(M_{m,k} - M_{b,k})/M_{b,k} \times 100$, where $M_{m,k}$ and $M_{b,k}$ denote the performance of the k-th task for m and b respectively. The indicator $\delta_k = 0$ if lower values indicate better performance and otherwise, 1. **(2) Mean Rank**: the average rank of each metric across tasks: $\textbf{MR} = \frac{1}{K}\sum_{k=1}^{K}\text{rank}(M_k)$, where $\text{rank}(M_k)$ denotes the ranking of the k-th metric of the model among all comparison methods. The lower $\Delta m\%$ and **MR** represents the better overall performance.

3) Effectiveness Comparison. We first provide performance comparisons on CelebA, Cityscapes and NYU-v2, in Table 1. Aligning with the metrics adopted in (Ban & Ji, 2024), all methods in the table report the best performance on the CelebA dataset, as well as the average performance over the last 10 epochs on both Cityscapes and NYU-v2 datasets. Detailed results of Cityscapes and NYU-v2

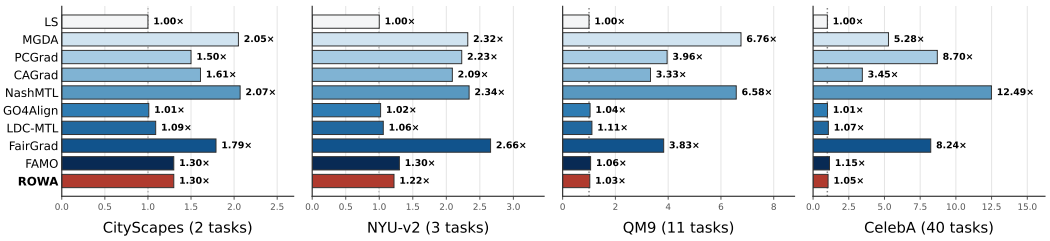


Figure 6: Average training time per epoch for well-performing MTL optimization methods, with LS being the standard time.

are shown in Tables 6 and 7. Collectively, Tables 1 and 6 demonstrate that ROWA attains the best overall MTL performance on the CelebA and Cityscapes datasets, surpassing prior approaches by a large margin. Notably, ROWA stands as the sole method that outperforms STL on average, and yields notable gains on the depth estimation, on Cityscapes. Despite not being the top method on the NYU-v2 dataset as shown in Tables 1 and 7, ROWA still successfully surpasses the -4.0 threshold, remaining highly competitive among all loss-oriented methods.

Results on QM9 are shown in Tables 2 and 3. Aligning with (Ban & Ji, 2024), all methods report the best performance in Table 2 evaluated with batch sizes of 120, and Table 3 evaluated with 60. Detailed results are shown in Tables 8 and 9. Although our reproduced results for Go4Align differ somewhat from the original paper (Shen et al., 2024), their results align closely with results reported in recent literature (Murthy et al., 2026). Table 2 demonstrates that ROWA significantly outperforms all baselines. Table 3 shows that ROWA consistently yields superior results over the three loss-oriented counterparts.

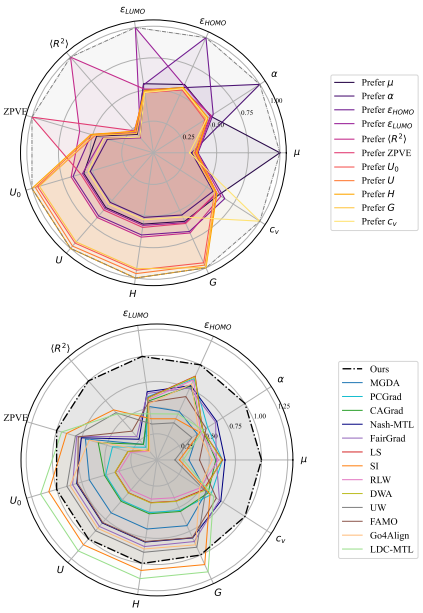


Figure 5: Pareto front exploration of ROWA, on QM9 (11-task) dataset.

4) Training Time Comparison. We conduct an in-depth training time comparison between ROWA and its generic counterpart FairGrad, on CelebA and QM9. Both datasets feature a relatively large number of tasks. Figure 4 reports the wall-clock time required by each algorithm to firstly reach a target $\Delta_m\%$ under identical random seeds. As shown in the figure, ROWA significantly accelerates the training process, reaching target $\Delta m\%$ milestones at least 2.3 times faster than FairGrad across all evaluated settings. In addition, we report the average training time per epoch over 5 epochs of ROWA and baseline methods in Figure 6, which demonstrates that our ROWA achieves computational efficiency comparable to LS, where it incurs only a 1.03x and 1.05x relative time cost on QM9 and CelebA, respectively. Furthermore, the training time of ROWA is much lower than that of most gradient-oriented methods, and aligns closely with other loss-oriented methods.

5.2 Ablation Studies

1) Pareto Front Exploration. We further explore the impact of the $\boldsymbol{r}$-preference on ROWA. The experiment is conducted on the QM9 dataset, whose performance of 11 distinct tasks (μ, α, ϵ_{HOMO}, ϵ_{LUMO}, $\Delta\epsilon$, $\langle R^2 \rangle$, ZPVE, U_0, U, H, and G) is measured by its corresponding test loss. We apply a strong preference to each of the 11 tasks by assigning a dominant weight of 0.8 to the targeted task, while uniformly distributing the remaining weight (0.02 each) among the other 10 tasks. Figure 5 shows the aggregation of the losses across 11 distinct preference configurations.

The upper subfigure of Figure 5 illustrates how ROWA converges to preference-guided Pareto front. Notably, we can observe that assigning a higher preference to a specific objective consistently leads to a better test loss for that task, indicated by a further outward projection on the radar chart. The

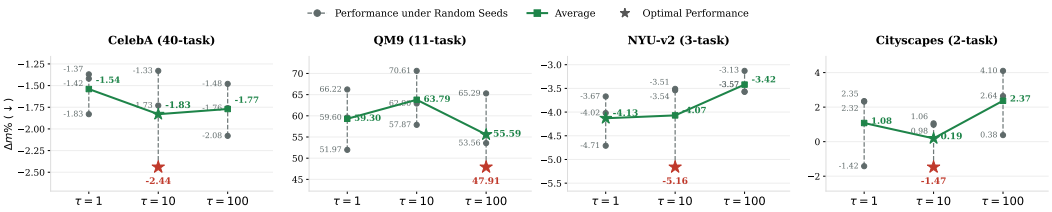
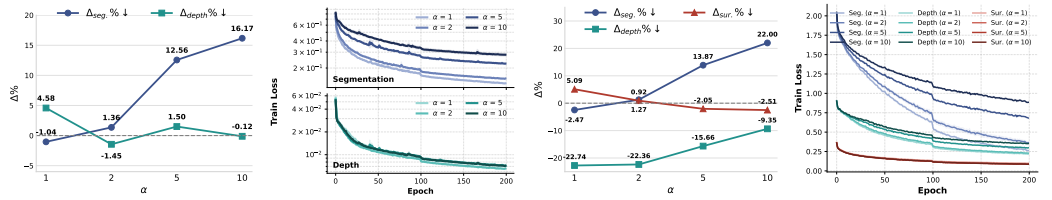


Figure 7: Sensitivity analysis of τ, illustrating the $\Delta m\%(\downarrow)$ variation under equal preferences.



(a) Cityscapes (2-task)

(b) NYU-v2 (3-task)

Figure 8: Impact of α on task-specific performance, over Cityscape and NYU-v2 datasets.

expansive shaded envelope demonstrates that ROWA effectively navigates a vast footprint of the Pareto front. Furthermore, the lower subfigure highlights our exploration capability compared to most baselines, as ROWA achieves more comprehensive and balanced coverage across all tasks.

2) Sensitivity to τ. Figure 7 investigates the sensitivity of the overall performance to the smoothing parameter τ, under the equal preference. It can be observed that, varying τ does not lead to catastrophic performance drops, and the average line remain relatively stable across 4 benchmarks. Moreover, red stars indicate that careful tuning of τ can trigger significant performance breakthroughs, demonstrating its value as a flexible hyperparameter.

3) Ablation on α. We investigate the effect of different α criteria on NYU-v2 and Cityscapes datasets, by setting $\alpha \in \{1, 2, 5, 10\}$. The detailed results are presented in Table 10 and Table 11 in the appendix. Here, we provide Figure 8 to illustrate the impact of adjusting α on the performance metrics of individual tasks. The figure shows the training losses and $\Delta_t\%$ corresponding to task t, where $\Delta_t\%$ represents the averaged $\Delta\%$ of all metrics of task t.

Table 4: Results on Cityscapes (2-task), and NYU-v2 (3-task) datasets across various α.

Method	Cityscapes		NYU-v2	
	MR ↓	$\Delta m\%$ ↓	MR ↓	$\Delta m\%$ ↓
ROWA ($\alpha = 1$)	2.38	1.77	2.78	-2.77
ROWA ($\alpha = 2$)	**1.63**	**-0.11**	2.44	**-4.21**
ROWA ($\alpha = 5$)	3.13	7.03	**2.33**	-1.52
ROWA ($\alpha = 10$)	2.88	7.53	2.44	1.41

As illustrated in Figure 8, Tables 10 and 11, different α lead to variations in both overall and task-specific performance. Increasing α can generally improve the $\Delta\%$ of lower-loss tasks (e.g., Depth in Cityscapes and Surface Normal in NYU-v2). This trend implies that a larger α shifts the weight allocation in favor of these tasks to explicitly compensate for their smaller variation magnitudes. The overall performance results shown in Table 4 indicate that $\alpha = 2$ is the most effective fairness configuration for balancing the performance of various tasks.

6 Conclusion

In this work, we introduced ROWA, a low-complexity generic online weight adaptation framework for MTL, which only uses $\mathcal{O}(1)$ gradient computation time, accommodates various α criteria, and allows a broad exploration of user-preferred Pareto front. Theoretically, we proved that ROWA convergences to a Pareto stationary solution with $\mathcal{O}(1/\sqrt{T})$ iteration complexity. Empirically, we demonstrated that ROWA achieves competitive performance against the state-of-the-art baseline MTL methods.

AI USE STATEMENT

We used generative AI tools solely to polish the language. All AI-assisted revisions were reviewed by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

We have not identified any specific ethical concerns that require additional discussion in this work

REPRODUCIBILITY STATEMENT

We may consider making data and code openly accessible when necessary.

REFERENCES

Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. In *International Conference on Machine Learning*, 2024.

Yaran Chen, Dongbin Zhao, Le Lv, and Qichao Zhang. Multi-task learning for dangerous object detection in autonomous driving. *Information Sciences*, 432:559–571, 2018.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

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

Matthias Ehrgott. *Multicriteria optimization*. Springer, 2005.

Heshan Fernando, Han Shen, Miao Liu, Subhajit Chaudhury, Keerthiram Murugesan, and Tianyi Chen. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *International Conference on Learning Representations*, 2023.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870, 2018.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, 2018.

Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth tchebycheff scalarization for multi-objective optimization. In *International Conference on Machine Learning*, 2024.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1871–1880, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.

Michinari Momma, Chaosheng Dong, and Jia Liu. A multi-objective/multi-task learning framework induced by pareto stationarity. In *International Conference on Machine Learning*, pp. 15895–15907, 2022.

Surya Murthy, Kushagra Gupta, Mustafa O. Karabag, David Fridovich-Keil, and ufuk topcu. DiBS-MTL: Transformation-invariant multitask learning with direction oracles, 2026. URL https://openreview.net/forum?id=MxR3uzejdO.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *Proceedings of the International Conference on Machine Learning*, volume 162, pp. 16428–16446, 2022.

Xiaohan Qin, Xiaoxing Wang, and Junchi Yan. Revisiting fairness in multitask learning: A performance-driven approach for variance reduction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20492–20501, 2025a.

Xiaohan Qin, Xiaoxing Wang, and Junchi Yan. Towards consistent multi-task learning: Unlocking the potential of task-specific parameters. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10067–10076, 2025b.

Shuang Qiu, Dake Zhang, Rui Yang, Boxiang Lyu, and Tong Zhang. Traversing pareto optimal policies: Provably efficient multi-objective reinforcement learning. *arXiv preprint arXiv:2407.17466*, 2024.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):1–7, 2014.

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

Jiayi Shen, Qi Wang, Zehao Xiao, Nanne Van Noord, and Marcel Worring. Go4align: Group optimization for multi-task alignment. In *Advances in Neural Information Processing Systems*, volume 37, pp. 111382–111405, 2024.

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

Shagun Sodhani and A. Z. Mtrl - multi task rl algorithms. https://github.com/facebookresearch/mtrl, 2021.

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3614–3633, 2021.

Menghan Wang, Yujie Lin, Guli Lin, Keping Yang, and Xiao-ming Wu. M2grl: A multi-task multi-view graph representation learning framework for web-scale recommender systems. In *Proceedings of International Conference on Knowledge Discovery & Data Mining*, pp. 2349–2358, 2020.

Xiao Wang, Yuying Han, Dazi Li, Fei Zhang, and Min Tang. From gradient volume to shapley fairness: Towards fair multi-task learning. In *International Conference on Learning Representations*, volume 2026, pp. 129987–130040, 2026.

Peiyao Xiao, Hao Ban, and Kaiyi Ji. Direction-oriented multi-objective learning: Simple and provable stochastic algorithms. *Advances in Neural Information Processing Systems*, 36, 2024.

Peiyao Xiao, Chaosheng Dong, Shaofeng Zou, and Kaiyi Ji. Ldc-mtl: Balancing multi-task learning through scalable loss discrepancy control. In *European Conference on Computer Vision*, pp. 201–218, 2026.

Mingjing Xu, Peizhong Ju, Jia Liu, and Haibo Yang. Psmgd: Periodic stochastic multi-gradient descent for fast multi-objective optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21770–21778, 2025.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33: 5824–5836, 2020a.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100, 2020b.

Qi Zhang, Peiyao Xiao, Kaiyi Ji, and Shaofeng Zou. On the convergence of multi-objective optimization under generalized smoothness. *arXiv preprint arXiv:2405.19440*, 2024.

Qi Zhang, Peiyao Xiao, Shaofeng Zou, and Kaiyi Ji. Mgda converges under generalized smoothness, provably. In *International Conference on Learning Representations*, volume 2025, pp. 10425–10456, 2025.

Zhihan Zhang, Wenhao Yu, Mengxia Yu, Zhichun Guo, and Meng Jiang. A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pp. 943–956, 2023.

Zhiyao Zhang, Zhuqing Liu, Xin Zhang, Wen-Yen Chen, Jiyan Yang, and Jia Liu. Multi-objective bilevel learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28653–28661, 2026a.

Zhiyao Zhang, Myeung Suk Oh, Zhen Qin, Jiaxiang Li, Xin Zhang, and Jia Liu. A tale of two problems: Multi-task bilevel learning meets equality constrained multi-objective optimization. *arXiv preprint arXiv:2605.09094*, 2026b.

Zhipeng Zhou, Liu Liu, Peilin Zhao, and Wei Gong. Injecting imbalance sensitivity for multi-task learning. In *International Joint Conference on Artificial Intelligence*, 2025.