LDC-MTL: Balancing Multi-Task Learning through Scalable Loss Discrepancy Control

Peiyao Xiao[1], Chaosheng Dong[2], Shaofeng Zou[3], and Kaiyi Ji[1]

[1] University at Buffalo, Buffalo, NY 14228, USA
{peiyaoxi,kaiyiji}@buffalo.edu
[2] Amazon.com Inc, Seattle, WA 98109, USA
chaosd@amazon.com
[3] Arizona State University, Tempe, AZ 85281, USA.
zou@asu.edu

Abstract. Multi-task learning (MTL) has been widely adopted for its ability to simultaneously learn multiple tasks. While existing gradient manipulation methods often yield more balanced solutions than simple scalarization-based approaches, they typically incur a significant computational overhead of $\mathcal{O}(K)$ in both time and memory, where K is the number of tasks. In this paper, we propose LDC-MTL, a simple and scalable loss discrepancy control approach for MTL, formulated from a bilevel optimization perspective. Our method incorporates two key components: (i) a bilevel formulation for fine-grained loss discrepancy control, and (ii) a scalable first-order bilevel algorithm that requires only $\mathcal{O}(1)$ time and memory. Theoretically, we prove that LDC-MTL guarantees convergence not only to a stationary point of the bilevel problem with loss discrepancy control but also to an ϵ-accurate Pareto stationary point for all K loss functions under mild conditions. Extensive experiments on diverse multi-task datasets demonstrate the superior performance of LDC-MTL in both accuracy and efficiency. Code is available at https://github.com/OptMN-Lab/LDC-MTL.

Keywords: Multi-task learning · Loss discrepancy control · Bilevel optimization · $\mathcal{O}(1)$ time and memory

1 Introduction

In recent years, Multi-Task Learning (MTL) has received increasing attention for its ability to predict multiple tasks simultaneously using a single model, thereby reducing computational overhead. This versatility has enabled a wide range of applications, including autonomous driving [7], recommendation systems [44], and natural language processing [55].

One of the main challenges in MTL is the imbalance and discrepancy in task losses, where different tasks progress at uneven rates during training. This discrepancy stems from several sources: variations in loss magnitudes due to differing units or scales (e.g., meters vs. millimeters) [19, 28], heterogeneity in task types (e.g., regression vs. classification) [10, 23], and conflicting gradient

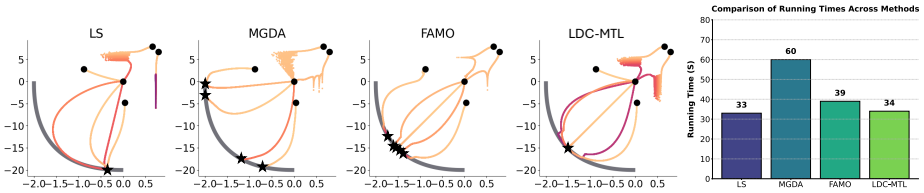

Fig. 1: The loss trajectories of a toy 2-task learning problem from [25] and the runtime comparison of different MTL methods for 50000 steps. Stars on the Pareto front denote the convergence points. Although FAMO [25] achieves more balanced results than Linear Scalarization (LS) and MGDA [11], it converges to different points on the Pareto front. LDC-MTL reaches the same balanced point with a computational cost comparable to the LS. Full experimental details can be found in the Appendix A.3.

directions across tasks [26, 54]. When unaddressed, such discrepancies may cause certain tasks to dominate the optimization trajectory, ultimately leading to degraded performance on others.

To mitigate this issue, MTL research generally follows two main paradigms. The first is the class of scalarization-based methods, which transform MTL into a single-objective optimization problem by aggregating task losses, typically through weighted or averaged sums. Early works adopted static weighting schemes for their simplicity and scalability [4], but these often led to degraded multi-task performance relative to single-task baselines, largely due to persistent gradient conflicts under fixed weights [46]. As a remedy, more recent approaches explore dynamic loss weighting strategies that adapt during training [10, 19, 23, 28]. However, these methods do not explicitly address loss discrepancy, allowing task interference to persist and leading to imbalanced performance across tasks. The second line of work involves gradient manipulation techniques, which aim to promote balanced optimization by explicitly resolving gradient conflicts. These approaches seek update directions that are more equitable across tasks [2, 11, 13, 26, 33, 46, 54]. While often effective in reducing task interference, they typically require computing and storing gradients from all K tasks at each iteration, incurring $O(K)$ time and memory costs. This scalability bottleneck poses challenges for large-scale MTL scenarios involving deep architectures and massive datasets.

In this paper, we propose a simple and scalable loss discrepancy control approach for MTL from a novel bilevel optimization perspective with $\mathcal{O}(1)$ time and memory costs. Our approach comprises two key components: a bilevel formulation for fine-grained loss discrepancy control, and a scalable first-order bilevel algorithmic design. Our specific contributions are summarized as follows.

- **Bilevel formulation for loss discrepancy control.** At the core of our bilevel formulation, the lower-level problem optimizes the model parameters by minimizing a weighted sum of individual loss functions. Meanwhile, the upper-level problem adjusts these weights to minimize the discrepancies among the loss functions, ensuring balanced learning across tasks.

- **Efficient algorithms with $O(1)$ time and memory cost.** We develop Loss Discrepancy Control for Multi-Task Learning (LDC-MTL), a highly efficient algorithm tailored to solve the proposed bilevel problem with loss discrepancy control. Unlike traditional bilevel methods, LDC-MTL has a fully single-loop structure without second-order gradient computations, resulting in an overall $\mathcal{O}(1)$ time and memory complexity. The 2-task toy example in Figure 1 illustrates that our LDC-MTL method achieves a more balanced solution compared to other competitive approaches while maintaining superior computational efficiency.
- **Empirical performance.** Extensive experiments show that our proposed LDC-MTL method outperforms a wide range of scalarization-based and gradient manipulation approaches across multiple supervised multi-task datasets, including QM9 [35], CelebA [29], and Cityscapes [9], while also demonstrating superior efficiency and scalability.
- **Experimental analysis.** We conduct a deeper exploration showing that task losses under LDC-MTL become more concentrated and consistently lower. As a side effect of controlling loss discrepancies, gradient conflicts are also reduced. In addition, when compared to weight-swept linear scalarization (LS), LDC-MTL solutions align with the Pareto frontier (PF) and achieve consistently better results. Our experiments also suggest that the dynamic training trajectory, rather than just the final model parameters, plays a key role in the strong performance of our method. Overall, these results highlight the importance of dynamic loss discrepancy control in LDC-MTL.
- **Theoretical guarantees.** Theoretically, we show that LDC-MTL guarantees convergence not only to a stationary point of the bilevel problem with loss discrepancy control but also to an ϵ-accurate Pareto stationary point for all K individual loss functions under suitable conditions.

2 Related Works

Multi-task learning. MTL has recently garnered significant attention in practical applications. One line of research focuses on model architecture, specifically designing various sharing mechanisms [20, 37]. Another direction addresses the mismatch in loss magnitudes across tasks, proposing methods to balance them. For example, [19] balanced tasks by weighting loss functions based on homoscedastic uncertainties, while [28] dynamically adjusted weights by considering the rate of change in loss values for each task.

Besides, one prominent approach frames MTL as a Multi-Objective Optimization (MOO) problem. [38] introduced this perspective in deep learning, inspiring methods based on the Multi-Gradient Descent Algorithm (MGDA) [11]. Subsequent work has aimed to address gradient conflicts. For instance, PCGrad [54] resolves conflicts by projecting gradients onto the normal plane, GradDrop [8] randomly drops conflicting gradients, and CAGrad [26] constrains update directions to balance gradients. Additionally, Nash-MTL [33] formulates MTL as a bargaining game among tasks, while FairGrad [2] incorporates α-fairness into

gradient adjustments. GO4Align further proposes to group tasks for dynamic progress alignment [41]. [1] introduces a novel gradient aggregation approach using Bayesian inference to reduce the running time. A more recent work, ConsMTL [34], proposes a bilevel formulation that updates shared parameters and task-specific parameters in upper- and lower-level problems, respectively. Though it achieves outstanding performance, most methods do not have a special treatment for the task-specific parameters, and we exclude the ConsMTL in the experiment.

Prior works [21, 48] have shown that several gradient-based MTL methods fail to outperform linear scalarization (LS) with weight sweeping. In contrast, our extensive experiments demonstrate that our method yields solutions that dominate those obtained by other methods in Figure 6. On the theoretical side, [56] analyzed the convergence properties of stochastic MGDA, and [13] proposed a method to reduce bias in the stochastic MGDA with theoretical guarantees. More recent advancements include a double-sampling strategy with provable guarantees introduced by [46] and [6].

Bilevel optimization. Bilevel optimization, first introduced by [3], has been extensively studied over the past few decades. Early research primarily treated it as a constrained optimization problem [17, 42]. More recently, gradient-based methods have gained prominence due to their effectiveness in machine learning applications. Many of these approaches approximate the hypergradient using either linear systems [12, 18] or automatic differentiation techniques [14, 30]. However, these methods become impractical in large-scale settings due to their significant computational cost [47, 51]. The primary challenge lies in the high cost of gradient computation: approximating the Hessian-inverse vector requires multiple first- and second-order gradient evaluations, and the nested sub-loops exacerbate this inefficiency. To address these limitations, recent studies have focused on reducing the computational burden of second-order gradients. For example, some methods reformulate the lower-level problem using value-function-based constraints and solve the corresponding Lagrangian formulation [22, 49]. The work studies convex bilevel problems and proposes a zeroth-order optimization method with finite-time convergence to the Goldstein stationary point [5]. Meanwhile, several works investigate multi-objective bilevel optimization problems, but their computational cost remains high because of the inner iterations or dependence on Hessian terms [52, 53]. In this work, we propose a simplified first-order bilevel method for MTL, motivated by intriguing empirical findings.

3 Preliminary

Scalarization-based methods. MTL aims to optimize multiple tasks (objectives) simultaneously with a single model. The straightforward approach is to optimize a weighted summation of all loss functions: $\min_x L_{total}(x) = \sum_{i=1}^{K} w_i l_i(x)$, where $x \in \mathbb{R}^d$ denotes the model parameter, $l_i(x) : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ represents the loss function of the i-th task and K is the number of tasks. This approach faces three key challenges: 1) loss values could differ in scale, 2) fixed weights can lead to significant gradient conflicts, potentially allowing one task to dominate the

learning process [45, 46]; and 3) the overall performance is highly sensitive to the weighting of different losses [19]. Consequently, such methods often struggle with performance imbalances across tasks.

Gradient manipulation methods. To mitigate gradient conflicts, gradient manipulation methods dynamically compute an update d^t at each epoch to balance progress across tasks, where t is the epoch index. The update d^t is typically a convex combination of task gradients, expressed as:

$$d^t = G(x^t)w^t, \text{ where } w^t = h(G(x^t)),$$

with $G(x^t) = [\nabla l_1(x^t), \nabla l_2(x^t), \ldots, \nabla l_K(x^t)]^\top$. The weight vector w^t is determined by a function $h(\cdot) : \mathbb{R}^{K\times d} \to \mathbb{R}^K$, which varies depending on the specific method. However, these methods often require computing and storing the gradients of all K tasks during each epoch, making them less scalable and resource-intensive, particularly in large-scale scenarios. Therefore, it is necessary to develop lightweight methods that achieve balanced performance.

Pareto concepts. Solving the MTL problem is challenging because it is difficult to identify a common x that achieves the optima for all tasks. Instead, a widely accepted target is finding a Pareto stationary point. Suppose we have two points x_1 and x_2. It is claimed that x_1 dominates x_2 if $l_i(x_1) \leq l_i(x_2)\ \forall i \in [K]$, and $\exists j\ l_j(x_1) < l_j(x_2)$. A point is Pareto optimal if it is not dominated by any other point, implying that no task can be improved further without sacrificing another. Besides, a point x is a Pareto stationary point if $\min_{w\in\mathcal{W}} \|G(x)w\| = 0$.

4 Loss Discrepancy Control for Multi-Task Learning

In this section, we present our bilevel loss discrepancy control framework for multi-task learning. As illustrated in Figure 2, this framework contains a fine-grained bilevel loss discrepancy control procedure and a simplified first-order optimization pipeline. We first introduce the motivation and high-level idea that guide the overall design before moving to the technical details.

4.1 Motivation and High-Level Idea

Loss discrepancy commonly arises in MTL for several reasons. First, loss functions may operate on different scales due to task heterogeneity (e.g., classification vs. regression, as in QM9 [35]) or because they are measured in varying units (e.g., meters, centimeters, or millimeters; [19]). Second, losses can evolve at different rates depending on task difficulty [43]. As a result, although all task losses are expected to converge toward zero, their convergence speeds are often imbalanced when treated with equal importance.

To mitigate this issue, some prior approaches, e.g., [43], introduce a ranking-based strategy that optimizes only a subset of tasks with the largest current losses. While effective in reducing discrepancy, such methods require solving an additional non-smooth optimization problem, which limits scalability. Moreover,

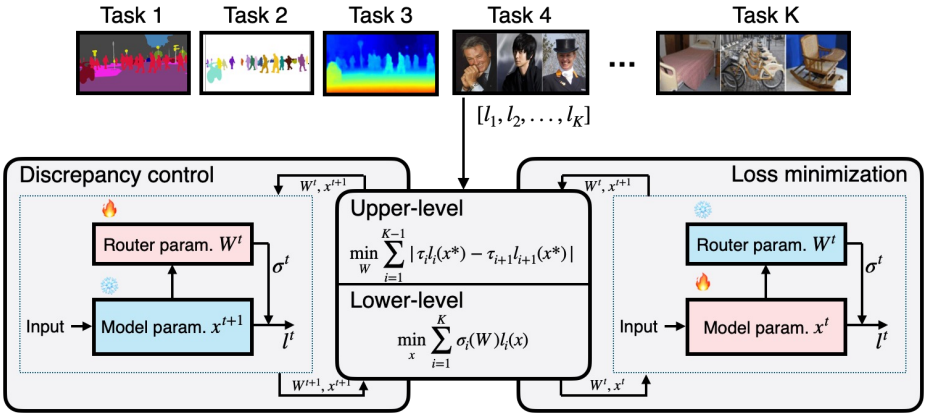

Fig. 2: Our bilevel loss discrepancy control pipeline for multi-task learning. First, different task losses will be computed. Then, the lower-level problem optimizes the model parameter x^t by minimizing the weighted sum of task losses, and the upper-level problem optimizes the router model parameter W^t for loss discrepancy control.

[36] proposes to reduce loss gaps across different groups to enhance model fairness. In preference-based MOO, the goal is to identify a Pareto-optimal point such that $p_1 l_1 = p_2 l_2 = \cdots = p_K l_K$, where $p_1, \ldots, p_K$ denote user-specified preferences. In balanced MTL, these preferences are naturally equal, reducing the goal to aligning task losses so that they remain close to one another.

Motivated by this perspective, we explore a natural question: *can loss discrepancy in MTL be addressed by explicitly reducing the mutual gaps among task losses, while simultaneously ensuring that the solutions remain sufficiently close to the Pareto-stationary set?* In the next section, we provide a positive solution through a bilevel optimization formulation.

4.2 Bilevel formulation for loss discrepancy control

Building on this motivation, we propose a bilevel optimization-based approach, where the upper-level problem aims to mitigate the loss discrepancy, while the lower-level problem ensures that the solution remains within the Pareto-stationary set. Specifically, we consider the following formulation:

$$\begin{aligned} &\min_W \sum_{i=1}^{K-1} |\tau_i l_i(x^*) - \tau_{i+1} l_{i+1}(x^*)| := f(W, x^*) \\ &\text{s.t. } x^* \in \arg\min_x \sum_{i=1}^{K} \sigma_i(W) l_i(x) := g(W, x), \end{aligned} \tag{1}$$

where we denote $x^* = x^*(W)$ for notational convenience. Note that we define a routing function $\sigma(W) \in \mathbb{R}^K$, which is parameterized by a small neural network

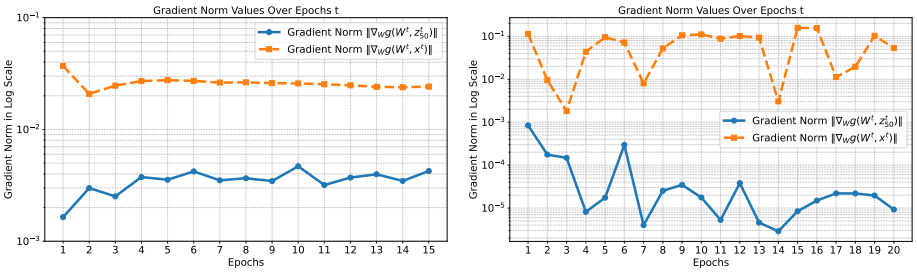

Fig. 3: Gradient norm values during the training process on the CelebA(left) and Cityscapes(right) datasets. Similar phenomena have also been observed in other datasets.

with a softmax output layer. It takes the shared feature as the input and the output weights K for different tasks. For the upper-level weight vector $\tau = (\tau_1, ..., \tau_K)$, it controls the loss discrepancy, and we provide two effective options: (i) $\tau = \sigma(W)$ and (ii) $\tau = \mathbf{1}$ that work well in experiments. It can be seen from Eq. (1) that the lower-level problem minimizes the weighted sum of losses w.r.t. the model parameters x, while the upper-level problem minimizes the accumulated weighted loss gaps w.r.t. the parameters W, controlling the loss discrepancy among tasks. Notably, the optimal solution does not require all losses to be equal, and there will be a **trade-off** between loss minimization and loss discrepancy control, as shown in Remark 1.

4.3 Scalable first-order algorithm design

To enable large-scale applications, we adopt an efficient first-order method to solve the problem in Eq. (1). Inspired by recent advances in first-order bilevel optimization [22, 50], we reformulate it into an equivalent constrained optimization problem as follows.

$$\min_{W,x} f(W,x) \;\; \text{s.t.} \underbrace{\sum_{i=1}^{K} \sigma_i(W) l_i(x) - \sum_{i=1}^{K} \sigma_i(W) l_i(x^*) \leq 0.}_{\text{penalty function } p(W,x)}$$

Then, given a penalty constant $\lambda > 0$, penalizing $p(W,x)$ into the upper-level loss function yields

$$\min_{W,x} f(W,x) + \lambda \sum_{i=1}^{K} \Big(\sigma_i(W) l_i(x) - \sigma_i(W) l_i(x^*) \Big). \tag{2}$$

Intuitively, a larger λ allows more precise training on model parameters x such that x converges closer to x^*. Conversely, a smaller λ prioritizes upper-level loss discrepancy control during training. The main challenge of solving the penalized problem above lies in the updates of W, as shown below:

$$W^{t+1} = W^t - \alpha \Big(\nabla_W f(W^t, x^t) + \lambda \big(\nabla_W g(W^t, x^t) - \nabla_W g(W^t, z_N^t) \big) \Big), \tag{3}$$

Algorithm 1 LDC-MTL

Initialize: W^0, x^0
for $t = 0, 1, \ldots, T-1$ **do**
  $x^{t+1} \leftarrow x^t - \alpha\Big(\nabla_x f(W^t, x^t) + \lambda \nabla_x g(W^t, x^t)\Big)$
  $W^{t+1} \leftarrow W^t - \alpha\Big(\nabla_W f(W^t, x^t) + \lambda \nabla_W g(W^t, x^t)\Big)$
end for

where t is the epoch index, α is the step size, and z_N^t is an approximation of $x_t^* \in \arg\min_x g(W^t, x)$ through the following loop of N iterations each epoch with a step size β.

$$z_{n+1}^t = z_n^t - \beta \nabla_z g(W^t, z_n^t), n = 0, 1, ...N-1, \tag{4}$$

where N is typically chosen to be sufficiently large, ensuring that z_N^t closely approximates x_t^* (the full algorithm is provided in Algorithm 2 in the appendix). Consequently, this sub-loop of iterations incurs significant computational overhead, driven by the high dimensionality of z (matching that of the model parameters) and the large value of N.

Remark 1. This formulation encourages task losses to be close but not necessarily equal. The penalty constant λ controls the trade-off between minimizing the sum of weighted losses and reducing the loss discrepancy among tasks.

Scalable Algorithm. Our experiments show that the magnitude of the gradient norm $\|\nabla_W g(W^t, z_N^t)\|$ remains small, typically orders of magnitude smaller than the gradient norm $\|\nabla_W g(W^t, x^t)\|$, which is used to update the outer parameters W. This behavior is illustrated in Figure 3 on the Cityscapes and CelebA datasets. Specifically, we set $N = 50$ during training. On average, the ratio $\|\nabla_W g(W^t, x^t)\|/\|\nabla_W g(W^t, z_N^t)\|$ exceeds 100 on the Cityscapes dataset, despite some fluctuations. Under these conditions, the term $\nabla_W g(W, z_N^t)$ can be safely neglected, thereby eliminating the need for the expensive loop in Eq. (4). This approximation has been effectively utilized in large-scale applications, such as fine-tuning large language models, to reduce memory and computational costs [39]. It also serves as a foundation for our proposed algorithm, Loss Discrepancy Control for Multi-Task Learning (LDC-MTL), described in Algorithm 1. LDC-MTL employs a fully single-loop structure, which requires only a single gradient computation for both variables per epoch, resulting in a $\mathcal{O}(1)$ time and memory cost. In Section 6, we show that our LDC-MTL method attains both an ϵ-accurate stationary point for the bilevel problem in Eq. (1) and an ϵ-accurate Pareto stationary point for the original loss functions under mild conditions.

5 Empirical Results

In this section, we conduct extensive practical experiments under multi-task classification, regression, and mixed settings to demonstrate the effectiveness

Table 1: Results on Cityscapes (2-task) dataset. Each experiment is repeated 3 times with different random seeds, and the average is reported. The best results are highlighted in **bold**, while the second-best results are indicated with underlines. *Following prior works [2, 13, 26, 46], we report the mean values of $\Delta m\%$ for all results in the main text, with standard deviations provided in Appendix A.*

Method	Segmentation		Depth		$\Delta m\% \downarrow$	MR
	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓		
STL	74.01	93.16	0.0125	27.77		
Gradient manipulation methods with $\mathcal{O}(K)$ computational cost						
MGDA [11]	68.84	91.54	0.0309	33.50	44.14	13.50
PCGrad [54]	75.13	93.48	0.0154	42.07	18.29	9.63
GradDrop [8]	75.27	93.53	0.0157	47.54	23.73	8.75
CAGrad [26]	75.16	93.48	0.0141	37.60	11.64	8.13
IMTL-G [27]	75.33	93.49	0.0135	38.41	11.10	6.38
MoCo [13]	<u>75.42</u>	93.55	0.0149	34.19	9.90	5.25
Nash-MTL [33]	75.41	<u>93.66</u>	<u>0.0129</u>	35.02	6.82	<u>4.00</u>
FairGrad [2]	**75.72**	**93.68**	0.0134	32.25	<u>5.18</u>	**2.25**
Scalarization-based methods with $\mathcal{O}(1)$ computational cost						
LS	75.18	93.49	0.0155	46.77	22.60	9.38
SI	70.95	91.73	0.0161	33.83	14.11	12.75
RLW [23]	74.57	93.41	0.0158	47.79	24.38	12.25
DWA [28]	75.24	93.52	0.0160	44.37	21.45	9.25
UW [19]	72.02	92.85	0.0140	30.13	5.89	9.00
FAMO [25]	74.54	93.29	0.0145	32.59	8.13	8.75
GO4Align [41]	72.63	93.03	0.0164	<u>27.58</u>	8.11	10.75
LDC-MTL (ours)	74.53	93.42	**0.0128**	**26.79**	**-0.57**	6.00

of our method. More experimental details can be found in Appendix A. All experiments are conducted on one NVIDIA A6000.

Baselines and evaluation. To demonstrate the effectiveness of our proposed method, we evaluate its performance against a broad range of baseline approaches. The compared methods include scalarization-based algorithms, such as Linear Scalarization (LS), Scale-Invariant (SI), Random Loss Weighting (RLW) [23], Dynamic Weight Average (DWA) [28], Uncertainty Weighting (UW) [19], FAMO [25], STCH [24], and GO4Align [41]. We also benchmark against gradient manipulation methods, including Multi-Gradient Descent Algorithm (MGDA) [11], PCGrad [54], GradDrop [8], CAGrad [26], IMTL-G [27], MoCo [13], Nash-MTL [33], FairGrad [2]. Besides, we reproduce the results in [41] on the NYU-v2 dataset since it is also a bilevel approach, though with a different focus, denoted as GO4Align*. To provide a comprehensive evaluation, we report the performance of each task and employ two additional metrics: $\mathbf{\Delta m}\%$ to quantify overall performance [31] and the mean rank (MR) across all task metrics. Specifically, the $\mathbf{\Delta m}\%$ metric measures the average relative performance drop of a multi-task

Table 2: Results on NYU-v2 (3-task) dataset. Each experiment is repeated 3 times with different random seeds, and the average is reported. GO4Align* denotes reproduced results obtained using the authors' released code with the same hyperparameters.

Method	Segmentation		Depth		Surface Normal					$\Delta m\% \downarrow$	MR↓
	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓	Angle Distance ↓		Within t° ↑				
					Mean	Median	11.25	22.5	30		
STL	38.30	63.76	0.6754	0.2780	25.01	19.21	30.14	57.20	69.15		
Gradient manipulation methods with $\mathcal{O}(K)$ computational cost											
MGDA [11]	30.47	59.90	0.6070	0.2555	24.88	19.45	29.18	56.88	69.36	1.38	9.00
PCGrad [54]	38.06	64.64	0.5550	0.2325	27.41	22.80	23.86	49.83	63.14	3.97	12.78
GradDrop [8]	39.39	65.12	0.5455	0.2279	27.48	22.96	23.38	49.44	62.87	3.58	11.61
CAGrad [26]	39.79	65.49	0.5486	0.2250	26.31	21.58	25.61	52.36	65.58	0.20	7.39
IMTL-G [27]	39.35	65.60	0.5426	0.2256	26.02	21.19	26.20	53.13	66.24	-0.76	6.56
MoCo [13]	40.30	**66.07**	0.5575	0.2135	26.67	21.83	25.61	51.78	64.85	0.16	7.11
Nash-MTL [33]	40.13	65.93	**0.5261**	0.2171	25.26	20.08	28.40	55.47	68.15	-4.04	4.22
FairGrad [2]	39.74	66.01	0.5377	0.2236	24.84	19.60	29.26	56.58	69.16	**-4.66**	**3.78**
Scalarization-based methods with $\mathcal{O}(1)$ computational cost											
LS	39.29	65.33	0.5493	0.2263	28.15	23.96	22.09	47.50	61.08	5.59	13.44
SI	38.45	64.27	0.5354	0.2201	27.60	23.37	22.53	48.57	62.32	4.39	12.00
RLW [23]	37.17	63.77	0.5759	0.2410	28.27	24.18	22.26	47.05	60.62	7.78	16.22
DWA [28]	39.11	65.31	0.5510	0.2285	27.61	23.18	24.17	50.18	62.39	3.57	12.44
UW [19]	36.87	63.17	0.5446	0.2260	27.04	22.61	23.54	49.05	63.65	4.05	11.78
FAMO [25]	38.88	64.90	0.5474	0.2194	25.06	19.57	29.21	56.61	68.98	-4.10	6.00
GO4Align* [41]	41.26	65.92	0.5461	0.2279	27.20	22.53	24.19	50.36	63.60	1.94	9.00
STCH [24]	**41.35**	66.07	0.4965	**0.2010**	26.55	21.81	24.84	51.39	64.86	-1.35	5.17
LDC-MTL(ours)	38.04	65.92	0.5402	0.2278	**24.70**	**19.19**	**29.97**	**57.44**	**69.69**	-4.40	4.50

model compared to its corresponding single-task learning (STL). Formally, it is defined as: $\Delta m\% = \frac{1}{K}\sum_{i=1}^{K}(-1)^{\delta_k}(M_{m,k} - M_{b,k})/M_{b,k} \times 100$, where $M_{m,k}$ and $M_{b,k}$ represent the performance of the k-th task for the multi-task model m and single-task model b, respectively. The indicator $\delta_k = 1$ if higher values indicate better performance and 0 otherwise.

5.1 Experimental results

Results on the four benchmark datasets are provided in Table 1, Table 2, Table 3, and Table 7 in the appendix. We observe that LDC-MTL outperforms most existing methods on both the CelebA and QM9 datasets, achieving almost the lowest performance drops of $\Delta m\% = -1.31$ and $\Delta m\% = 49.5$. Detailed results for the QM9 dataset illustrate that it achieves a balanced performance across all tasks. These results highlight the effectiveness of our method in handling a large number of tasks in both classification and regression settings. Meanwhile, it also achieves the lowest performance drop, with $\Delta m\% = -0.57$ on the Cityscapes dataset, while delivering comparable results with FairGrad on the NYU-v2 dataset ($\Delta m\% = -4.40$). Meanwhile, the reproduced GO4Align does not achieve the state-of-the-art performance. These findings highlight the capability of LDC-MTL to handle mixed MTL scenarios.

Efficiency comparison. We compare the running time of well-performing approaches in Figure 4. In particular, our method introduces negligible overhead compared to LS with at most a 1.11× increase, aligning with other $\mathcal{O}(1)$ methods such as GO4Align and FAMO. In contrast, gradient manipulation methods, which take the computational cost $\mathcal{O}(K)$, become significantly slower in many-task scenarios. For example, Nash-MTL requires approximately 12×, and the state-of-the-art ConsMTL requires 11× more training time than LDC-MTL on the CelebA dataset.

5.2 Experimental analysis

Loss discrepancy and gradient conflict. To demonstrate the effectiveness of our bilevel formulation for loss discrepancy control, we conduct a detailed analysis of the loss distribution on the CelebA dataset, comparing LS, FAMO, and GO4Align with our proposed method. As shown in Figure 8 in the appendix and statistics in Table 4, the distribution of all 40 task-specific losses reveals that our approach yields more concentrated and consistently lower values. Moreover, we randomly select 8 out of 40 tasks and check the gradient cosine similarity among them. Figure 5 illustrates the cosine similarities of task gradients after the 15th epoch, which shows that the gradient conflict is mitigated as a side-effect.

Table 3: Results on CelebA (40-task) and QM9 (11-task) datasets.

Method	CelebA		QM9	
	$\Delta m\% \downarrow$	MR	$\Delta m\% \downarrow$	MR
Gradient manipulation methods				
MGDA [11]	14.85	12.03	120.5	10.68
PCGrad [54]	3.17	7.92	125.7	8.55
CAGrad [26]	2.48	7.35	112.8	9.45
IMTL-G [27]	0.84	5.75	77.2	7.95
Nash-MTL [33]	2.84	6.17	62.0	<u>5.27</u>
FairGrad [2]	<u>0.37</u>	<u>5.67</u>	57.9	5.55
Scalarization-based methods				
LS	4.15	7.47	177.6	10.45
SI	7.20	8.97	77.8	7.18
RLW [23]	1.46	6.15	203.8	11.91
DWA [28]	3.20	8.12	175.3	10.14
UW [19]	3.23	6.90	108.0	9.18
FAMO [25]	1.21	5.85	58.5	6.64
GO4Align [41]	0.88	N/A	<u>52.7</u>	6.50
STCH [24]	N/A	N/A	56.9	5.82
LDC-MTL (ours)	**-1.31**	**2.62**	**49.5**	**4.73**

Pareto front Comparison. Prior work has noted that certain multi-task learning methods, such as RLW and PCGrad, do not outperform LS, which can approximate the Pareto front through weight sweeping [21, 48]. To this end, we conduct a careful comparison among our LDC-MTL, GO4Align, FAMO, and LS with weight sweeping on the Cityscapes dataset. From Table 9 in the appendix, even after careful weight sweeping, LS does not outperform our approach. The scatter plot in Figure 6 (a-b) also illustrates that LDC-MTL solutions dominate those solutions of the compared methods.

Table 4: Statistics of loss values for LDC-MTL, GO4Align, FAMO, and LS on the CelebA dataset. Lower mean and std indicate better and more stable performance.

Method	**Mean ↓**	**Std ↓**	**Min**	**Max**
LDC-MTL	**0.189**	**0.133**	0.029	**0.538**
GO4Align	0.238	0.154	**0.026**	0.660
FAMO	0.231	0.141	0.027	0.621
LS	0.287	0.195	0.032	0.729

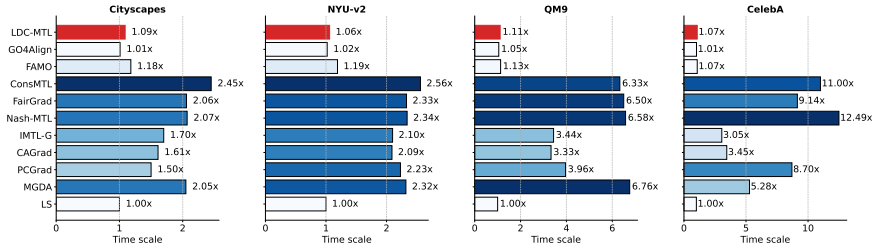

Fig. 4: Time scale comparison among well-performing approaches, with LS considered the reference method for standard time.

Impact of task weight. To further investigate the importance of dynamic weighting, we provide the evolution of task weights for the 11 tasks on the QM9 dataset in Figure 6 (c). It is evident that the task weights adapt meaningfully during the early stages of training and gradually converge later on. Besides, the converged weights are nearly equal, recovering to LS. However, LS does not perform well as shown in Table 7. These findings suggest that, for dynamic-weight methods, the *training trajectory*, not just the final weights, plays a critical role in achieving strong performance.

Table 5: Hyperparameter tuning results of λ on the CelebA dataset.

Method	$\Delta m\% \downarrow$
FairGrad [2]	0.37
LDC-MTL ($\lambda = 0.005$)	-1.27
LDC-MTL ($\lambda = 0.008$)	-1.16
LDC-MTL ($\lambda = 0.01$)	**-1.31**
LDC-MTL ($\lambda = 0.02$)	-0.96

Hyperparameter sensitivity. In our method, hyperparameters include the step size α and the penalty constant λ. For the step size, we adopt the settings from prior experiments without extensive tuning. While we use the same step size for updates to both W and x in our implementation, these can be adjusted independently in practice. For the penalty constant λ, we determine optimal values through a grid search and provide additional experimental results in Table 5 and Table 8 in the appendix.

6 Theoretical Analysis

In this section, we provide convergence analysis for our LDC-MTL method. First, we provide some useful definitions and assumptions.

Definition 1. *Given $L > 0$, a function ℓ is said to be L-Lipschitz-continuous on $\mathcal{X}$ if it holds for any $x, x' \in \mathcal{X}$ that $\|\ell(x) - \ell(x')\| \leq L\|x - x'\|$. A function ℓ is said to be L-Lipschitz-smooth if its gradient is L-Lipschitz-continuous.*

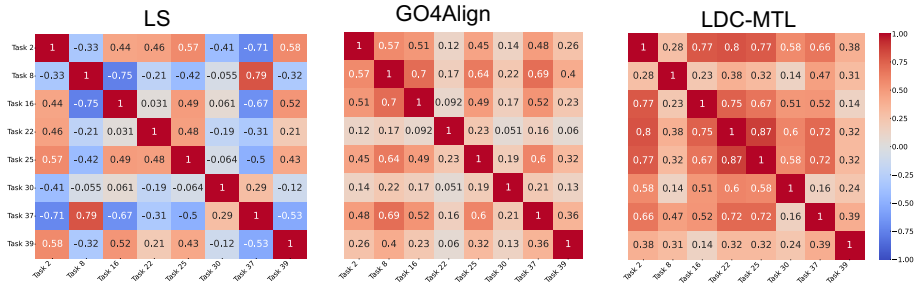

Fig. 5: Cosine similarities of task gradients for LS, GO4Align and LDC-MTL, respectively; LDC-MTL exhibits much higher gradient similarity among tasks, suggesting reduced gradient conflict.

Definition 2 (Pareto stationarity). *We say x is an ϵ-accurate Pareto stationary point for loss functions $\{l_i(x)\}$ if $\min_{w\in\mathcal{W}} \|G(x)w\|^2 = \mathcal{O}(\epsilon)$, where $G(x) = [\nabla l_1(x), \nabla l_2(x), ..., \nabla l_K(x)]^\top$.*

Inspired by [40], we also define the following two surrogates of the original bilevel problem in Eq. (1).

Definition 3. *Define two surrogate bilevel problems as*

$$\mathcal{BP}_\lambda : \min_{W,x} f(W,x) + \lambda(g(W,x) - g(W,x^*)), \tag{5}$$
$$\mathcal{BP}_\epsilon : \min_{W,x} f(W,x) \;\; s.t.\; g(W,x) - g(W,x^*) \le \epsilon,$$

where $\mathcal{BP}_\lambda$ is the penalized bilevel problem, and $\mathcal{BP}_\epsilon$ recovers to the original problem if $\epsilon = 0$.

Assumption 1 (Lipschitz and smoothness) *There exists a constant L such that the upper-level function $f(W,\cdot)$ is L-Lipschitz continuous. There exists constants L_f and L_g such that functions $f(W,x)$ and $g(W,x)$ are L_f- and L_g-Lipschitz-smooth.*

Assumption 2 (Polyak-Lojasiewicz (PL) condition) *The lower-level function $g(W,\cdot)$ satisfies the $\frac{1}{\mu}$-PL condition if there exists a $\mu > 0$ such that given any W, it holds for any feasible x that,*

$$\|\nabla_x g(W,x)\|^2 \ge \frac{1}{\mu}(g(W,x) - g(W,x^*)).$$

Lipschitz continuity and smoothness are standard assumptions in the study of bilevel optimization [16, 18]. While the absolute values in the upper-level function in Eq. (1) are non-smooth, they can be easily modified to ensure smoothness, such as by using a soft absolute value function of the form $y = \sqrt{x^2 + \gamma}$ where γ is a small positive constant. Moreover, the PL condition can be satisfied in over-parameterized neural network settings [15, 32]. The following theorem presents the convergence analysis of our algorithms.

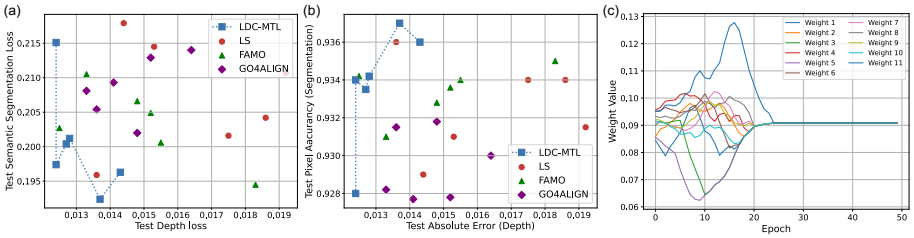

Fig. 6: (a–b) Comparison between LDC-MTL, weight-swept LS, FAMO, and GO4Align with different random seeds on the Cityscapes dataset. In both cases, LDC-MTL solutions lie on the Pareto frontier. (c) Weight change of 11 tasks using LDC-MTL during the training on the QM9 dataset.

Theorem 1. *Suppose Assumptions 1- 2 are satisfied. Select hyperparameters*

$$\alpha \in \Big(0, \frac{1}{L_f + \lambda(2L_g + L_g^2\mu)}\Big], \ \beta \in (0, \frac{1}{L_g}], \lambda = L\sqrt{3\mu\epsilon^{-1}}, \ \textit{and } N = \Omega(\log(\alpha t)).$$

(i) Our method with the updates Eq. (3) *and Eq.* (4) *(i.e., Algorithm 2 in the appendix) finds an ϵ-accurate stationary point of the problem $\mathcal{BP}_\lambda$. If this stationary point is a local/global solution to $\mathcal{BP}_\lambda$, it is also a local/global solution to $\mathcal{BP}_\epsilon$. Furthermore, it is also an ϵ-accurate Pareto stationary point for loss functions $l_i(x), i = 1, ..., K$.*

(ii) Moreover, if $\|\nabla_W g(W^t, z_N^t)\| = \mathcal{O}(\epsilon)$ for $t = 1, ..., T$. The simplified method in Algorithm 1 also achieves the same convergence guarantee as that in (i) with $\mathcal{O}(\epsilon^{-2})$ iterations.

The complete proof is provided in Theorem 2 in the appendix. In the first part of Theorem 1, we establish a connection between the stationarity of $\mathcal{BP}_\lambda$ and Pareto stationarity. Secondly, it introduces an additional gradient vanishing assumption, which has been validated in our experiments. It demonstrates that our simplified LDC-MTL method can also attain an ϵ-accurate stationary point for the problem $\mathcal{BP}_\lambda$ and an ϵ-accurate Pareto stationary point for the original loss functions.

7 Conclusion

We introduced LDC-MTL, a scalable loss discrepancy control approach for multi-task learning based on bilevel optimization. Our method achieves efficient loss discrepancy control with only $\mathcal{O}(1)$ time and memory complexity while guaranteeing convergence to both a stationary point of the bilevel problem and an ϵ-accurate Pareto stationary point for all task loss functions. Extensive experiments demonstrate that LDC-MTL outperforms existing methods in both accuracy and efficiency, highlighting its effectiveness for large-scale MTL. For

future work, we plan to explore the application of our method to broader multi-task learning problems, including recommendation systems.

Acknowledgements

The work of P. Xiao and K. Ji was generously supported by the NSF Career Award under Grant No. 2442418, NSF grants CCF-2311274, ECCS-2326592, and CAREER-2442418. The work of S. Zou was supported in part by DARPA under Agreement No.D25AP00191, and by the National Science Foundation under Grants ECCS-2438392(CAREER) and CCF-2438429.

Bibliography

[1] Achituve, I., Diamant, I., Netzer, A., Chechik, G., Fetaya, E.: Bayesian uncertainty for gradient aggregation in multi-task learning. arXiv preprint arXiv:2402.04005 (2024)

[2] Ban, H., Ji, K.: Fair resource allocation in multi-task learning. arXiv preprint arXiv:2402.15638 (2024)

[3] Bracken, J., McGill, J.T.: Mathematical programs with optimization problems in the constraints. Operations Research **21**(1), 37–44 (1973)

[4] Caruana, R.: Multitask learning. Machine Learning **28**, 41–75 (1997)

[5] Chen, L., Xu, J., Zhang, J.: Bilevel optimization without lower-level strong convexity from the hyper-objective perspective. arXiv preprint arXiv:2301.00712 (2023)

[6] Chen, L., Fernando, H., Ying, Y., Chen, T.: Three-way trade-off in multi-objective learning: Optimization, generalization and conflict-avoidance. Advances in Neural Information Processing Systems **36** (2024)

[7] Chen, Y., Zhao, D., Lv, L., Zhang, Q.: Multi-task learning for dangerous object detection in autonomous driving. Information Sciences **432**, 559–571 (2018)

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

[9] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3223 (2016)

[10] Dai, Y., Fei, N., Lu, Z.: Improvable gap balancing for multi-task learning. In: Uncertainty in Artificial Intelligence. pp. 496–506. PMLR (2023)

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

[12] Domke, J.: Generic methods for optimization-based modeling. In: Artificial Intelligence and Statistics. pp. 318–326. PMLR (2012)

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

[14] Franceschi, L., Donini, M., Frasconi, P., Pontil, M.: Forward and reverse gradient-based hyperparameter optimization. In: International Conference on Machine Learning. pp. 1165–1173. PMLR (2017)

[15] Frei, S., Gu, Q.: Proxy convexity: A unified framework for the analysis of neural networks trained by gradient descent. Advances in Neural Information Processing Systems **34**, 7937–7949 (2021)

[16] Ghadimi, S., Wang, M.: Approximation methods for bilevel programming. arXiv preprint arXiv:1802.02246 (2018)

[17] Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for linear bilevel programming. SIAM Journal on Scientific and Statistical Computing **13**(5), 1194–1217 (1992)
[18] Ji, K., Yang, J., Liang, Y.: Bilevel optimization: Convergence analysis and enhanced design. In: International Conference on Machine Learning. pp. 4882–4892. PMLR (2021)
[19] Kendall, A., Gal, Y., 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. pp. 7482–7491 (2018)
[20] Kokkinos, I.: Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6129–6138 (2017)
[21] Kurin, V., De Palma, A., Kostrikov, I., Whiteson, S., Mudigonda, P.K.: In defense of the unitary scalarization for deep multi-task learning. Advances in Neural Information Processing Systems **35**, 12169–12183 (2022)
[22] Kwon, J., Kwon, D., Wright, S., Nowak, R.D.: A fully first-order method for stochastic bilevel optimization. In: International Conference on Machine Learning. pp. 18083–18113. PMLR (2023)
[23] Lin, B., Ye, F., Zhang, Y., Tsang, I.W.: Reasonable effectiveness of random weighting: A litmus test for multi-task learning. arXiv preprint arXiv:2111.10603 (2021)
[24] Lin, X., Zhang, X., Yang, Z., Liu, F., Wang, Z., Zhang, Q.: Smooth tchebycheff scalarization for multi-objective optimization. arXiv preprint arXiv:2402.19078 (2024)
[25] Liu, B., Feng, Y., Stone, P., Liu, Q.: Famo: Fast adaptive multitask optimization. Advances in Neural Information Processing Systems **36** (2024)
[26] Liu, B., Liu, X., Jin, X., Stone, P., Liu, Q.: Conflict-averse gradient descent for multi-task learning. Advances in Neural Information Processing Systems **34**, 18878–18890 (2021)
[27] Liu, L., Li, Y., Kuang, Z., Xue, J., Chen, Y., Yang, W., Liao, Q., Zhang, W.: Towards impartial multi-task learning. In: Proceedings of the International Conference on Learning Representations (ICLR) 2021 (2021)
[28] Liu, S., Johns, E., Davison, A.J.: 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)
[29] Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3730–3738 (2015)
[30] Maclaurin, D., Duvenaud, D., Adams, R.: Gradient-based hyperparameter optimization through reversible learning. In: International Conference on Machine Learning. pp. 2113–2122. PMLR (2015)
[31] Maninis, K.K., Radosavovic, I., Kokkinos, I.: Attentive single-tasking of multiple tasks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1851–1860 (2019)

[32] Mei, J., Xiao, C., Szepesvari, C., Schuurmans, D.: On the global convergence rates of softmax policy gradient methods. In: International Conference on Machine Learning. pp. 6820–6829. PMLR (2020)
[33] Navon, A., Shamsian, A., Achituve, I., Maron, H., Kawaguchi, K., Chechik, G., Fetaya, E.: Multi-task learning as a bargaining game. arXiv preprint arXiv:2202.01017 (2022)
[34] Qin, X., Wang, X., Yan, J.: 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 (2025)
[35] Ramakrishnan, R., Dral, P.O., Rupp, M., Von Lilienfeld, O.A.: Quantum chemistry structures and properties of 134 kilo molecules. Scientific Data **1**(1), 1–7 (2014)
[36] Roh, Y., Lee, K., Whang, S.E., Suh, C.: Fairbatch: Batch selection for model fairness. arXiv preprint arXiv:2012.01696 (2020)
[37] Ruder, S., Bingel, J., Augenstein, I., Søgaard, A.: Latent multi-task architecture learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4822–4829 (2019)
[38] Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. Advances in Neural Information Processing Systems **31** (2018)
[39] Shen, H., Chen, P.Y., Das, P., Chen, T.: Seal: Safety-enhanced aligned llm fine-tuning via bilevel data selection. arXiv preprint arXiv:2410.07471 (2024)
[40] Shen, H., Chen, T.: On penalty-based bilevel gradient descent method. In: International Conference on Machine Learning. pp. 30992–31015. PMLR (2023)
[41] Shen, J., Wang, C., Xiao, Z., Van Noord, N., Worring, M.: Go4align: Group optimization for multi-task alignment. arXiv preprint arXiv:2404.06486 (2024)
[42] Shi, C., Lu, J., Zhang, G.: An extended kuhn–tucker approach for linear bilevel programming. Applied Mathematics and Computation **162**(1), 51–63 (2005)
[43] Si, Z., Hu, S., Ji, K., Lyu, S.: Meta-learning with heterogeneous tasks. In: International Joint Conference on Artificial Intelligence. pp. 74–94. Springer (2025)
[44] Wang, M., Lin, Y., Lin, G., Yang, K., Wu, X.m.: M2grl: A multi-task multi-view graph representation learning framework for web-scale recommender systems. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2349–2358 (2020)
[45] Wang, Y., Xiao, P., Ban, H., Ji, K., Zou, S.: Finite-time analysis for conflict-avoidant multi-task reinforcement learning. arXiv preprint arXiv:2405.16077 (2024)
[46] Xiao, P., Ban, H., Ji, K.: Direction-oriented multi-objective learning: Simple and provable stochastic algorithms. Advances in Neural Information Processing Systems **36** (2024)
[47] Xiao, P., Ji, K.: Communication-efficient federated hypergradient computation via aggregated iterative differentiation. In: International Conference on Machine Learning. pp. 38059–38086. PMLR (2023)

[48] Xin, D., Ghorbani, B., Gilmer, J., Garg, A., Firat, O.: Do current multi-task optimization methods in deep learning even help? Advances in neural information processing systems **35**, 13597–13609 (2022)
[49] Yang, Y., Ban, H., Huang, M., Ma, S., Ji, K.: Tuning-free bilevel optimization: New algorithms and convergence analysis. arXiv preprint arXiv:2410.05140 (2024)
[50] Yang, Y., Xiao, P., Ji, K.: Achieving $\mathcal{O}(\epsilon^{-1.5})$ complexity in hessian/jacobian-free stochastic bilevel optimization. arXiv preprint arXiv:2312.03807 (2023)
[51] Yang, Y., Xiao, P., Ji, K.: Simfbo: Towards simple, flexible and communication-efficient federated bilevel learning. Advances in Neural Information Processing Systems **36** (2024)
[52] Ye, F., Lin, B., Cao, X., Zhang, Y., Tsang, I.: A first-order multi-gradient algorithm for multi-objective bi-level optimization. arXiv preprint arXiv:2401.09257 (2024)
[53] Ye, F., Lin, B., Yue, Z., Guo, P., Xiao, Q., Zhang, Y.: Multi-objective meta learning. Advances in Neural Information Processing Systems **34**, 21338–21351 (2021)
[54] Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient surgery for multi-task learning. Advances in Neural Information Processing Systems **33**, 5824–5836 (2020)
[55] Zhang, Z., Yu, W., Yu, M., Guo, Z., Jiang, M.: A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods. arXiv preprint arXiv:2204.03508 (2022)
[56] Zhou, S., Zhang, W., Jiang, J., Zhong, W., Gu, J., Zhu, W.: On the convergence of stochastic multi-objective gradient manipulation and beyond. Advances in Neural Information Processing Systems **35**, 38103–38115 (2022)