Improving Gradient Trade-offs between Tasks in Multi-task Text Classification

Heyan Chai[1], **Jinhao Cui**[1], **Ye Wang**[2], **Min Zhang**[1], **Binxing Fang**[1,3] and **Qing Liao**[1,3*]
[1] Harbin Institute of Technology, Shenzhen, China
[2] National University of Defense Technology, China
[3] Peng Cheng Laboratory, Shenzhen, China
{chaiheyan,cuijinhao}@stu.hit.edu.cn, ye.wang@nudt.edu.cn
zhangmin2021@hit.edu.cn, fangbx@cae.cn, liaoqing@hit.edu.cn

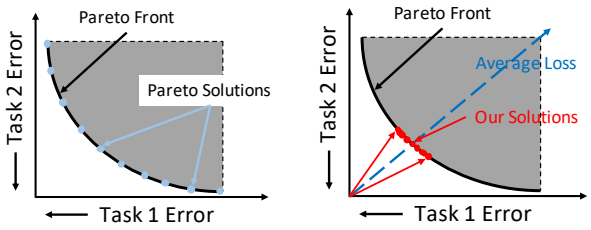
Abstract

Multi-task learning (MTL) has emerged as a promising approach for sharing inductive bias across multiple tasks to enable more efficient learning in text classification. However, training all tasks simultaneously often yields degraded performance of each task than learning them independently, since different tasks might conflict with each other. Existing MTL methods for alleviating this issue is to leverage heuristics or gradient-based algorithm to achieve an arbitrary Pareto optimal trade-off among different tasks. In this paper, we present a novel gradient trade-off approach to mitigate the task conflict problem, dubbed GetMTL, which can achieve a specific trade-off among different tasks nearby the main objective of multi-task text classification (MTC), so as to improve the performance of each task simultaneously. The results of extensive experiments on two benchmark datasets back up our theoretical analysis and validate the superiority of our proposed GetMTL.

1 Introduction

Multi-task Learning (MTL), which aims to learn a single model that can tackle multiple correlated but different tasks simultaneously, makes multiple tasks benefit from each other and obtain superior performance over learning each task independently (Caruana, 1997; Ruder, 2017; Liu et al., 2015; Mao et al., 2020). By discovering shared information/structure across the tasks, it has gained attention in many areas of research and industrial communities, such as computer vision (Misra et al., 2016; Gao et al., 2019; Yogamani et al., 2019; Sun et al., 2020) and text classification (Liu et al., 2017; Xiao et al., 2018; Mao et al., 2021, 2022).

However, it is observed in multi-task text classification (MTC) scenarios that some tasks could conflict with each other, which may be reflected via conflicting gradients or dominating gradients (Yu et al., 2020; Vandenhende et al., 2022), leading to the degraded performance of MTL due to poor training. How to make a proper trade-off among jointing different tasks in MTC is a difficult problem. Recently, several methods have been proposed to mitigate gradient conflicts issue via both *loss balance* (linear weighted scalarization) such as homoscedastic uncertainty (Kendall et al., 2018) and task variance regularization (Mao et al., 2021), and *gradient balance* like Pareto optimality (Sener and Koltun, 2018; Mao et al., 2020). Existing methods devote to finding an arbitrary Pareto optimality solution in the Pareto set, which achieve a single arbitrary trade-off among all tasks. However, they can only satisfy the improved performance on part of tasks, not all tasks simultaneously. This means that these methods can not converge to a minimum average loss of all objectives.



(a) Pareto Optimality (b) GetMTL (Ours)

Figure 1: Graphical interpretation of existing Pareto multi-task learning methods for a two-task learning problem. (a) Pareto optimal solutions are arbitrary and uncontrollable. (b) Our GetMTL can find the specific solutions nearby the main objective (Average loss).

To illustrate our idea, we give a two-task learning example shown in Figure 1. As shown in Figure (1a), it is observed that Pareto optimality-based methods can generate a set of Pareto solutions for a given two-task learning problem. However, some of Pareto solutions can increase the *task 1 error* while decreasing *task 2 error*, leading to unsatisfactory overall performance for MTL model. This im-

* Corresponding Author

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2565–2579

plies that not all Pareto solutions always satisfy the goal of mitigating the tasks conflicts in MTL, and thus failing to achieve a better trade-off between tasks. Therefore, it is necessary to find a specific trade-off between tasks that is beyond what only using Pareto optimality can achieve.

To address this issue, inspired by multi-objective optimization (Sener and Koltun, 2018), we argue that a more efficient way to mitigate task conflicts is to find a gradient trade-off between tasks in the neighborhood of the average loss rather than exhaustively searching for a proper solution from the set of Pareto solutions. As shown in Figure 1b, the Pareto solutions nearby the average loss can achieve a better trade-off between *task 1* and *task 2*, leading to better performance on both tasks at the same time. Based on it, in this paper, we propose a novel gradient trade-off multi-task learning approach, named **GetMTL**, to mitigate task conflicts in multi-task text classification. Specifically, the gradients of each task are utilized to derive an update vector that can minimize the conflicts among task gradients in the neighborhood of the average gradient, so as to achieve a better trade-off performance among joint training tasks. In summary, the main contributions of our work are as follows:

- A novel multi-task learning approach based on gradient trade-off between different tasks (GetMTL) is proposed to deal with task conflict in multi-task text classification problems, so as to improve the performance of all tasks simultaneously.
- We give in-depth theoretical proofs and experimental analyses on establishing converge guarantees of our GetMTL.
- We extensively verify the effectiveness of our GetMTL on two real-world text classification datasets, and the results show that our GetMTL performs competitively with a variety of state-of-the-art methods under a different number of task sets.

2 Related Works

Multi-task Learning methods jointly minimize all task losses based on either loss balance methods (Kendall et al., 2018; Chen et al., 2018; Mao et al., 2021, 2022) or gradient balance methods (Sener and Koltun, 2018; Mao et al., 2020). The loss balance methods adaptively adjust the tasks weights during training based on various heuristic approaches, such as task uncertainty quantification (Kendall et al., 2018), gradient normalization (Chen et al., 2018), task difficulty prioritization (Guo et al., 2018), dynamic weight average (Liu et al., 2019), random loss weighting (Lin et al., 2021), task variance regularization (Mao et al., 2021), and meta learning-based approach (Mao et al., 2022). These methods are mostly heuristic and can have unstable performance while ignoring the task conflicts among all tasks, leading to the bad generalization performance of MTL models.

Recently, some gradient balance based methods have been proposed to mitigate task conflicts for improving task performance. For example, Désidéri (2012) leverages multiple-gradient descent algorithm (MGDA) to optimize multiple objectives. Due to the guarantee of convergence to Pareto stationary point, this is an appealing approach. Sener and Koltun (2018) cast the multi-objective problem as a multi-task problem and devote to finding an arbitrary Pareto optimal solution. Mao et al. (2020) propose a novel MTL method based Tchebycheff procedure for achieving Pareto optimal without any convex assumption. However, these methods only consider achieving an arbitrary Pareto optimal solution while it is not the main objective. Unlike these methods, we propose an MTL approach based on multi-objective optimization and seek to find a set of solutions that are Pareto optimality and nearby the main MTC objective $\mathcal{L}_0$.

3 Preliminaries

Consider a multi-task learning problem with T[1] tasks over an input space $\mathcal{X}$ and a collection of task spaces $\{\mathcal{Y}^t\}_{t\in[T]}$, where each task contains a set of i.i.d. training samples $\mathcal{D}_t = \{x_i, y_i^t\}_{i\in[n_t]}$, T is the number of tasks, and n_t is the number of training samples of task t. The goal of MTL is to find parameters $\{\theta^{sh}, \theta^1, ..., \theta^T\}$ of a model $\mathcal{F}$ that can achieve high average performance across all training tasks over $\mathcal{X}$, defined as $\mathcal{F}(\mathcal{X}, \theta^{sh}, \cdots, \theta^t) : \mathcal{X} \rightarrow \mathcal{Y}$, where θ^{sh} denotes the parameters shared between tasks and θ^t denotes the task-specific parameters of task t. In particular, we further consider a parametric task-specific map as $f^t(\cdot, \theta^{sh}, \theta^t) : \mathcal{X} \rightarrow \mathcal{Y}^t$. We also consider task-specific loss functions $\ell_t(\cdot, \cdot) : \mathcal{Y}^t \times \mathcal{Y}^t \rightarrow \mathbb{R}^+$. We also denote the multi-task loss as $\mathcal{L}(\theta) = \sum_i^T \ell_i(\theta)$, and the gradients of each task

[1]For ease of distinction, we denote the transpose of the vector as the superscript T.

as $g_i = \nabla \ell_i(\theta)$ for the particular θ. In this paper, we choose the average loss as main objective of MTC problem, defined as $\mathcal{L}_0(\theta) = \frac{1}{T}\sum_i^T \ell_i(\theta)$.

3.1 MTL as Multi-objective Optimization

MTL can be formulated as a specific case of multiple-objective optimization (MOO), which optimizes a set of potentially conflicting objectives (Sener and Koltun, 2018; Mao et al., 2020). Given objective functions of T tasks, $\ell_1, \ldots, \ell_T$, we formulate the optimization objective of MTL as the vectors of objective values :

$$\min_{\theta^{sh}, \theta^1, \ldots, \theta^T} \Big(\ell(\theta^{sh}, \theta^1), \ldots, \ell(\theta^{sh}, \theta^T)\Big) \quad (1)$$

Since there is no natural linear ordering on vectors, it is not possible to compare solutions and thus no single solution can optimize all objectives simultaneously. In other words, there is no clear optimal value. Alternatively, we can achieve Pareto optimality to obtain different optimal trade-offs among all objectives to solve MOO problem.

Definition 1 (Pareto dominance)**.** *Given two points $\{\theta, \overline{\theta}\}$ in Ω, a point θ Pareto dominates $\overline{\theta}$ ($\theta \preccurlyeq \overline{\theta}$) for MTL if two conditions are satisfied:*

(i) No one strictly prefers $\overline{\theta}$ to θ, that is, $\forall i \in \{1, \ldots, T\}, \ell_i(\theta^{sh}, \theta^i) \leq \ell_i(\overline{\theta}^{sh}, \overline{\theta}^i)$.

(ii) At least one point strictly prefers θ to $\overline{\theta}$, that is, $\exists j \in \{1, ..., T\}, \ell_j(\theta^{sh}, \theta^j) < \ell_j(\overline{\theta}^{sh}, \overline{\theta}^j)$.

Definition 2 (Pareto optimality)**.** *θ^* is a Pareto optimal point and $\ell(\theta^*)$ is a Pareto optimal objective vector if it does not exist $\hat{\theta} \in \Omega$ such that $\hat{\theta} \preccurlyeq \theta^*$. That is, a solution that is not dominated by any other is called Pareto optimal.*

The set of all Pareto optimal solutions is called the Pareto set, and the image of Pareto set in the loss space is called Pareto front (Lin et al., 2019). In this paper, we focus on gradient-based multi-objective optimization to achieve an appropriate Pareto trade-off among all tasks, which can approximate the Pareto front that minimizes the average loss.

3.2 Gradient-based Multi-Objective Optimization

Gradient-based MOO (Sener and Koltun, 2018) aims to find a direction d that we can iteratively find the next solution $\theta^{(t+1)}$ that dominates the previous one $\theta^{(t)}$ ($\ell(\theta^{(t+1)}) \leq \ell(\theta^{(t)})$) by moving against d with step size η, i.e. $\theta^{(t+1)} = \theta^{(t)} - \eta d$. Désidéri (2012); Sener and Koltun (2018) propose to use multiple gradient descent algorithm (MGDA) that converges to a local Pareto optimal by iteratively using the descent direction d, which can be obtained as follows:

$$\begin{aligned} d^* = \arg \min_{d \in \mathbb{R}^m, \alpha \in \mathbb{R}} \alpha + \frac{1}{2}\|d\|^2 \\ s.t.\ \ \nabla \ell_i(\theta^{(t)})^\mathsf{T} d \leq \alpha, \ \ i = 1, ..., T. \end{aligned} \quad (2)$$

where d^* is the direction that can improve all tasks. Essentially, gradient-based MOO methods minimize the loss by combining gradients with adaptive weights, and obtaining an arbitrary Pareto optimality solution, ignoring the true objective (the average loss) (Liu et al., 2021). In this paper, we generalize this method and propose a novel gradient-based approach to achieve a gradient trade-off among tasks for mitigating task conflicts, as well as constrain the solution that can minimize the average loss ($\mathcal{L}_0(\theta)$).

4 Gradient Trade-offs for Multi-task Text Classification

Following most MTL methods, as shown in Figure 2, we employ the hard parameter sharing MTL architecture, which includes f^{sh} parameterized by heavy-weight task-shared parameters θ^{sh} and f^t parameterized by light-weight task-specific parameters θ^t. All tasks take the same shared intermediate feature $z = f^{sh}(x; \theta^{sh})$ as input, and the t-th task-specific network outputs the prediction as $f^t(z; \theta^t)$. Since task-shared parameters θ^{sh} are shared by all tasks, the different tasks may conflict with each other, leading to the degraded performance of MTL model. In this paper, we hypothesize that one of the main reasons for task conflicts arises from gradients from different tasks competing with each other in a way that is detrimental to making progress. We propose a novel gradient-based MOO optimization to find a gradient trade-off among tasks in the neighborhood of the average loss, so as to mitigate task conflicts. Note that, we omit the subscript sh of task-shared parameters θ^{sh} for the ease of notation.

4.1 GetMTL

Given a task i, we define its gradient as $g_i = \nabla \ell_i(\theta)$ via back-propagation from the raw loss ℓ_i, and g_i represents the optimal update direction for task i. However, due to the inconsistency of the

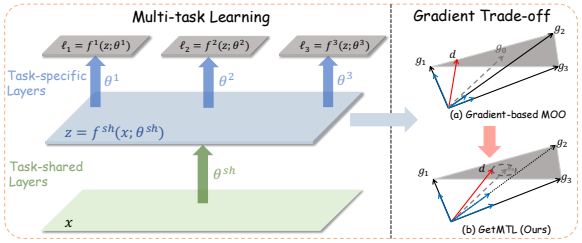


Figure 2: Overview of GetMTL. *Left.* The left part of the figure is our MTL architecture. *Right.* We show the update direction (red) d obtained by gradient-based MOO method and our GetMTL on three gradients (g_1, g_2 and g_3) in $\mathbb{R}^3$, where g_i denotes the gradient (black) of i-th task, g_0 is the average gradient, and blue arrows denote the projections of update direction to each task gradient.

optimal update direction of task-shared parameters for each task, different task gradients may conflict with each other, leading to the training of networks being stuck in the over-training of some tasks and the under-training of other tasks. Intuitively, it is desirable to find a direction that can minimize the task conflicts among different tasks as well as achieve Pareto optimality to improve the performance of MTL model.

We first achieve an arbitrary Pareto optimal via finding a descent direction d_{des} by searching for a minimum-norm point in the *Convex Hull* $\mathcal{CH}$ of gradients, defined by,

$$\mathcal{CH} := \{G\beta \mid \beta \in \mathcal{S}^T\}, \tag{3}$$

$$\text{s.t. } \mathcal{S}^T = \left\{\beta \in \mathbb{R}_+^T \;\middle|\; \sum\nolimits_{j=1}^{T} \beta_j = 1\right\} \tag{4}$$

where $G \in \mathbb{R}^{T\times m} = \{g_1, ..., g_T\}$ is the matrix of task gradient, $\mathcal{S}^T$ is the T-dimensional regular simplex. We use the multiple gradient descent algorithm (MGDA) (Sener and Koltun, 2018) to obtain an arbitrary Pareto optimal by iteratively using the descent direction, defined by,

$$d_{des} = \arg\min_{d\in\mathcal{CH}} \|d\|_2^2 \tag{5}$$

In addition, the d_{des} can be reformulated as a linear combination of all task gradients, defined by,

$$d_{des} = \sum\nolimits_{i=1}^{T} \beta_i g_i \tag{6}$$

where $g_i = \nabla\ell_i(\theta)$ is the i-th task gradient. It implies that, when converges to an arbitrary Pareto optimal, the optimal gradient value of each task via back-propagation is $\beta_i g_i$, defined as $g_{\beta_i} = \beta_i g_i$.

However, moving against d_{des} does not guarantee that the solution meets the requirements of multi-task text classification task (MTC), that is, to alleviate the gradient conflict among tasks in MTC, so as to improve the performance of all tasks. To address this issue, we seek a direction that enables us to move from a solution $\theta^{(t)}$ to $\theta^{(t+1)}$ such that both $\theta^{(t+1)}$ dominates $\theta^{(t)}$ ($\mathcal{L}(\theta^{(t+1)}) \leq \mathcal{L}(\theta^{(t)})$) and alleviate the gradient conflict among all tasks. Based on it, as shown in Figure 2(b), we propose to search for an update direction d in the *Convex Hull* $\mathcal{CH}_\beta$ of back-propagation gradients such that it can improve any worst objective and converge to an optimum of MTC objective $\mathcal{L}_0(\theta)$. We first find the worst task gradient with respect to the update direction d, that is, it has a maximum angle with d, which can be formulated via the following optimization problem,

$$\min_i \langle g_{\beta_i}, d\rangle, \; s.t. - g_{\beta_i}^\mathsf{T} d \leq 0, i = 1, ..., T \tag{7}$$

where g_{β_i} is the i-task gradient after optimizing by MGDA algorithm.

To improve the worst gradient of any task and achieve a trade-off between all task gradients in a neighborhood of the average gradient (defined as $g_0 = \frac{1}{T}\sum_{i=1}^{T} g_i$), we formulate this gradient trade-off optimization problem via the following *Maximin Optimization Problem* (dual problem).

Problem 1.

$$\begin{aligned} \max_{d\in\mathbb{R}^m} \min_{i\in[T]} \; & \langle g_{\beta_i}, d\rangle \\ \text{s.t. } & \|d - g_0\| \leq \varepsilon g_0^\mathsf{T} d, \\ & - g_0^\mathsf{T} d \leq 0 \end{aligned} \tag{8}$$

where $g_{\beta_i} = \beta_i g_i$ is the back-propagation gradient value of i-th task via solving Eq. (5), $\varepsilon \in (0, 1]$ is a hyper-parameter that controls the stability of MTC model.

4.2 Solving Maximin Problem

Since the optimal direction d can also be defined in the convex hull $\mathcal{CH}_\beta$ of g_{β_i}, we can get

$$\mathcal{CH}_\beta := \{G_\beta \boldsymbol{w} \mid \boldsymbol{w} \in \mathcal{W}^T\}, \tag{9}$$

where $G_\beta \in \mathbb{R}^{T\times m} = \{g_{\beta_1}, ..., g_{\beta_T}\}$ is task gradient matrix, $\mathcal{W}^T = \{\boldsymbol{w} \in \mathbb{R}_+^T \mid \sum_{j=1}^{T} w_j = 1\}$ is the T-dimensional probability simplex, and $\boldsymbol{w} = (w_1, ..., w_T)$. Therefore, we can get $\min_i \langle g_{\beta_i}, d\rangle = \min_{w\in\mathcal{W}^T} \langle \sum_i w_i g_{\beta_i}, d\rangle$ and Problem 1 can be transformed into the following form.

Algorithm 1: GetMTL Algorithm.

Input: The number of task T, loss functions $\{\ell_i\}_{i=1}^T$, network parameters $\theta^{(t)}$ at t step, the pre-specified hyper-parameter $\varepsilon \in (0, 1]$ and step size $\mu \in \mathbb{R}^+$.

1: Task Gradients: $g_i = \nabla \ell_i(\theta^{(t)}), i \in [T]$
2: Main Objective: $g_0 = \sum_{i=1}^T g_i$
3: Obtain $\{\beta_1, ...\beta_T\}$ by solving Eq.(5).
4: Compute $g_w = \sum_i w_i g_{\beta_i}$, where $g_{\beta_i} = \beta_i g_i$
5: Obtain $\{w_1, ..., w_T\}$ by solving Eq.(14)
6: Find direction d^* by using Eq.(13)

Output: $\theta^{(t+1)} = \theta^{(t)} - \mu \left(\frac{g_0}{1-\varepsilon^2\|g_0\|^2} + \frac{\varepsilon\|g_0\|^2 g_w}{(1-\varepsilon^2\|g_0\|^2)\|g_w\|} \right)$.

Problem 2.

$$\max_{d\in\mathbb{R}^m} \min_{w\in\mathcal{W}^T} \langle g_w, d\rangle \quad \text{s.t. } \|d - g_0\| \le \varepsilon g_0^\mathsf{T} d, \tag{10}$$

where $g_w = \sum_{i=1}^T w_i g_{\beta_i}$ is the convex combination in $\mathcal{CH}_\beta$. For a given vector $\lambda \in \mathbb{R}^+$ with non-negative components, the corresponding *Lagrangian* associated with the Eq.(10) is defined as

$$\max_{d\in\mathbb{R}^m} \min_{\lambda, w\in\mathcal{W}^T} g_w^\mathsf{T} d - \lambda(\|d - g_0\|^2 - \varepsilon^2 (g_0^\mathsf{T} d)^2)/2 \tag{11}$$

Since the objective for d is concave with linear constraints and $w \in \mathcal{W}^T$ is a compact set [2], according to the Sion's minimax theorem (Kindler, 2005), we can switch the *max* and *min* without changing the solution of Problem 2. Formally,

$$\min_{\lambda, w\in\mathcal{W}^T} \max_{d\in\mathbb{R}^m} g_w^\mathsf{T} d - \lambda\|d - g_0\|^2/2 + \lambda\varepsilon^2 (g_0^\mathsf{T} d)^2/2 \tag{12}$$

We get the optimal solution of primal problem (Problem 1) by solving the dual problem of Eq.(12) (See the Appendix A for a detailed derivation procedure). Then we have

$$d^* = \frac{g_w + \lambda^* g_0}{(1 - \varepsilon^2 g_0^2)\lambda^*}, \text{where} \quad \lambda^* = \frac{\|g_w\|}{\varepsilon\|g_0\|^2} \tag{13}$$

where λ^* is the optimal Lagrange multiplier, d^* is the optimal update direction of MTC model. We can reformulate the problem of Eq.(12) as following optimization problem w.r.t. w.

$$\min_{w\in\mathcal{W}^T} \mathcal{J}(w) = \frac{g_0^\mathsf{T} g_w + \varepsilon\|g_0\|^2\|g_w\|}{1 - \varepsilon^2\|g_0\|^2} \tag{14}$$

[2] Compact set: a set that is bounded and closed.

TASKS	NEWSGROUPS
COMP	GRAPHICS, OS.MS-WINDOWS.MISC, SYS.MAC.HARDWARE, WINDOWS.X
REC	AUTOS, SPORT.BASEBALL, MOTORCYCLES, SPORT.HOCKEY
SCI	CRYPT, SPACE, MED, ELECTRONICS
TALK	POLITICS.MISC, POLITICS.GUNS, POLITICS.MIDEAST, RELIGION.MISC

Table 1: Tasks of topic classification dataset.

where g_w is defined as $g_w = \sum_{i=1}^T w_i g_{\beta_i}$. The detailed derivation is provided in Appendix A. Algorithm 1 shows all the steps of GetMTL algorithm in each iteration.

4.3 Theoretical Analysis

In this section, we analyze the equivalence of solutions to dual problem and then give a theoretical analysis about convergence of GetMTL algorithm. We define the Lagrangian of problem in Eq.(10),

$$L(d, \lambda, w) = g_w^\mathsf{T} d - \frac{\lambda}{2}(\|d - g_0\|^2 - \varepsilon^2 (g_0^\mathsf{T} d)^2)$$

Theorem 4.1 (Equivalence of Optimal Value of Dual Problem)**.** *Assume that both primal problem and dual problem have optimal values, let* $p^* = \max_d \min_{\lambda,w} L(d, \lambda, w)$ *and* $q^* = \min_{\lambda,w} \max_d L(d, \lambda, w)$. *Then,* $p^* = \max_d \min_{\lambda,w} L(d, \lambda, w) \le \min_{\lambda,w} \max_d L(d, \lambda, w) = q^*$.

Proof. The proof is provided in Appendix B. ■

Theorem 4.2 (Convergence of GetMTL)**.** *Assume loss functions* ℓ_i *are convex and differential, and* $\nabla\ell_i(\theta^{(t)})$ *is L-lipschitz continuous with* $L > 0$. *The update rule is* $\theta^{(t+1)} = \theta^{(t)} - \mu^{(t)} d$, *where* d *is defined in Eq.(13) and* $\mu^{(t)} = \min_{i\in[k]} \frac{\|d-g_0\|}{c\cdot L\cdot d^2}$. *All the loss functions* $\left(\ell_1(\theta^{(t)}) \cdots \ell_T(\theta^{(t)})\right)$ *converges to* $(\ell_1(\theta^*) \cdots \ell_T(\theta^*))$.

Proof. The proof is provided in Appendix C. ■

5 Experimental Setup

5.1 Experimental Datasets

We conduct experiments on two MTC benchmarks to evaluate the proposed GetMTL. 1) Amazon Review dataset (Blitzer et al., 2007) contains product reviews from 14 domains (See Details in Appendix D), including apparel, video, books, electronics, DVDs and so on. Each domain gives rise to a binary classification task and we follow Mao et al.

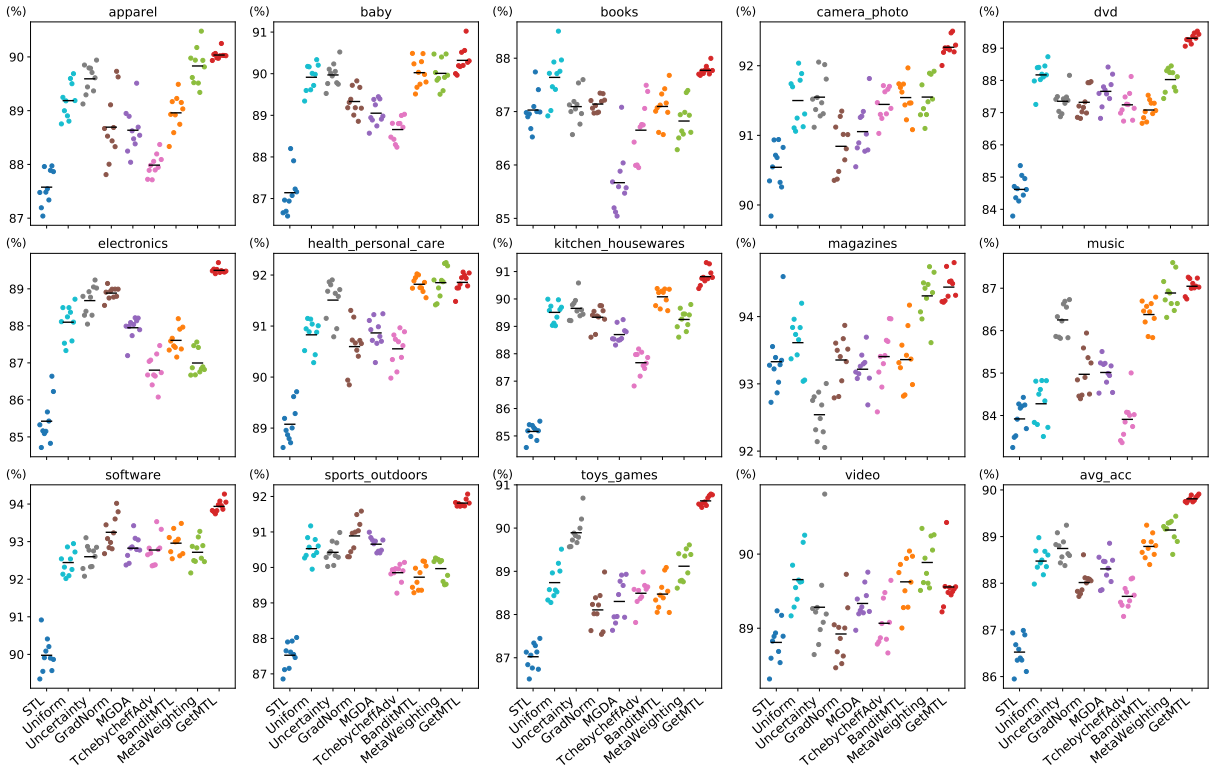


Figure 3: Experimental results on Amazon Review dataset. We plot the classification accuracy of all baselines for all 14 tasks and average performance. Each colored cluster illustrates the classification accuracy performance of a method over 10 runs.

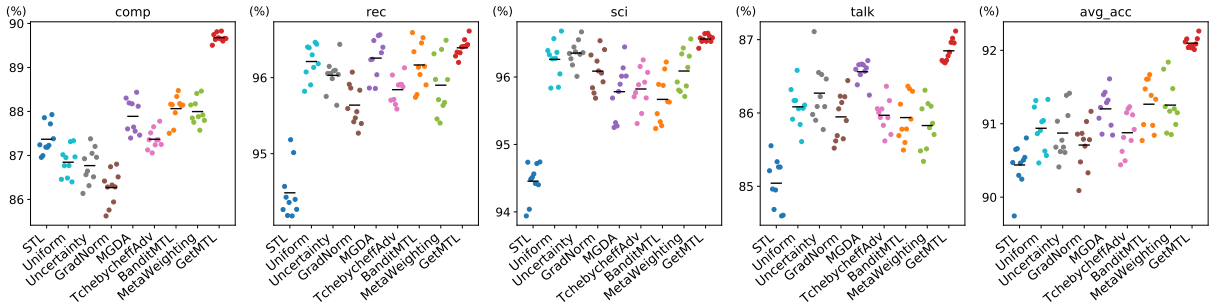


Figure 4: Experimental results on topic classification dataset. We plot classification accuracy of all baselines for all 14 tasks and *avg_acc*. Each colored cluster illustrates classification accuracy of a method over 10 runs.

(2021) to treat 14 domains in the dataset as distinct tasks, creating a dataset with 14 tasks, with 22180 training instances and 5600 test instances in total. 2) Topic classification dataset, 20 Newsgroup[3], consists of approximately 20,000 newsgroup documents, partitioned evenly across 20 different newsgroups. We follow Mao et al. (2021) to select 16 newsgroups from 20 Newsgroup dataset shown in Table 1 and then divide them into four groups. Each group gives rise to a 4-way classification task, creating a dataset with four 4-way classification tasks, which is a more challenging dataset than amazon review dataset.

[3]http://qwone.com/ jason/20Newsgroups/

5.2 Experimental Implementation

We follow the standard MTC setting and adopt the same network architectures with the most recent baselines for fair comparisons (Mao et al., 2021). We adopt the hard parameter sharing MTL framework shown in Figure 2, where task-shared network is a TextCNN with kernel size of 3,5,7 and task-specific network is a fully connected layer with a softmax function. Adam is utilized as the optimizer to train the model over 3000 epochs with a learning rate of 1e-3 for both sentiment analysis and topic classification. We set the batch size to 256.

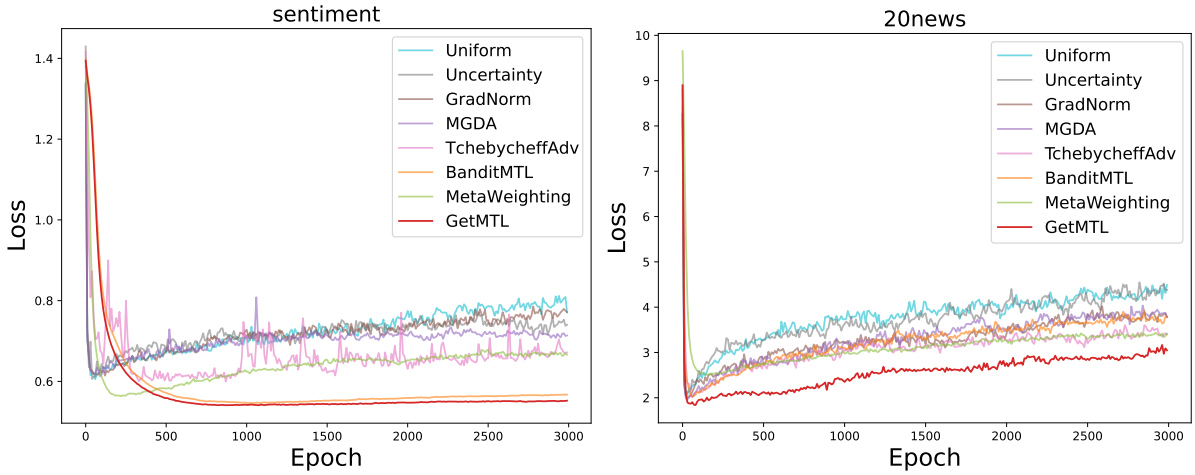


Figure 5: Learning curve of comparison methods in both amazon review and topic classification datasets.

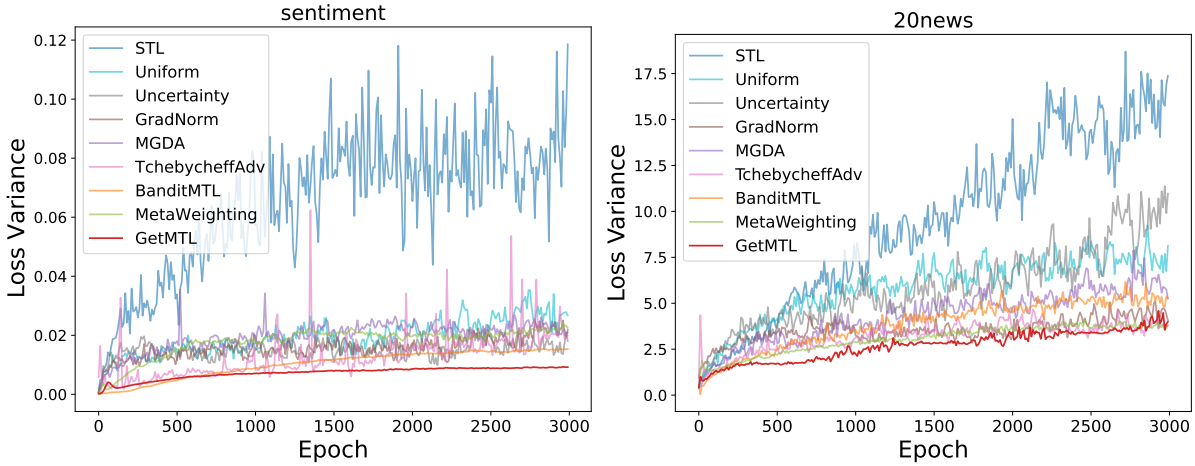


Figure 6: Evolution of task variance during training of baseline methods and GetMTL on the amazon review and topic classification datasets.

5.3 Comparison Models

We compare the proposed GetMTL with a series of MTC baselines, including

Single-Task Learning (STL): learning each task independently.

Uniform Scaling: learning tasks simultaneously with uniform task weights.

Uncertainty: using the uncertainty weighting method (Kendall et al., 2018).

GradNorm: learning tasks simultaneously with gradient normalization method (Chen et al., 2018).

TchebycheffAdv: using adversarial Tchebycheff procedure (Mao et al., 2020).

MGDA: using gradient-based multi-objective optimization method (Sener and Koltun, 2018).

BanditMTL: learning tasks simultaneously with multi-armed bandit method (Mao et al., 2021).

MetaWeighting: using adaptive task weighting method (Mao et al., 2022).

6 Experimental Results

6.1 Main Results

The main comparison results of GetMTL on two benchmark datasets are shown in Figure 3 and 4. It is clear that (See detailed numerical comparison results in Appendix D), our proposed GetMTL model performs consistently better than the all comparison methods on all tasks of both amazon review and topic classification datasets, and its average performance is superior to that of all baselines. This verifies the effectiveness of our GetMTL method in MTC problem. More concretely, in comparison with the gradient-based MOO optimization model (MGDA), our GetMTL achieves significant improvement across all datasets. This indicates that achieving a gradient trade-off nearby average loss to mitigate task conflicts can better improve all task performance and generalization ability of MTC model.

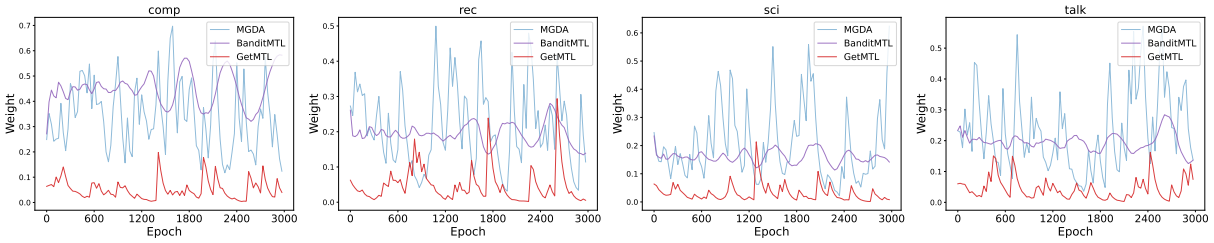
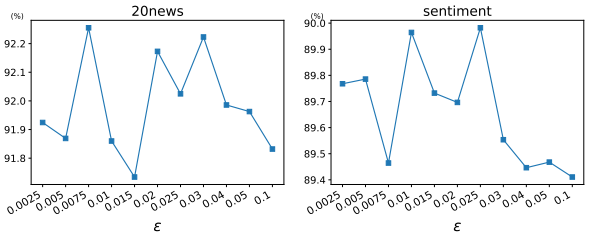


Figure 7: Task weights of comparison methods on four tasks (including comp, rec, sci, and talk tasks) in topic classification dataset. Task weights obtained from MGDA, BanditMTL and GetMTL throughout the optimization process. For better visualization, we plot points every 30 epochs.



(a) Amazon review dataset. (b) Topic classification datset.

Figure 8: Impact of different values of ε.

6.2 Empirical Analysis on Convergence

In Section 4.3, we theoretically prove the convergence of our proposed GetMTL. Furthermore, we conduct extensive experiments about the convergence to better demonstrate the advantages of GetMTL shown in Figure 5. It is clear that the learning curve of GetMTL is constantly decreasing as the number of iterations increases and converges to the lowest loss value compared with other baselines. It indicates that GetMTL can guarantee the convergence of the objective value and obtain better performance of all learning tasks.

In addition, we also conduct extensive experiments to investigate how GetMTL mitigates task conflict during training. We plot the task variance (variance between the task-specific losses) of all baselines on both amazon review and topic classification datasets shown in Figure 6. It can be observed that all MTL baselines have lower task variance than STL method, which illustrates that MTL methods can indeed boost the learning of all tasks compared with STL method. Moreover, GetMTL has the lowest task variance and smoother evolution during training than other MTL baselines. This implies that our proposed GetMTL indeed mitigates task conflicts compared with other MTL methods.

6.3 The Evolution of Task Weight w

In this section, we visualize the task weights of our GetMTL and two weight adaptive MTL methods (MGDA and BanditMTL) throughout the training process using the topic classification dataset shown in Figure 7. It can be observed from these four figures that the weight adaption process of our GetMTL is different from that of MGDA and BanditMTL. GetMTL can automatically learn the task weights without pre-defined heuristic constraints. The weights adaption process of GetMTL is more stable and the search space is more compact compared with other MTL baselines.

6.4 Impact of the Values of ε

To investigate the impact of using different values of ε on the performance of our GetMTL, we conduct experiments on two datasets, and the results are shown in Figure 8. Noting that model with $\varepsilon = 0.0075$ and $\varepsilon = 0.025$ perform overall better than other values on these two datasets, respectively. The model with larger value of ε performs unsatisfactorily overall all tasks on two datasets, one possible reason is that larger ε makes d pull far away from the average loss g_0 (see the conditions in Eq. (9)). That is, Pareto optimality found by GetMTL is getting further and further away from MTC objective $\mathcal{L}_0$, which can be quite detrimental to some tasks' performance, leading to degraded average performance.

7 Conclusion

In this paper, we propose a novel gradient trade-off multi-task learning approach to mitigate the task conflict problem, which can achieve a specific trade-off among different tasks nearby the main objective of multi-task text classification problem. Moreover, we present a series of theoretical proofs to illustrate the effectiveness and superiority of our GetMTL. Experimental results on two benchmark

datasets show that our GetMTL achieves state-of-the-art performance in Multi-task Text Classification problem.

Limitations

Our GetMTL needs to compute the g_i for each task i at each iteration and requires a backward-propagation procedure over the model parameters. Every iteration requires one forward-propagation followed by T backward-propagation procedure and computation of backward-propagation is typically more expensive than the forward-propagation. Here, we define the time of one forward pass and one backward pass as E_f and E_b, respectively. The time of optimization process is defined as E_o. Therefore, the total time E of GetMTL is defined,

$$\begin{aligned} E &= E_f + TE_b + E_o \\ &\approx TE_b + E_o \end{aligned}$$

For few-task learning scenario ($T < 100$), usually $E_o \ll E_b$ and GetMTL still works fine. However, for large-scale task set (like $T \gg 100$), usually $E_o \gg E_b$ or $E_o \gg TE_b$. Consequently, our GetMTL may get stuck in the optimization and backward-propagation process at each iteration. Therefore, the major limitation of our work is that it can not be applied to scenarios with large-scale task sets.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62076079), Guangdong Major Project of Basic and Applied Basic Research (No.2019B030302002), The Major Key Project of PCL(Grant No.PCL2022A03), and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005).

References

Dimitri P Bertsekas. 1997. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics,*. The Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 793–802. PMLR.

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

Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L. Yuille. 2019. NDDR-CNN: layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3205–3214.

Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. 2018. Dynamic task prioritization for multitask learning. In *Proceedings of the European conference on computer vision (ECCV)*, volume 11220 of *Lecture Notes in Computer Science*, pages 282–299. Springer.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 7482–7491. Computer Vision Foundation / IEEE Computer Society.

Jürgen Kindler. 2005. A simple proof of sion's minimax theorem. *The American Mathematical Monthly*, 112(4):356–358.

Baijiong Lin, Feiyang Ye, and Yu Zhang. 2021. A closer look at loss weighting in multi-task learning. *CoRR*, abs/2111.10603.

Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. 2019. Pareto multi-task learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 12037–12047.

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

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1–10. Association for Computational Linguistics.

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

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 912–921. The Association for Computational Linguistics.

Yuren Mao, Zekai Wang, Weiwei Liu, Xuemin Lin, and Wenbin Hu. 2021. Banditmtl: Bandit-based multi-task learning for text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pages 5506–5516. Association for Computational Linguistics.

Yuren Mao, Zekai Wang, Weiwei Liu, Xuemin Lin, and Pengtao Xie. 2022. Metaweighting: Learning to weight tasks in multi-task learning. In *Findings of the Association for Computational Linguistics: ACL*, pages 3436–3448. Association for Computational Linguistics.

Yuren Mao, Shuang Yun, Weiwei Liu, and Bo Du. 2020. Tchebycheff procedure for multi-task text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 4217–4226. Association for Computational Linguistics.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3994–4003.

Yurii Nesterov. 1998. Introductory lectures on convex programming volume i: Basic course. *Lecture notes*, 3(4):5.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 525–536.

Ximeng Sun, Rameswar Panda, Rogério Feris, and Kate Saenko. 2020. Adashare: Learning what to share for efficient deep multi-task learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*.

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2022. Multi-task learning for dense prediction tasks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3614–3633.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. 2020. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *The Journal of Machine Learning Research*, 21(1):9047–9076.

Liqiang Xiao, Honglun Zhang, and Wenqing Chen. 2018. Gated multi-task network for text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 726–731. Association for Computational Linguistics.

Senthil Kumar Yogamani, Christian Witt, Hazem Rashed, Sanjaya Nayak, Saquib Mansoor, Padraig Varley, Xavier Perrotton, Derek O'Dea, Patrick Pérez, Ciarán Hughes, Jonathan Horgan, Ganesh Sistu, Sumanth Chennupati, Michal Uricár, Stefan Milz, Martin Simon, and Karl Amende. 2019. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 9307–9317.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*.

A Derivations of GetMTL Algorithm

Lemma A.1. *Let d^* be the solution of*

$$\max_{d\in\mathbb{R}^m} \min_{i\in[T]} \langle g_{\beta_i}, d\rangle \text{ , s.t. } \|d-g_0\| \le \varepsilon g_0^\mathsf{T} d, \quad (15)$$

where $\varepsilon \in (0,1]$, $\{g_i \in \mathbb{R}^m \mid \forall i \in \{0,1,...,T\}\}$, and $g_{\beta_i} = \beta_i g_i \in \mathbb{R}^m$. Then we have

$$d^* = \left(\frac{g_0}{1-\varepsilon^2\|g_0\|^2} + \frac{\varepsilon\|g_0\|^2 g_{w^*}}{(1-\varepsilon^2\|g_0\|^2)\|g_{w^*}\|}\right), \quad (16)$$

where $g_0 = \frac{1}{T}\sum_{i=1}^T g_i$, and $g_{w^*} = \sum_{i=1}^T w_i^* g_{\beta_i}$. The w^* is the solution of

$$min_{w\in\mathcal{W}^T} \mathcal{J}(w) = \frac{g_0^\mathsf{T} g_w + \varepsilon\|g_0\|^2\|g_w\|}{1-\varepsilon^2\|g_0\|^2}, \quad (17)$$

where $\mathcal{W}^T = \{w \in \mathbb{R}_+^T \mid \sum_{j=1}^T w_j = 1\}$. We have,

$$\min_i g_i^\mathsf{T} d^* = \frac{g_0^\mathsf{T} g_{w^*} + \varepsilon\|g_0\|^2\|g_{w^*}\|}{1-\varepsilon^2\|g_0\|^2}. \quad (18)$$

Proof. We first construct Lagrange function of the objective in Eq.(10),

$$L(d,\lambda,w) = g_w^\mathsf{T} d - \lambda(\|d-g_0\|^2 - \varepsilon^2(g_0^\mathsf{T} d)^2)/2 \quad (19)$$

According the Lagrange duality and Sion's minimax theorem (Kindler, 2005), we can switch the *max* and *min* without changing the solution and then the primal problem can be reformulated as following form,

$$\min_{\lambda,w\in\mathcal{W}^T} \max_{d\in\mathbb{R}^m} g_w^\mathsf{T} d - \lambda(\|d-g_0\|^2 - \varepsilon^2(g_0^\mathsf{T} d)^2)/2 \quad (20)$$

With λ, w fixing, we first solve the max of $L(d,\lambda,w)$ w.r.t. d,

$$\max_d L(d,\lambda,w) = g_w^\mathsf{T} d - \frac{\lambda}{2}(\|d-g_0\|^2 - \varepsilon^2(g_0^\mathsf{T} d)^2) \quad (21)$$

We set the gradient of $L(d,\lambda,w)$ with respect to d equal to zero,

$$\nabla_d L(d,\lambda,w) = g_w - \lambda(d-g_0) + \lambda\varepsilon^2\|g_0\|^2 d = 0, \quad (22)$$

We can get the optimal d^*,

$$d^* = \frac{g_w + \lambda g_0}{(1-\varepsilon^2 g_0^2)\lambda}, \quad (23)$$

and we plug the solution d^* in $L(d,w,\lambda)$ to obtain $\hat{L}(d,\lambda,w)$,

$$\min_{w,\lambda} \hat{L}(\lambda,w) = \frac{(\|g_w\| + \lambda\|g_0\|)^2}{2\lambda(1-\varepsilon^2\|g_0\|^2)} - \frac{\lambda}{2}\|g_0\|^2, \quad (24)$$

Then, we set the gradient of $\hat{L}(\lambda,w)$ with respect to λ equal to zero,

$$\nabla_\lambda \hat{L}(\lambda,w) = -\frac{\|g_w\|^2}{2\lambda^2(1-\varepsilon^2\|g_0\|^2)} - \frac{\|g_0\|^2}{2} + \frac{\|g_0\|^2}{2(1-\varepsilon^2\|g_0\|^2)} = 0 \quad (25)$$

We can get the optimal λ^*,

$$\lambda^* = \frac{\|g_w\|}{\varepsilon\|g_0\|^2}. \quad (26)$$

We then plug the λ^* in d^* to obtain,

$$d^* = \left(\frac{g_0}{1-\varepsilon^2\|g_0\|^2} + \frac{\varepsilon\|g_0\|^2 g_w}{(1-\varepsilon^2\|g_0\|^2)\|g_w\|}\right), \quad (27)$$

Finally, plugging d^* and λ^* into the objective in Eq.(20), we can obtain the following optimization problem $\mathcal{J}(w)$,

$$\min_{w\in\mathcal{W}^T} \mathcal{J}(w) = \frac{g_0^\mathsf{T} g_w + \varepsilon\|g_0\|^2\|g_w\|}{1-\varepsilon^2\|g_0\|^2}, \quad (28)$$

We can obtain w^* by solving following optimization problem $\mathcal{J}(w)$ w.r.t. w, formally,

$$w^* = \arg\min_{w\in\mathcal{W}^T} \mathcal{J}(w) = \frac{g_0^\mathsf{T} g_w + \varepsilon\|g_0\|^2\|g_w\|}{1-\varepsilon^2\|g_0\|^2}, \quad (29)$$

■

B Proof of Theorem 4.1

Following the proof of Lemma A, we use same Lagrangian function in Eq.(19) for simplicity,

$$L(d,w,\lambda) = g_w^\mathsf{T} d - \lambda(\|d-g_0\|^2 - \varepsilon^2(g_0^\mathsf{T} d)^2)/2 \quad (30)$$

Proof. Let $\mathcal{P}_D(\lambda,w) = \max_d L(d,\lambda,w)$ and $\mathcal{P}_P(d) = \min_{\lambda,w} L(d,\lambda,w)$. Then we can get,

$$\min_{\lambda,w} L(d,\lambda,w) \le L(d,\lambda,w) \le \max_d L(d,\lambda,w) \quad (31)$$

Thus, we have,

$$\mathcal{P}_P(d) \le \mathcal{P}_D(\lambda,w) \quad (32)$$

Since both primal problem and dual problem have optimal solutions, we have,

$$\max \mathcal{P}_P(d) \le \min \mathcal{P}_D(\lambda,w) \quad (33)$$

Finally, we get

$$p^* = \max_{d} \min_{\lambda, w} L(d, \lambda, w) \leq \min_{\lambda, w} \max_{d} L(d, \lambda, w) = q^* \tag{34}$$

Since the dual problem is a convex programming and the solutions d^*, λ, and w meet Karush-Kuhn-Tucker (KKT) (Bertsekas, 1997; Désidéri, 2012) conditions, we can get,

$$p^* = q^* = L(d^*, \lambda^*, w^*) \tag{35}$$

That is, the optimal value defined by Eq. (14) is equal to optimal value defined by Eq. (9). Therefore, we can solve complex *Maximin Optimization Problem* in Eq.(9) by solving its dual problem. ■

C Proof of Theorem 4.2

Lemma C.1. *If ℓ is differential and L-smooth, $\nabla\ell$ is L-Lipschitz continuous, then*

$$\ell(\theta') \leq \ell(\theta) + \nabla\ell(\theta)^\mathsf{T}(\theta' - \theta) + \frac{L}{2}\|\theta' - \theta\|^2 \tag{36}$$

Proof. Using the fundamental theorem of calculus with the continuous function $\nabla\ell$, we can get,

$$\begin{aligned}
\ell(\theta') &= \ell(\theta) + \int_0^1 \nabla\ell(\theta + t(\theta' - \theta))^\mathsf{T}(\theta' - \theta)\, dt \\
&= \ell(\theta) + \nabla\ell(\theta)^\mathsf{T}(\theta' - \theta) \\
&+ \int_0^1 (\nabla\ell(\theta + t(\theta' - \theta)) - \nabla\ell(\theta))^\mathsf{T}(\theta' - \theta) dt \\
&\leq \ell(\theta) + \nabla\ell(\theta)^\mathsf{T}(\theta' - \theta) \\
&+ \int_0^1 \|\nabla\ell(\theta + t(\theta' - \theta)) - \nabla\ell(\theta)\|\|\theta' - \theta\| dt \\
&\text{(Using the definition of Lipschitz-continuous)} \\
&\leq \ell(\theta) + \nabla\ell(\theta)^\mathsf{T}(\theta' - \theta) + \int_0^1 tL\|\theta' - \theta\|^2 dt \\
&= \ell(\theta) + \nabla\ell(\theta)^\mathsf{T}(\theta' - \theta) + \frac{L}{2}\|\theta' - \theta\|^2
\end{aligned} \tag{37}$$

■

Proof of Theorem 4.2

Proof. Let $\{\theta^{(t)}\}_{t=1}^{\infty}$ be model parameters sequence generated by using update rule $\theta^{(t+1)} = \theta^{(t)} - \mu^{(t)} d$ where d is defined in Eq.(13). Since all $\nabla\ell_i$ are Lipschitz continuous, for each loss $\{\ell_i\}_{i\in[T]}$, we have using Lemma C.1,

$$\begin{aligned}
\ell_i(\theta^{(t+1)}) &\leq \ell_i(\theta^{(t)}) + \nabla\ell_i(\theta^{(t)})^\mathsf{T}(\theta^{(t+1)} - \theta^{(t)}) \\
&+ \frac{L}{2}\|\theta^{(t+1)} - \theta^{(t)}\|^2 \\
&= \ell_i(\theta^{(t)}) - \mu^{(t)}\nabla\ell_i(\theta^{(t)})^\mathsf{T} d + \frac{L}{2}\|\mu^{(t)} d\|^2 \\
&\text{(Using the constraint} \|d - g_0\| \leq \varepsilon g_0^\mathsf{T} d) \\
&\leq \ell_i(\theta^{(t)}) - \frac{\mu^{(t)}\|d - g_0\|}{\varepsilon} + \frac{(\mu^{(t)})^2}{2} L\|d\|^2 \\
&= \ell_i(\theta^{(t)}) - \frac{\mu^{(t)}\|d - g_0\|}{\varepsilon} + \frac{\mu^{(t)}}{2}\min_j \frac{\|d - g_0\|}{\varepsilon} \\
&\leq \ell_i(\theta^{(t)}) - \frac{\mu^{(t)}\|d - g_0\|}{2\varepsilon} \leq \ell_i(\theta^{(t)})
\end{aligned} \tag{38}$$

This inequality implies that the objective function value of all tasks strictly decreases with each iteration when using the GetMTL algorithm. We next analyze the rationality of step size $\mu^{(t)}$ in Lemma C.2.

■

Lemma C.2. *The convergence of Gradient Descent with step size μ is guaranteed only if the step size $\mu > 0$ is carefully chosen such that $\mu < 1/L$ (Nesterov, 1998; Ward et al., 2020) where $L > 0$ is the e Lipschitz smoothness constant. Then we have,*

$$0 < \mu < 1/L \tag{39}$$

Proof. (1) Proof of left part of inequality.

$$\mu = \min_{i\in[k]} \frac{\|d - g_0\|}{\varepsilon \cdot L \cdot d^2}, \text{ s.t. } \varepsilon \in (0, 1], L > 0 \tag{40}$$

Therefore, we can get $\mu > 0$.

(2) Proof of right part of inequality.

$$\begin{aligned}
\mu &= \min_{i\in[k]} \frac{\|d - g_0\|}{\varepsilon \cdot L \cdot \|d\|^2} \text{ (using } \|d - g_0\| \leq \varepsilon \cdot g_0^\mathsf{T} d) \\
&\leq \min_{i\in[k]} \frac{\varepsilon g_0^\mathsf{T} d}{\varepsilon \cdot L \cdot \|d\|^2} = \frac{g_0^\mathsf{T} \cdot d}{L \cdot \|d\|^2} \\
&= \frac{\|g_0\| \cdot \|d\| \cos\varphi}{L \cdot \|d\|^2} = \frac{\|g_0\| \cos\varphi}{\|d\|} \cdot \frac{1}{L}
\end{aligned}$$

where $\varphi \in [0°, 90°)$ denotes the angle of d and g_0. In general, we all penalize gradient norm for improving the generalization and stability. We thus can get $\|d\|^2 - \|g_0\|^2 > 0$ when $\varepsilon \in (0, 1]$. Then,

$$\mu \leq \frac{\|g_0\|\|d\| \cos\varphi}{L \cdot \|d\|^2} = \frac{|g_0| \cos\varphi}{\|d\|} \cdot \frac{1}{L} < \frac{1}{L},$$

Then, we can get $0 < \mu < 1/L$.

■

Tasks	STL	Uniform	Uncertainty	GradNorm	MGDA	TchebycheffAdv	BanditMTL	MetaWeighting	GetMTL(Ours)
COMP	87.36	86.84	86.76	86.26	87.88	87.36	88.06	87.99	**89.67**
REC	94.48	96.21	96.02	95.63	96.25	95.84	96.16	95.9	**96.39**
SCI	94.45	96.26	96.35	96.08	95.78	95.82	95.66	96.08	**96.56**
TALK	85.04	86.08	86.27	85.94	86.56	85.96	85.93	85.82	**86.84**
AVG	90.43	90.93	90.87	90.7	91.2	90.87	91.26	91.25	**92.09**

Table 2: The complete performance of 4 tasks in topic classification dataset with our GetMTL and other MTL baselines.

Tasks	STL	Uniform	Uncertainty	GradNorm	MGDA	TchebycheffAdv	BanditMTL	MetaWeighting	GetMTL(Ours)
Apparel	87.57	89.18	89.59	88.69	88.63	87.98	88.95	89.83	**90.03**
Baby	87.14	89.91	89.96	89.33	89.05	88.65	90.02	90.01	**90.32**
Books	87.02	87.64	87.09	87.14	85.66	86.65	87.09	86.82	**87.77**
Camera	90.54	91.49	91.54	90.84	91.05	91.44	91.54	91.54	**92.26**
Dvd	84.61	88.17	87.35	87.32	87.65	87.24	87.08	88.02	**89.30**
Electronics	85.42	88.09	88.68	88.88	87.94	86.80	87.60	86.99	**89.49**
Health	89.07	90.82	91.50	90.59	90.86	90.55	91.81	**91.85**	**91.85**
Kitchen	85.16	89.51	89.65	89.33	88.69	87.67	90.07	89.25	**90.81**
Magazines	93.32	93.61	92.54	93.35	93.21	93.40	93.36	94.30	**94.43**
Music	83.92	84.27	86.25	84.97	85.01	83.90	86.37	86.88	**87.04**
Software	89.97	92.44	92.59	93.24	92.82	92.77	92.95	92.71	**93.93**
Sports	87.52	90.52	90.42	90.88	90.65	89.85	89.72	89.96	**91.81**
Toys	87.02	88.73	89.89	88.10	88.30	88.49	88.47	89.11	**90.62**
Video	88.8	89.65	89.28	88.92	89.33	89.06	89.62	**89.88**	89.55
Avg	86.52	88.47	88.74	88.01	88.30	87.71	88.78	89.14	**89.80**

Table 3: The complete performance of 14 tasks in amazon review dataset with our GetMTL and other MTL baselines.

D Complete Performance of Each Task for Amazon Dataset

Amazon review dataset includes 14 domains, such as *Apparel*, *Baby*, *Books*, *Camera*, *Dvd*, *Electronics*, *Health*, *Kitchen*, *Magazines*, *Music*, *Software*, *Sports*, *Toys*, and *Video*. Each domain is treated as a 14 binary classification task.

We provide the full comparison on the amazon review and topic classification datasets in Table 3 and Table 2 respectively. Table 2 shows that our GetMTL can achieve the best average classification accuracy of 92.09%, outperforming the second-best model BanditMTL by a margin of 0.83%. Moreover, our GetMTL can also beat other baselines on each individual tasks. Table 3 reports the performance of all 14 tasks on amazon review dataset. Our proposed GetMTL achieves the best performance on 13 out of 14 tasks and obtain best average classification accuracy.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section of Limitations

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section of GetMTL, Experimental datasets

☑ B1. Did you cite the creators of artifacts you used?
Experimental datasets

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
It is published by the authors.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section of Experimental Implementation

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☒ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
No response.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
No response.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
No response.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.