A Model-Agnostic Approach to Mitigate Gradient Interference for Multi-Task Learning

Heyan Chai, Zhe Yin, Ye Ding, *Member, IEEE*, Li Liu, *Senior Member, IEEE*,
Binxing Fang, *Member, IEEE*, and Qing Liao, *Member, IEEE*

***Abstract*—Multitask learning (MTL) is a powerful technique for jointly learning multiple tasks. However, it is difficult to achieve a tradeoff between tasks during iterative training, as some tasks may compete with each other. Existing methods manually design specific network models to mitigate task conflicts, but they require considerable manual effort and prior knowledge about task relationships to tune the model so as to obtain the best performance for each task. Moreover, few works have offered formal descriptions of task conflicts and theoretical explanations for the cause of task conflict problems. In this article, we provide a formal description of task conflicts that are caused by the gradient interference problem of tasks. To alleviate this issue, we propose a novel model-agnostic approach to mitigate gradient interference (MAMG) by designing a *gradient clipping rule* that directly modifies the interfering components on the gradient interfering direction. Specifically, MAMG is model-agnostic and thus it can be applied to a large number of multitask models. We also theoretically prove the convergence of MAMG and its superiority to existing MTL methods. We evaluate our method on a variety of real-world large datasets, and extensive experimental results confirm that MAMG can outperform some state-of-the-art algorithms on different types of tasks and can be easily applied to various methods.**

***Index Terms*—Deep learning, gradient interference, multitask learning.**

I. Introduction

WHILE deep learning has shown surprising promise in enabling machines to learn more complex tasks, the large number of training data requirements of existing deep learning models make it difficult to learn a great variety of capabilities in real-world scenarios, especially when all tasks are learned independently from scratch. A straightforward approach to address this issue is to train a network on all tasks jointly, with the aim of leveraging the shared network structure across tasks to improve the performance of all tasks and achieve greater efficiency than training tasks individually. To some extent, training all tasks jointly can help tasks with limited data obtain better performance, as they can capture the shared information from other joint tasks. Caruana [1] proposed a new learning paradigm called multitask learning (MTL), which jointly learns all tasks to obtain superior performance over learning each task independently. Under the assumption that seemingly unrelated real-world tasks have strong dependencies due to the existing of shared processes, and a similar way of generating data [2], MTL has achieved great success in various research fields, such as image recognition [3], [4], [5], [6], autonomous driving [7], [8], disease diagnosis [9], [10], [11], and natural language processing [12], [13], [14]. Therefore, MTL plays a vital role in the actual application of deep learning.

However, learning multiple tasks simultaneously is a difficult optimization problem that can result in worse overall performance compared with training tasks individually [15]. We assume that the worse overall performance of the MTL model is caused by the imbalanced relationships between joint training tasks; that is, *different tasks may compete with each other during training*. This task conflict problem leads to poor data efficiency, leading to the degraded performance of all tasks.

Most prior works [16], [17], [18], [19], [20], [21], [22] have focused on designing customized network architecture to eliminate the gap of the gradient between tasks to alleviate the task conflict problem for different tasks. Moreover, auxiliary task-based approaches [13], [23], [24], [25] are dedicated to exploring the relationship between tasks, with the aim of finding more relevant auxiliary tasks to avoid gradient interference between tasks. The two aforementioned types of methods heavily rely on considerable manual effort, such as hand-designed architecture and hand-selected auxiliary tasks. More recently, we have seen a shift of paradigm in MTL, where a kind of gradient projection approach (PCGrad [15]) has been proposed to balance the relationship between joint training tasks. PCGrad simply selects the normal plane of the other task gradient as the conflicting plane, which can result

Manuscript received 11 August 2022; revised 18 October 2022; accepted 10 November 2022. Date of publication 6 December 2022; date of current version 22 November 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB2104003; in part by the National Natural Science Foundation of China under Grant 62076079; and in part by the Guangdong Major Project of Basic and Applied Basic Research under Grant 2019B030302002. This article was recommended by Associate Editor Q. Shen. *(Corresponding author: Qing Liao.)*

Heyan Chai, Zhe Yin, and Binxing Fang are with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, Guangdong, China (e-mail: chaiheyan@stu.hit.edu.cn; yinzhe@stu.hit.edu.cn; fangbx@cae.cn).

Ye Ding is with the School of Cyberspace Security, Dongguan University of Technology, Dongguan 523808, China (e-mail: dingye@dgut.edu.cn).

Li Liu is with the College of System Engineering, National University of Defense Technology, Changsha 410073, Hunan, China (e-mail: dreamliu2010@gmail.com).

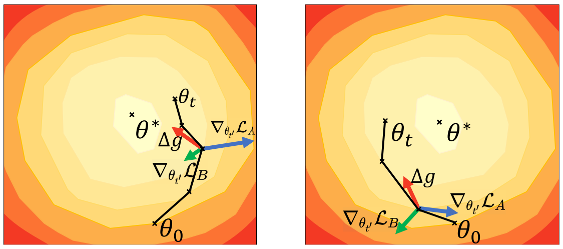
Qing Liao is with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, Guangdong, China, and also with the Department of New Networks, Peng Cheng Laboratory, Shenzhen 518055, Guangdong, China (e-mail: liaoqing@hit.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2022.3223377.

Digital Object Identifier 10.1109/TCYB.2022.3223377

2168-2267 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.


θ_t
θ^*
$\nabla_{\theta_{t'}}\mathcal{L}_A$
Δg
$\nabla_{\theta_{t'}}\mathcal{L}_B$
θ_0
θ_t
θ^*
Δg
$\nabla_{\theta_{t'}}\mathcal{L}_A$
$\nabla_{\theta_{t'}}\mathcal{L}_B$
θ_0

(a) (b)

Fig. 1. Illustration of *gradient interference problem* on a 2-D MTL system. The green and blue arrows denote the gradient vectors of tasks A and B, $\nabla_{\boldsymbol{\theta}_{t'}}\mathcal{L}_A$ and $\nabla_{\boldsymbol{\theta}_{t'}}\mathcal{L}_B$, respectively. The red arrow, $\Delta \boldsymbol{g}$ denotes the gradient descent direction of the entire multitask model, and $\boldsymbol{\theta}^*$ is the optimal model parameters. (a) Difference in the gradient magnitudes of tasks A and B may be much larger, which indicates that task A and task B compete with each other. (b) Angle between the two task gradients may be an obtuse angle on the 2-D space, and the difference in gradient magnitudes is generally smaller, which also indicates that the occurrence of gradient interference. The relative magnitudes of gradients of tasks are on the same scale. $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_t$ represent the model parameters of the 0th and t-th iteration, respectively. See text in Section I for details. This figure is best viewed in color.

in a larger task bias when the projected task gradient direction is close to the gradient of the MTL model. Consequently, it can lead to worse overall performance of all learning tasks due to the poor choice of conflict plane. In this work, we use task gradient interference to formally describe the task conflict problem and then propose a novel gradient clipping rule to mitigate the task conflict problem.

To better illustrate the issue of gradient interference, we take two tasks as examples, shown in Fig. 1, to illustrate how the gradient interference problem occurs. As shown in Fig. 1, the MTL model converges to the position where it is far from the optimal solution during optimization. Fig. 1(a) illustrates that the difference between the gradients of tasks A and B is large in terms of magnitude, which can cause network to focus on training task A and thus result in insufficient training for task B. Consequently, the overall performance of the MTL model will be degraded. Fig. 1(b) illustrates that the difference between the gradients of tasks A and B is large in terms of direction, which can cause the gradients of these two tasks to cancel each other out in certain directions. Such a situation can also lead to the degraded performance of the MTL model. In this article, we refer to the above situations as *gradient interference*. This article aims to mitigate the gradient interference problem to obtain better performance for all tasks.

Therefore, we argue that a more efficient way to mitigate the gradient interference problem is to modify the gradients directly so as to eliminate gradient conflicts. In this article, we propose a novel model-agnostic approach to mitigate gradient interference (MAMG) for improving the performance of all tasks. First, we give a formal definition of gradient interference. If the gradient interference problem exists, we will mitigate gradient interference by clipping the interfering gradient component on the gradient interfering direction, preventing the conflict gradient component from degrading the performance of the multitask model. Then, we theoretically prove the convergence of MAMG and its superiority to standard MTL methods. Moreover, MAMG can be applied to other models to improve its performance because it can lead to convergence to the minimizer of the loss function. The main contributions of this article can be summarized as follows.

1) We formally describe the *gradient interference* problem and give a concrete definition of gradient conflict, which has rarely been discussed in previous works.
2) We propose a novel model-agnostic approach to mitigate gradient interference for MTL models by designing a *gradient clipping rule* to directly modify the interfering components on the gradient interfering direction, which is general and not dependent on the relationship between tasks.
3) We theoretically prove that MAMG improves upon standard MTL models and analyze the convergence of MAMG compared with existing MTL models.
4) We evaluate MAMG on many real-world datasets, which include a variety of different-level pixel labeling tasks, image classification tasks, and text classification tasks, and the results show that our approach performs competitively with a variety of state-of-the-art methods under different combinations of tasks.

II. Related Work

In this section, we briefly review the existing methods for MTL in deep neural networks. Usually, based on the way of parameter sharing, MTL can be categorized into hard parameter sharing-based approaches and soft parameter sharing-based approaches.

A. Hard Parameter Sharing-Based Approaches

Hard parameter sharing approaches are generally applied by sharing the hidden layers among all tasks while setting task-specific layers for each task [2], [26], [27], [28], [29], [30], [31], [32]. In hard parameter sharing, all parameters are divided into two parts: 1) task-shared parameters and 2) task-specific parameters. The most common hard parameter sharing approaches consist of a shared network layer that branches out into task-specific networks [2], [27], [28]. UberNet [26] jointly processed a large number of low-, mid-, and high-level vision tasks by designing a hard parameter-sharing model, which was achieved by designing different heads for different tasks across different layers. Long et al. [33] utilized novel tensor normal priors over parameter tensors of task-specific layers to capture the task relatedness by designing the hard parameter-sharing model. The above models do not automatically determine when to branch out and usually need manual efforts to adjust the branching point. To address this issue, some recent works [22], [29], [30], [31], [32], [34] have proposed efficient network architecture search strategies that can automatically learn where to branch or share within a hard parameter sharing multitask model. Similarly, Bragman et al. [35] proposed a stochastic filter groups (SFGs) mechanism to learn task-shared and task-specific representations, which was achieved by assigning convolution kernels for task-shared and task-specific groups.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.

B. Soft Parameter Sharing-Based Approaches

Soft-parameter sharing approaches have a common structure: each task has its own model with its own set of parameters but different feature-/information-sharing mechanisms to tackle the cross-task information sharing. A large number of multitask models have been designed in the way of soft parameter sharing [16], [17], [36], [37], [38], [39], [40], [41]. For example, Sun et al. [37] proposed a sparse sharing mechanism to automatically find a sparse sharing structure by capturing the relationships among tasks. Misra et al. [16] employed a learnable linear combination to fuse the features shared by all learning tasks. Therefore, the network can dynamically determine the degree of the features shared among tasks. Subsequently, various techniques have been proposed to improve the feature fusion mechanism, such as Sluice Networks [40], NDDR-CNN [17], and MTAN [39]. More concretely, Sluice Networks [40] employed some skip connections that can help the model select different sharing subspaces on different network layers. NDDR-CNN [17] applied a dimensionality reduction mechanism to fuse the shared feature, which employed a 1 × 1 convolutional layer to process the channel-wise features. Liu et al. [39] used a shared backbone network to extract a general pool of features, and then designed task-specific attention modules for each task that can be used to select features from the feature pool for each task. Different from designing different strategies for feature sharing, a few recent works have employed a knowledge-distilling mechanism to fully use the features shared in the multitask network. For example, Xu et al. [36] employed spatial attention to distill information from other task predictions and then fused it into the target task. Zhang et al. [38] adaptively diffused similar patterns recurring across different tasks by propagating cross-task and task-specific patterns. Moreover, Zhou et al. [41] proposed a pattern-structure diffusion framework to capture and diffuse task-specific and task-across pattern structures for boosting the performance of MTL. However, all these approaches require considerable effort to design more complex network structures and process complex task interactions, especially when the number of tasks grows rapidly.

Unlike the above two types of parameter-sharing methods focusing on designing effective parameter-sharing strategies, some approaches leverage other techniques to facilitate the training of MTL. Some approaches focus on dynamically adjusting the weights of tasks [25], [27], [39], [42]. For example, several auxiliary-learning approaches have been proposed to utilize the similarity between main and auxiliary tasks to dynamically adjust the relationship between tasks [25], [42]. Liu et al. [39] encouraged all tasks to train at the same speed by enabling the gradient magnitude of tasks to be similar. Kendall et al. [27] utilized homoscedastic uncertainty to measure the difference of different tasks and then used it to dynamically adjust the weights of tasks. Some methods treat the MTL problem as multiobjective optimization (MOO) and apply the Pareto Optimization to solve the MTL problem [2], [43], [44]. More recently, Yu et al. [15] proposed a gradient projection approach to project the conflicted task gradient onto another gradient and then remove the conflicted gradient component to alleviate task conflicts.

Current MTL models focus on designing effective feature-sharing strategies to help share cross-task features. However, they on the one hand need huge efforts to balance task-specific losses during training to solve task conflicts, and on the other hand lack a formal theoretical explanation to interpret why they can solve the task conflicts. We thus propose a novel gradient clipping algorithm to mitigate this issue.

III. Our Methodology

In this section, we first present the formal problem definitions of MTL and gradient interference. Then, an efficient gradient clipping rule is introduced in detail in the following section. Finally, we theoretically prove and analyze the effectiveness and superiority of MAMG over existing approaches.

A. Problem Formulation and Notation

Many MTL applications generally define a single-objective optimization problem by simply averaging the task gradients to optimize the multitask objective. This strategy cannot fundamentally address gradient interference among tasks, which leads to significantly degraded performance. Analysis of the causes of gradient interference problems has rarely been undertaken in previous works, and thus this is the first work to deeply explore the causes of gradient interference problems.

To better analyze the gradient interference problem in MTL, we take two tasks as examples to illustrate the gradient interference problem, as shown in Fig. 1. There are two situations where the gradient interference problem can occur.

1) When the two tasks compete or conflict with each other, the difference in gradient magnitudes of tasks may be much larger. As shown in Fig. 1(a), task A and task B compete with each other. More specifically, the gradient of task A is much larger in magnitude than that of task B, and $\|\nabla_{\boldsymbol{\theta}_{t'}}\mathcal{L}_A\| \gg \|\nabla_{\boldsymbol{\theta}_{t'}}\mathcal{L}_B\|$, where $\|\nabla_{\boldsymbol{\theta}_{t'}}\mathcal{L}_A\|$ denotes the magnitude of the gradient of task A at parameter $\boldsymbol{\theta}_{t'}$. Moreover, the angle between the two gradients is obtuse on the 2-D space. Therefore, task A will dominate the overall gradient of the multitask model, which can cause networks to focus on the training of task A by back-propagating a larger gradient. Task B will eventually be overwhelmed by task A when iterating continuously. The performance of the model will be significantly degraded.
2) When the two tasks compete or conflict with each other, the angle between two task gradients may be obtuse on the 2-D space, and the difference in gradient magnitudes will be generally smaller. As shown in Fig. 1(b), the difference in the gradient magnitudes of tasks A and B is small, $\|\nabla_{\boldsymbol{\theta}_{t'}}\mathcal{L}_A\| \approx \|\nabla_{\boldsymbol{\theta}_{t'}}\mathcal{L}_B\|$, so the gradient interference problem does not exist in the gradient magnitude. However, the angle between gradients is obtuse on the 2-D space, and we assume that the gradients of these two tasks may cancel each other out in certain directions. Thus, these tasks will be far from being fully

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.

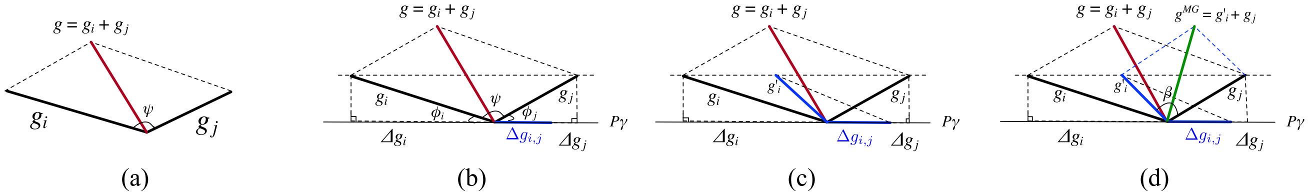


Fig. 2. Clipping the interfering gradients on a 2-D MTL system. (a) Tasks i and j have conflicting gradients. (b) We define a gradient interfering direction $\boldsymbol{P\gamma}$ and calculate the interfering components of tasks i and j, denoted as $\Delta \boldsymbol{g}_i$ and $\Delta \boldsymbol{g}_j$, respectively. (c) We clip the interfering gradient $\Delta \boldsymbol{g}_{i,j}$ and obtain the modified gradient of task i, called $\boldsymbol{g}_i'$. (d) We mitigate the gradient interference problem in the MTL system and obtain the modified gradient $\boldsymbol{g}^{MG}$ without interference.

trained, and then the model will converge to a position far from the optimal point shown in Fig. 1(b).

Since the magnitudes of gradients are highly dependent on the network structure, we assume that the cause of gradient interference is the angle between the task gradients. A more complex neural-network structure can produce greater conflicts in gradient magnitudes. In this article, we focus on mitigating the gradient interference problem on the angle between task gradients. Let the gradient of task i be $\boldsymbol{g}_i = \nabla \mathcal{L}^i(\boldsymbol{\theta})$. We formally define situation (2) as follows.

Definition 1: Given a mini-batch of training samples from tasks i and j, where $\nabla_{\boldsymbol{\theta}} \mathcal{L}^i$ and $\nabla_{\boldsymbol{\theta}} \mathcal{L}^j$ denote gradients from samples of tasks i and j, respectively, over the network parameters $\boldsymbol{\theta}$, we define $\alpha_{i,j}$ as a condition of gradient interference in the gradient direction. Formally

$$\alpha_{i,j} = \operatorname{sign}\left(\left\langle \nabla_{\boldsymbol{\theta}} \mathcal{L}^i, \nabla_{\boldsymbol{\theta}} \mathcal{L}^j \right\rangle\right) \tag{1}$$

where $\alpha_{i,j} = -1$ denotes that tasks i and j conflict with each other, and 0 otherwise. Essentially, the interfering gradients hinder the learning of parameter $\boldsymbol{\theta}$ and degrade the impact of individual tasks on the overall model. As a consequence, the gradient interference leads to a low-quality local optimum w.r.t. $\boldsymbol{\theta}$.

We observe the presence of the gradient interference, shown in Fig. 1. A two-task example is used to analyze the conditions when gradient interference happens. Motivated by this analysis, we propose a novel approach to mitigate the gradient interference problem by designing a gradient clipping rule to directly modify the gradient of conflicting tasks.

B. Proposed Method

Our goal is to reduce the negative impact of the gradient interference problem by directly clipping the gradients. In this section, we design a gradient clipping rule to mitigate the gradient interference problem. The core details of the rule are presented through a two-task example, shown in Fig. 2. We also theoretically prove the superiority and effectiveness of directly clipping interfering gradients in the next section.

The gradient interference problem often happens when gradients from different tasks have components in conflicting directions, as shown in Definition 1. Intuitively, we directly modify the conflicting gradients by designing the gradient clipping rule, which does not affect the network structure. In other words, it is a model-agnostic approach and does not depend on specific models.

We hypothesize that components of the gradient for different tasks may interfere with each other in a certain plane or direction. Therefore, to address this problem, we should find that direction and modify the interfering gradients. For a two-task MTL model, the gradient can be computed as $\boldsymbol{g} = \boldsymbol{g}_i + \boldsymbol{g}_j$, where $\boldsymbol{g}_i$ and $\boldsymbol{g}_j$ are the gradients of tasks i and j, respectively. Ideally, the gradient components of $\boldsymbol{g}_i$ and $\boldsymbol{g}_j$ on the tangent plane of gradient $\boldsymbol{g}$ should cancel each other out, and these components do not help the model converge to the optimum. As the global optimal gradient is hard to obtain, we cannot obtain the global optimal gradient or calculate the optimal tangent direction of the optimal gradient. Alternatively, we find a direction that is not the optimal interfering direction but is close to the tangent direction of $\boldsymbol{g}$, which is formally defined below.

Definition 2 (Gradient Interfering Direction): Given two task gradients, $\boldsymbol{g}_i$ and $\boldsymbol{g}_j$, if the tangent direction of the gradient of $\boldsymbol{g}$ exists, where the gradient components of $\boldsymbol{g}_i$ and $\boldsymbol{g}_j$ may cancel each other out, then we call this direction the *gradient interfering direction*. We define the *gradient interfering direction* between two gradients $\boldsymbol{g}_i$ and $\boldsymbol{g}_j$ as $\boldsymbol{P\gamma} = \boldsymbol{g}_i - \boldsymbol{g}_j$.

As shown in Fig. 2(a), when the gradient interference problem happens (it meets the condition of Definition 1), we compute the gradient components of each task on the interfering direction $\boldsymbol{P\gamma}$. We define the angle between gradients of each task and interfering direction as ϕ. Formally

$$\cos\phi_i = \frac{\boldsymbol{g}_i \cdot \boldsymbol{P\gamma}}{\|\boldsymbol{g}_i\|_2 \|\boldsymbol{P\gamma}\|_2}, \quad \cos\phi_j = \frac{\boldsymbol{g}_j \cdot \boldsymbol{P\gamma}}{\|\boldsymbol{g}_j\|_2 \|\boldsymbol{P\gamma}\|_2} \tag{2}$$

where ϕ_i and ϕ_j denote the angle between the gradients of tasks i and j and interfering direction $\boldsymbol{P\gamma}$, defined as $\phi_i = \langle \boldsymbol{g}_i, \boldsymbol{P\gamma} \rangle$, $\phi_j = \langle \boldsymbol{g}_j, \boldsymbol{P\gamma} \rangle$. Then, as shown in Fig. 2(b), we can calculate the gradient difference on the $\boldsymbol{P\gamma}$ by

$$\Delta \boldsymbol{g}_{i,j} = \|\boldsymbol{g}_i \cdot \cos\phi_i - \boldsymbol{g}_j \cdot \cos\phi_j\| \tag{3}$$

where $\boldsymbol{g}_i \cdot \cos\phi_i$ denotes the interfering gradient component of task i on interfering direction $P\gamma$, and $\Delta \boldsymbol{g}_{i,j}$ is the gradient difference of tasks i and j on direction $\boldsymbol{P\gamma}$. We use $\Delta \boldsymbol{g}_{i,j}$ to quantitatively describe the degree of gradient interference. Intuitively, we modify the gradient of tasks that meets the condition of Definition 1 by making full use of the gradient difference $\Delta \boldsymbol{g}_{i,j}$ on the interfering direction.

The core of alleviating the interfering gradient is to break the condition under which the gradient interference problem occurs. An intuitive way to achieve this is to reduce the angle between different task gradients to meet $\alpha_{i,j} \neq -1$,

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Gradient Clipping Rule

```
Input: Loss $\mathcal{L}^i$ of task $\mathcal{T}_i$, Network parameters $\boldsymbol{\theta}$, Task set
    $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^t$.
Output: Modified gradient $\boldsymbol{g}^{MG}$ of the multi-task model.
for $\mathcal{T}_i \in \mathcal{T}$ do
    $\boldsymbol{g}_i = \nabla \mathcal{L}^i(\boldsymbol{\theta})$
    for $\mathcal{T}_j$ samplefrom$\{\mathcal{T} \setminus \mathcal{T}_i\}$ do
        $\boldsymbol{g}_j = \nabla \mathcal{L}^i(\boldsymbol{\theta})$
        if $\boldsymbol{g}_i \cdot \boldsymbol{g}_j < 0$ then
            calculate gradient interfering direction $\boldsymbol{P\gamma} = \boldsymbol{g}_i - \boldsymbol{g}_j$.
            compute angle $\cos\phi_i$ and $\cos\phi_j$ by using eq. (2).
            calculate the interfering gradient $\Delta \boldsymbol{g}_{i,j}$ by using Eq. (3).
            obtain the clipped gradient $\boldsymbol{g}_i' = \boldsymbol{g}_i - \Delta \boldsymbol{g}_{i,j} \cdot \frac{\boldsymbol{P\gamma}}{\|\boldsymbol{P\gamma}\|}$.
        end if
    end for
end for
return modified gradient $\boldsymbol{g}^{MG} = \frac{1}{t}\sum_i^t \boldsymbol{g}_i'$
```

which breaks the condition defined in Definition 1 (defined as $\alpha_{i,j} = -1$). Therefore, we use the difference between the gradient g_i and the interfering gradient difference $\Delta \boldsymbol{g}_{i,j}$ to update the original gradient of task i. As shown in Fig. 2(c), $\boldsymbol{g}_i'$ is the modified gradient of task i. When we use $\boldsymbol{g}_i'$ to substitute $\boldsymbol{g}_i$, the angle between the gradients of task i and j becomes smaller (from ψ to β). This breaks the condition of the gradient interference problem. We mitigate the gradient interference problem to a certain extent. Formally, we calculate the modified $\boldsymbol{g}_i'$ by

$$\boldsymbol{g}_i' = \boldsymbol{g}_i - \Delta \boldsymbol{g}_{i,j} \cdot \frac{\boldsymbol{P\gamma}}{\|\boldsymbol{P\gamma}\|} \tag{4}$$

where $\boldsymbol{P\gamma}$ is the interfering direction described in Definition 2, $\Delta \boldsymbol{g}_{i,j}$ denotes the magnitude of difference of interfering gradients of different tasks, and $(\boldsymbol{P\gamma}/\|\boldsymbol{P\gamma}\|)$ denotes the direction of $\Delta \boldsymbol{g}_{i,j}$. Equation (4) is the operation between vectors. Finally, as shown in Fig. 2(d), we can obtain the gradient of the MTL model $\boldsymbol{g}^{MG}$ after mitigating the gradient interference. This is equivalent to removing the interfering component of the gradient for the task, thereby degrading the degree of destructive gradient interference between tasks. Our MAMG repeats the gradient clipping process across all of the other tasks randomly sampled from the current batch $\{\mathcal{T}_j\}_{j=1}^t \backslash \mathcal{T}_i$, resulting in the gradient $\boldsymbol{g}_i'$ that is applied for task $\mathcal{T}_i$. We also perform the same procedure for all tasks in the current batch to obtain the modified gradient. The detailed process of clipping interfering gradients is given in Algorithm 1.

C. Theoretical Analysis

In this section, we theoretically analyze the effectiveness and superiority of MAMG under a two-task learning scenario. We first give some definitions of concepts for simplicity.

Definition 3: Consider two task-loss functions $\mathcal{L}^1 : \mathbb{R}^n \to \mathbb{R}$ and $\mathcal{L}^2 : \mathbb{R}^n \to \mathbb{R}$. We define the two-task learning objective as $\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}^1(\boldsymbol{\theta}) + \mathcal{L}^2(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \mathbb{R}^n$, where $\boldsymbol{g}_1 = \nabla\mathcal{L}^1(\boldsymbol{\theta})$, $\boldsymbol{g}_2 = \nabla\mathcal{L}^2(\boldsymbol{\theta})$, and $\boldsymbol{g} = \boldsymbol{g}_1 + \boldsymbol{g}_2$.

First, we analyze the effectiveness of the MAMG gradient clipping rule by proving the convergence of MAMG under the L-Lipschitz assumptions in Theorem 1.

Theorem 1: If all the objective functions, $\mathcal{L}^1$ and $\mathcal{L}^2$, are differentiable and convex, then assuming the gradient of $\mathcal{L}$ is L-Lipschitz continuous with $L > 0$, the proposed MAMG gradient clipping rule with step size $t < (1/L)$ will converge to the optimal value $\mathcal{L}(\boldsymbol{\theta}^*)$.

Proof: Following the above definitions, let $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ denote the gradient of tasks 1 and 2, $\boldsymbol{P\gamma}$ be the interfering direction, and the condition of gradient interference be $\alpha_{1,2} = \text{sign}(\langle \boldsymbol{g}_1, \boldsymbol{g}_2 \rangle) = -1$. We define the angle between $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ as ψ. Therefore, when gradient interference happens, the angle between $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ meets $\cos\psi \geq 0$. At each iteration of model training, there are two cases: the gradient interference problem exists or does not exist.

Case 1: If the gradient interference problem does not happen, according to Definition 1, we know that $\cos\psi \geq 0$. We can use standard gradient descent algorithms (e.g., SGD and Adam) with step size $t \leq (1/L)$ to iteratively solve the solution until $\nabla\mathcal{L}(\boldsymbol{\theta}) = 0$.

Case 2: If the gradient interference problem happens, namely, $\cos\psi < 0$, we present the detailed process as follows.

According to the assumption that $\nabla\mathcal{L}(\boldsymbol{\theta})$ is Lipschitz continuous [45] with constant L, we extend $\mathcal{L}$ to quadratic at $\boldsymbol{\theta}$, and then we obtain the following inequality:

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}^+\right) &\leq \mathcal{L}(\boldsymbol{\theta}) + \nabla\mathcal{L}(\boldsymbol{\theta})^T\left(\boldsymbol{\theta}^+ - \boldsymbol{\theta}\right) + \frac{1}{2}\nabla^2\mathcal{L}(\boldsymbol{\theta})\left\|\boldsymbol{\theta}^+ - \boldsymbol{\theta}\right\|^2 \\ &\leq \mathcal{L}(\boldsymbol{\theta}) + \nabla\mathcal{L}(\boldsymbol{\theta})^T\left(\boldsymbol{\theta}^+ - \boldsymbol{\theta}\right) + \frac{1}{2}L\left\|\boldsymbol{\theta}^+ - \boldsymbol{\theta}\right\|^2.\end{aligned} \tag{5}$$

Now, we can use the MAMG gradient clipping rule to update the $\boldsymbol{\theta}^+$. According to Definition 3, (3), and (4), we can update $\boldsymbol{\theta}^+$ by $\boldsymbol{\theta}^+ = \boldsymbol{\theta} - t \cdot \boldsymbol{g} = \boldsymbol{\theta} - t \cdot (g_1 - \Delta\boldsymbol{g}_{1,2} \cdot (\boldsymbol{P\gamma}/\|\boldsymbol{P\gamma}\|) + \boldsymbol{g}_2)$

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}^+\right) \leq\ & \mathcal{L}(\boldsymbol{\theta}) - t \cdot \boldsymbol{g}^T\left(\boldsymbol{g} + \frac{\boldsymbol{g}_2^2 - \boldsymbol{g}_1^2}{\|\boldsymbol{P\gamma}\|^2} \cdot \left(\boldsymbol{g}_2 - \boldsymbol{g}_1\right)\right) \\ & + \frac{1}{2} \cdot L \cdot t^2\left\|\boldsymbol{g} + \frac{\boldsymbol{g}_2^2 - \boldsymbol{g}_1^2}{\|\boldsymbol{P\gamma}\|^2} \cdot \left(\boldsymbol{g}_2 - \boldsymbol{g}_1\right)\right\|^2.\end{aligned} \tag{6}$$

Then, using the identity $\boldsymbol{g} = \boldsymbol{g}_1 + \boldsymbol{g}_2$, we can obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}^+) \leq\ & \mathcal{L}(\boldsymbol{\theta}) - t\left(\left(\boldsymbol{g}_1 + \boldsymbol{g}_2\right)^2 + \frac{\boldsymbol{g}_2^2 - \boldsymbol{g}_1^2}{\|\boldsymbol{P\gamma}\|^2} \cdot \left(\boldsymbol{g}_2^2 - \boldsymbol{g}_1^2\right)\right) \\ & + \frac{1}{2}Lt^2\left\|\left(\boldsymbol{g}_1 + \boldsymbol{g}_2\right)^2 + 2\frac{\boldsymbol{g}_2^2 - \boldsymbol{g}_1^2}{\|\boldsymbol{P\gamma}\|^2}\left(\boldsymbol{g}_2^2 - \boldsymbol{g}_1^2\right) + \frac{\left(\boldsymbol{g}_2^2 - \boldsymbol{g}_1^2\right)^2}{\|\boldsymbol{P\gamma}\|^2}\right\|.\end{aligned} \tag{7}$$

Expanding further and simplifying the formula, we can obtain the final inequality:

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}^+\right) &\leq \mathcal{L}(\boldsymbol{\theta}) + 2 \cdot \left(Lt^2 - t\right)\|\boldsymbol{g}_1 + \boldsymbol{g}_2\|^2 \\ &\leq \mathcal{L}(\boldsymbol{\theta}) + 2 \cdot \left(Lt^2 - t\right)\|\boldsymbol{g}\|^2.\end{aligned} \tag{8}$$

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.

Using $t \leq (1/L)$, we can obtain that $Lt^2 - t = t(Lt - 1) \leq t(L \cdot (1/L) - 1) = t \cdot 0 = 0$, and thus $2(Lt^2 - t)\|\boldsymbol{g}\|^2 \leq 0$. Finally, we obtain the following inequality:

$$\mathcal{L}(\boldsymbol{\theta}^+) \leq \mathcal{L}(\boldsymbol{\theta}). \tag{9}$$

This inequality proves that the value of $\mathcal{L}$ will decrease during every iteration when using the proposed MAMG gradient clipping rule. Therefore, MAMG is effective and can guarantee that the objective function value is monotonically decreasing. ■

Lemma 1: For the two task examples shown in Fig. 2(d), let $\boldsymbol{g}^{MG} = \boldsymbol{g}_i' + \boldsymbol{g}_j$ be the modified gradient of the MTL model by the MAMG gradient clipping rule, and $\boldsymbol{g} = \boldsymbol{g}_i + \boldsymbol{g}_j$ be the original gradient without modification

$$\|\boldsymbol{g}\| \geq \left\|\boldsymbol{g}^{MG}\right\|. \tag{10}$$

Theorem 2: Given the gradient $\boldsymbol{g}^{MG}$ updated by using the MAMG rule, if all the objective functions $\mathcal{L}^1, \mathcal{L}^2, \ldots, \mathcal{L}^n$ are differentiable and $\nabla\mathcal{L}^n(\boldsymbol{\theta})$ is Lipschitz continuous with constant $L > 0$, then for the angle between gradient g and modified gradient $\cos(\boldsymbol{g}, \boldsymbol{g}^{MG}) \geq (1/2)$, there exists a step size $t \leq (1/L)$, such that

$$\mathcal{L}(\boldsymbol{\theta}^+) \leq \mathcal{L}(\boldsymbol{\theta}) - R \tag{11}$$

where $R \geq 0$ is a positive variable w.r.t step size t. Theorem 2 shows that applying the MAMG process can reach the optimal value $\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}^*)$.

Proof: According to our assumption that $\nabla\mathcal{L}(\boldsymbol{\theta})$ is Lipschitz continuous with constant $L > 0$, we can perform a quadratic expansion of $\mathcal{L}$ around $\boldsymbol{\theta}$, which is defined as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}^+) &\leq \mathcal{L}(\boldsymbol{\theta}) + \nabla\mathcal{L}(\boldsymbol{\theta})^T(\boldsymbol{\theta}^+ - \boldsymbol{\theta}) + \frac{1}{2}\nabla^2\mathcal{L}(\boldsymbol{\theta})\|\boldsymbol{\theta}^+ - \boldsymbol{\theta}\|^2 \\ &\qquad \text{(Using the definition of Lipschitz-continuous)} \\ &\leq \mathcal{L}(\boldsymbol{\theta}) + \nabla\mathcal{L}(\boldsymbol{\theta})^T(\boldsymbol{\theta}^+ - \boldsymbol{\theta}) + \frac{1}{2}L\|\boldsymbol{\theta}^+ - \boldsymbol{\theta}\|^2.\end{aligned} \tag{12}$$

Now, we can use the MAMG gradient clipping rule to update $\boldsymbol{\theta}^+$ by letting $\boldsymbol{\theta}^+ = \boldsymbol{\theta} - t \cdot \boldsymbol{g}^{MG}$ and $\boldsymbol{g}^{MG} = \boldsymbol{g}_1' + \boldsymbol{g}_2$. Then, we can obtain the simplified formula

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}^+) &\leq \mathcal{L}(\boldsymbol{\theta}) - t \cdot \boldsymbol{g}^T \cdot \boldsymbol{g}^{MG} + \frac{1}{2}Lt^2\left\|\boldsymbol{g}^{MG}\right\|^2 \\ &\qquad \left(\text{Using the assumption } \cos\left(\boldsymbol{g}, \boldsymbol{g}^{MG}\right) \geq \frac{1}{2}\right) \\ &\leq \mathcal{L}(\boldsymbol{\theta}) - \frac{1}{2}t \cdot \|\boldsymbol{g}\| \cdot \left\|\boldsymbol{g}^{MG}\right\| + \frac{1}{2}Lt^2\left\|\boldsymbol{g}^{MG}\right\|^2 \\ &\qquad \left(\text{Using the Lemma 1 } \|\boldsymbol{g}\| \geq \left\|\boldsymbol{g}^{MG}\right\|\right) \\ &\leq \mathcal{L}(\boldsymbol{\theta}) - \frac{1}{2}t \cdot \|\boldsymbol{g}\| \cdot \left\|\boldsymbol{g}^{MG}\right\| + \frac{1}{2}Lt^2\left\|\boldsymbol{g}^{MG}\right\| \cdot \|\boldsymbol{g}\|.\end{aligned} \tag{13}$$

After rearranging terms, we obtain the final inequality

$$\mathcal{L}(\boldsymbol{\theta}^+) \leq \mathcal{L}(\boldsymbol{\theta}) - \frac{1}{2} \cdot \left(t - L \cdot t^2\right) \cdot \|\boldsymbol{g}\| \cdot \left\|\boldsymbol{g}^{MG}\right\|. \tag{14}$$

Note that $(1/2) \cdot (t - L \cdot t^2) \cdot \|\boldsymbol{g}\| \cdot \|\boldsymbol{g}^{MG}\| \geq 0$ when $t \leq (1/L)$. Thus, the application of the MAMG gradient clipping rule can guarantee a strict decrease in the value of objective functions $\mathcal{L}(\boldsymbol{\theta}^+) \leq \mathcal{L}(\boldsymbol{\theta}) - R$ after a large number of iterations. When $(1/2)t \cdot \|\boldsymbol{g}\| \cdot \|\boldsymbol{g}^{MG}\| + (1/2)Lt^2\|\boldsymbol{g}^{MG}\| \cdot \|\boldsymbol{g}\| = 0$ if and only if $\|\boldsymbol{g}\| = 0$ or $\|\boldsymbol{g}^{MG}\| = 0$. ■

Theorems 1 and 2 prove that the application of MAMG guarantees a decrease in the value of objective functions, and it can help the model converge to the minimizer of $\mathcal{L}$. When the gradient interference problem happens, we can repeatedly apply the MAMG gradient clipping rule to update the model parameters $\boldsymbol{\theta}$, and the objective function value will strictly decrease until it reaches the optimal value $\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}^*)$.

For further analysis on superiority, we theoretically prove that MAMG improves upon the newest gradient method PCGrad [15] and standard MTL methods. First, according to [15], we define the multitask curvature as $\mathbf{H}(\mathcal{L}; \boldsymbol{\theta}, \boldsymbol{\theta}') = \int_0^1 \nabla\mathcal{L}(\boldsymbol{\theta})^T\nabla^2\mathcal{L}(\boldsymbol{\theta} + \tau(\boldsymbol{\theta}' - \boldsymbol{\theta}))\nabla\mathcal{L}(\boldsymbol{\theta})d\tau$, which is the average curvature of $\mathcal{L}$ between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ in the direction of $\nabla\mathcal{L}(\boldsymbol{\theta})$. The details are presented in the following theorem.

Theorem 3: Assume $\mathcal{L}(\boldsymbol{\theta})$ is differentiable, the gradient $\mathcal{L}(\boldsymbol{\theta})$ is Lipschitz continuous with constant $L > 0$, and multitask curvature $\mathbf{H}(\mathcal{L}; \boldsymbol{\theta}, \boldsymbol{\theta}') \geq L\|\boldsymbol{g}\|^2$. Let $\boldsymbol{\theta}^{MT}$ and $\boldsymbol{\theta}^{MG}$ be the parameters after updating $\boldsymbol{\theta}$ by $\boldsymbol{g}$ and $\boldsymbol{g}^{MG}$ modified by the MAMG gradient clipping rule, respectively. Both of them use step size $t \leq (1/L)$. Then

$$\mathcal{L}\left(\boldsymbol{\theta}^{MG}\right) \leq \mathcal{L}\left(\boldsymbol{\theta}^{PG}\right). \tag{15}$$

To prove Theorem 3, according to [15], let $\boldsymbol{\theta}^{PG} = \boldsymbol{\theta} - t \cdot \boldsymbol{g}^{PG} = \boldsymbol{\theta} - t \cdot (\boldsymbol{g} - [(\boldsymbol{g}_1 \cdot \boldsymbol{g}_2)/(\|\boldsymbol{g}_1\|^2)]\boldsymbol{g}_1 - ([\boldsymbol{g}_1 \cdot \boldsymbol{g}_2]/[\|\boldsymbol{g}_2\|^2])\boldsymbol{g}_2)$, and $\boldsymbol{\theta}^{MG} = \boldsymbol{\theta} - t \cdot \boldsymbol{g}^{MG}$.

Proof: We first use the definition of the Lipschitz-continuous gradient [45] to obtain the following result:

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}^{\text{PG}}\right) &= \mathcal{L}(\boldsymbol{\theta}) + \int_0^1 \left\langle \nabla\mathcal{L}\left(\boldsymbol{\theta} + \tau\left(\boldsymbol{\theta}^{PG} - \boldsymbol{\theta}\right)\right), \boldsymbol{\theta}^{PG} - \boldsymbol{\theta}\right\rangle d\tau \\ &= \mathcal{L}(\boldsymbol{\theta}) + \left\langle \nabla\mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta}^{PG} - \boldsymbol{\theta}\right\rangle \\ &\quad + \int_0^1 \left\langle \nabla\mathcal{L}\left(\boldsymbol{\theta} + \tau\left(\boldsymbol{\theta}^{PG} - \boldsymbol{\theta}\right)\right) - \nabla\mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta}^{PG} - \boldsymbol{\theta}\right\rangle d\tau \\ &\quad \left(\text{Expanding, using the identity } -t\boldsymbol{g}^{PG} = \boldsymbol{\theta}^{PG} - \boldsymbol{\theta}\right) \\ &= \mathcal{L}(\boldsymbol{\theta}) - t\boldsymbol{g}^T \cdot \boldsymbol{g}^{PG} \\ &\quad + t^2 \int_0^1 \nabla\mathcal{L}(\boldsymbol{\theta})^T \cdot \nabla^2\mathcal{L}\left(\boldsymbol{\theta} + \tau\left(\boldsymbol{\theta}^{PG} - \boldsymbol{\theta}\right)\right)\nabla\mathcal{L}(\boldsymbol{\theta})d\tau.\end{aligned} \tag{16}$$

According to the assumption $\mathbf{H}(\mathcal{L}; \boldsymbol{\theta}, \boldsymbol{\theta}') \geq L\|\boldsymbol{g}\|^2$, we can obtain the following result:

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}^{PG}\right) &\geq \mathcal{L}(\boldsymbol{\theta}) - t\boldsymbol{g}^T \cdot \boldsymbol{g}^{PG} + L \cdot t^2 \cdot \|\boldsymbol{g}^{PG}\|^2 \\ &= \mathcal{L}(\boldsymbol{\theta}) + 2\left(2Lt^2 - t\right) \cdot \|\boldsymbol{g}\|^2.\end{aligned} \tag{17}$$

According to Theorem 1 and (8), we can obtain the simplified upper bound of $\mathcal{L}(\boldsymbol{\theta}^{MG})$

$$\mathcal{L}\left(\boldsymbol{\theta}^{MG}\right) \leq \mathcal{L}(\boldsymbol{\theta}) + 2 \cdot \left(Lt^2 - t\right)\|\boldsymbol{g}\|^2. \tag{18}$$

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.

According to (17) and (18), we have the following inequality:

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}^{MG}\right) - \mathcal{L}\left(\boldsymbol{\theta}^{PG}\right) &\le \mathcal{L}(\boldsymbol{\theta}) + 2\cdot\left(Lt^2 - t\right)\cdot\|\boldsymbol{g}\|^2 \\ &\quad - \mathcal{L}(\boldsymbol{\theta}) - 2\left(2Lt^2 - t\right)\cdot\|\boldsymbol{g}\|^2 \\ &= \left(-2t^2\right)\cdot\|\boldsymbol{g}\|^2 \le 0. \qquad (19)\end{aligned}$$

Therefore, we prove Theorem 3, which further shows the superiority of MAMG compared to PCGrad. As it is proved that PCGrad is superior to standard MTL methods, we can draw the conclusion that MAMG is superior to existing methods, which can be formally described by $\mathcal{L}(\boldsymbol{\theta}^{MG}) \le \mathcal{L}(\boldsymbol{\theta}^{PG}) \le \mathcal{L}(\boldsymbol{\theta}^{MT})$. ■

IV. Experimental Setup

A. Datasets and Tasks

To evaluate the effectiveness of MAMG, extensive experiments are conducted on three real-world datasets. The details of these three datasets as follows.

CityScapes: The CityScapes dataset [46] focuses on semantic understanding of urban street scenes and consists of high-resolution street-view images. We select two tasks from this dataset: 1) the Semantic Segmentation task and 2) the Depth Prediction task, described in [39]. Similar to [22], CityScapes is divided into training, validation, and testing data, 2975/125/500.

NYUv2: The NYUv2 dataset [47] consists of 1449 RGB-D images and 464 diverse indoor scenes with detailed annotations. We construct two training scenarios by using this dataset: 1) following [16], [17], we construct a two-task learning scenario by using the Semantic Segmentation and Surface Normal Estimation task and 2) according to [39], we consider the Depth Prediction, Semantic Segmentation, and Surface Normal Estimation task jointly. We adopt 40-class annotation for Semantic Segmentation and follow the common train/val splits: 795 images for training and 654 images for validation.

Taskonomy: Taskonomy [48] is a large-scale dataset that contains over 4.5 million indoor images from over 5000 buildings, with annotations available for 26 tasks. Similar to [22], we use the tiny version of Taskonomy, consisting of 38 1840 indoor images from 35 buildings, with annotations available for 26 tasks. Following [49], we construct a five-task learning scenario by selecting the Surface Normal Estimation, Edge Detection, Keypoint Detection, Semantic Segmentation, and Depth Prediction task from 26 tasks. We use the standard tiny split benchmark, which contains 274 883 training samples, 52 443 validation samples, and 54 514 testing images.

MultiMNIST: MultiMNIST dataset is an MTL version of the MNIST dataset [50], formed by overlaying multiple images together. We randomly select two images with different digits from the MNIST dataset, and then combine these two images to form a new image by putting one digit on the top-left corner and the other one on the bottom-right corner [2], [15]. Therefore, for each image of MultiMNIST dataset, we have two classification tasks: 1) classifying the digit on the top-left (task 1) and 2) classifying the digit on the bottom-right (task 2). We construct 60K images and 10K images for training and testing, respectively.

Multitask CelebA: CelebA includes 200K face images annotated with 40 attributes [51]. We view each attribute as a binary classification task and thus we convert it to a 40-way MTL problem following [15].

Multitask CIFAR-100: CIFAR-100 [52] includes 100 classes with 600 images each. Following [15], we treat 20 coarse labels in the original CIFAR-100 dataset as 20 tasks to construct Multitask CIFAR-100 dataset. Every task in Multitask CIFAR-100 dataset is a 5-way classification problem, with 2500 training images and 500 test images per task.

B. Evaluation Metrics

For different tasks, the evaluation metrics are different. We use the mean intersection over union (mIoU) and pixel accuracy (Pix Acc) to evaluate the Semantic Segmentation task. For the Surface Normal Estimation task, we use the mean and median angle distances of all the pixels for evaluation (the lower, the better). Moreover, we use the percentage of pixels that are within the angles of 11°, 22.5°, and 30° to the ground truth to evaluate the performance (the higher, the better) [53]. The performance of the Depth Prediction task is evaluated by absolute and relative errors (the lower, the better), and we evaluate the relative difference [54] between the prediction and ground truth by the percentage of $\delta = \max\{(y_{\text{pred}}/y_{gt}), (y_{gt}/y_{\text{pred}})\}$ within the threshold 1.25, 1.25^2, and 1.25^3 (the higher, the better). For the Taskonomy dataset, we compute the task-specific loss on test images as the evaluation metrics to measure the performance of each task [22]. Apart from reporting the absolute task performance with the above-mentioned metrics, we also compute the relative task performance $\Delta_{\mathcal{T}_i}$ with respect to the single-task baseline *STL* to evaluate the overall performance of each task $\mathcal{T}_i$ [22], [39]. The overall performance of each task can be evaluated as follows:

$$\Delta_{\mathcal{T}_i} = \frac{1}{m}\sum_{j=0}^{m}(-1)^{l_j}\left(M_{\mathcal{T}_i,j} - M_{\text{STL},j}\right)/M_{\text{STL},j} * 100\% \qquad (20)$$

where $l_j = 1$ if a lower value is better for the metric M_j and 0 otherwise; m is the number of the metrics; and $M_{\mathcal{T}_i,j}$ and $M_{STL,j}$ denote performance of task $\mathcal{T}_i$ and single-task baseline under the jth metric, respectively. To better evaluate the overall performance of compared baselines, we obtain the multitask performance by computing the average relative performance overall tasks by

$$\Delta_{\text{MTL}} = \frac{1}{T}\sum_{i=1}^{T}\Delta_{\mathcal{T}_i} \qquad (21)$$

where T is the number of tasks and Δ_{MTL} denotes the overall performance of the MTL model across all tasks.

C. Implementation Details

Following [22], [34], we use ResNet-34 [55] as our backbone and the ASPP decoder [56] as task-specific heads. For all datasets, the weight parameters are initialized by the Kaiming

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.

TABLE I
EXPERIMENTAL RESULTS OF SEMANTIC SEGMENTATION AND DEPTH PREDICTION TASKS IN CITYSCAPES

Model	Segmentation			Depth						
	(Higher Better)		(Higher Better)	Errors ↓ (Lower Better)		Within σ ↑ (Higher Better)			(Higher Better)	Δ_{MTL} ↑
	mIoU ↑	**PixAcc ↑**	$\Delta_{\mathcal{T}_{Seg}}$ ↑	**Abs Err**	**Rel Err**	**1.25**	**1.25^2**	**1.25^3**	$\Delta_{\mathcal{T}_{Depth}}$ ↑	
Single-Task	40.2	74.7	+0.0	0.017	0.33	70.3	86.3	93.3	+0.0	+0.0
Multi-Task	37.7	73.8	-3.7	0.018	0.34	72.4	88.3	94.2	-0.5	-2.1
Cross-Stitch [16]	42.5	56.2	-9.5	0.026	0.46	57.9	76.6	86.8	-26.1	-17.8
NDDR-CNN [17]	**48.8**	67.3	+5.8	0.024	0.38	61.5	80.1	89.8	-16.1	-5.1
PCGrad [15]	43.9	74.9	+3.4	0.018	0.33	70.4	87.9	94.3	-0.7	+1.4
AdaShare [22]	43.9	74.9	+4.8	0.079	0.33	71.4	87.9	94.1	-1.8	+1.5
AuxiLearn [25]	41.3	74.7	+1.3	0.035	0.40	44.8	84.3	92.7	-33.6	-16.1
MAMG(Ours)	45.8	**75.1**	**+7.3**	**0.015**	**0.32**	**76.2**	**90.2**	**95.2**	**+5.7**	**+6.5**

normal distribution $\mathcal{N}(0, \text{std}^2)$ [57], and the bias terms are initialized to zero. We use the Adam [58] approach to optimize the proposed MAMG. For the CityScapes and Taskonomy datasets, we set the learning rate of task-specific parameters and backbone parameters to 1e-3 and 1e-2, respectively. For the NYUv2 dataset, we set the learning rate to 1e-2. The batch size is set as 16. We also use L2 regularization (weight decay = 0.0001) to alleviate the overfitting problem. Similar to [22], [34], and [49], we use L1 loss for the Edge Detection, Keypoint Detection, and Depth Prediction tasks; Cross-entropy loss for the Semantic Segmentation task; and cosine similarity loss for the Surface Normal Estimation task. Note that all comparison methods use the same hyperparameter settings.

D. Compared Methods

To better illustrate the effectiveness of our method, we compare MAMG with some state-of-the-art baseline methods that design specific strategies to alleviate the gradient interference problem.

Single-Task Baseline: We use the same backbone and task-specific head to train each task separately, where each task has its own set of parameters. Following [22], we use the single-task performance as baseline to calculate the relative performance of each multitask model mentioned below.

Multitask Baseline: We also set a common multitask baseline (hard-parameters sharing structure), where all tasks share the backbone network but end up with separate task-specific head networks.

Cross-stitch networks [16] proposes a task-feature sharing mechanism by sharing the activations among all tasks. It mainly learns a linear combination of activations among all tasks to fuse shared information.

NDDR-CNN employs a dimensionality reduction mechanism to fuse activations among all tasks, which first concatenates the features with the same spatial resolution, and then applies a 1×1 convolution layer to reduce the number of channels and share the useful information by fusing the activations among all channels [17].

AdaShare employs the neural architecture search strategy to decide what to share across which tasks to achieve the best performance. This model focuses on learning a task-specific policy that can decide to share which layers for a given task during jointly training multitasks [22].

PCGrad develops a simple yet general gradient optimization strategy to alleviate the gradient conflicting problem, which projects a task's gradient onto the normal plane of the gradient of any other task [15]. It is similar to our proposed method, and thus we directly compare it under various settings.

AuxiLearn employs the multitask optimization strategy to learn nonlinear interactions among all tasks by applying implicit differentiation [25]. It can flexibly combine multiple task loss terms into a single coherent object function.

For a fair comparison, we utilize the same backbone network and task-specific head network with MAMG to implement Cross-Stitch Networks, NDDR-CNN, AdaShare, PCGrad, and AuxiLearn. Therefore, we plug the MAMG into Multitask baselines and then compare it with these models to illustrate the performance of MAMG. Because MTAN has a more complex network structure, we can not directly compare it. To further present the superiority of MAMG, we extend these models by plugging MAMG into some existing models and report the comparison results.

V. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of MAMG on different datasets, together with different numbers of tasks, to further illustrate the superiority of the proposed model under various scenarios. We conduct extensive experiments with the aim of answering the following questions: 1) Is MAMG competitive with other state-of-the-art methods on real-world datasets? 2) Is MAMG compatible with a wide range of tasks? 3) Can MAMG improve the performance of all learning tasks rather than only some of them? and 4) Is MAMG a general and model-agnostic approach?

A. Quantitative Results

Table I shows the performance of Semantic Segmentation and Depth Prediction tasks on the Cityscapes dataset under ten metrics. It is clear that MAMG outperforms all compared methods in 9 out of 10 metrics and obtains a 5% improvement over the state-of-the-art mothods. More specifically, some methods, such as Cross-Stitch, NDDR-CNN, and AuxiLearn, perform well in the Semantic Segmentation task,

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.

TABLE II
PERFORMANCE OF COMPARISON METHODS IN NYUv2 DATASET WITH TWO LEARNING TASKS: SEMANTIC SEGMENTATION AND SURFACE NORMAL ESTIMATION

Model	Segmentation			Surface Normal						
	mIoU ↑	PixAcc ↑	$\Delta_{\mathcal{T}_{Seg}}$ ↑	Angle Distance ↓		Within $t°$ ↑			$\Delta_{\mathcal{T}_{SN}}$ ↑	Δ_{MTL} ↑
				Mean	**Median**	**11.25°**	**22.5°**	**30°**		
Single-Task	27.8	58.5	+0.0	17.3	14.4	37.2	73.7	85.1	+0.0	+0.0
Multi-Task	22.6	55.0	-12.3	16.9	13.7	41.0	73.1	84.3	+3.1	-4.6
Cross-Stitch [16]	29.8	60.7	+5.6	18.1	15.7	27.6	73.6	86.5	-7.5	-0.9
NDDR-CNN [17]	31.6	62.1	+10.0	17.5	15.4	32.3	74.4	**86.9**	-3.6	+3.2
PCGrad [15]	30.3	61.3	+6.8	16.5	13.9	39.8	**74.8**	86.3	+3.6	+5.2
AdaShare [22]	30.7	61.7	+7.9	16.6	13.8	40.4	74.0	85.7	+3.6	+5.7
AuxiLearn [25]	23.1	54.5	-11.9	17.2	15.5	33.9	72.9	86.5	-3.0	-7.4
MAMG(Ours)	**32.0**	**62.7**	**+11.2**	**15.9**	**12.3**	**46.8**	73.6	84.4	**+9.4**	**+10.3**

TABLE III
PERFORMANCE OF COMPARISON METHODS IN NYUv2 DATASET WITH THREE LEARNING TASKS: SEMANTIC SEGMENTATION, SURFACE NORMAL ESTIMATION, AND DEPTH PREDICTION

Model	Segmentation			Surface Normal						Depth						
	mIoU ↑	PixAcc ↑	$\Delta_{\mathcal{T}_{Seg}}$ ↑	Angle Distance ↓		Within $t°$ ↑			$\Delta_{\mathcal{T}_{SN}}$ ↑	Errors ↓		Within σ ↑			$\Delta_{\mathcal{T}_{Depth}}$ ↑	Δ_{MTL} ↑
				Mean	**Median**	**11.25°**	**22.5°**	**30°**		**Abs**	**Rel**	**1.25**	**1.25^2**	**1.25^3**		
Single-Task	27.5	58.9	+0.0	17.5	15.2	34.9	73.3	85.7	+0.0	0.62	0.25	57.9	85.8	95.7	+0.0	+0.0
Multi-Task	24.1	57.2	-7.6	16.6	13.4	42.5	73.2	84.6	+7.5	0.58	0.23	62.4	88.2	96.5	+5.2	+1.7
Cross-Stitch [16]	26.0	57.5	-3.9	17.7	15.1	31.3	74.6	86.3	-1.7	0.69	0.25	47.8	81.9	95.5	-6.8	-4.1
NDDR-CNN [17]	28.5	59.7	+2.5	17.2	15.2	34.0	74.6	**87.3**	+0.5	0.64	0.23	52.3	86.2	96.9	-0.8	+0.7
PCGrad [15]	28.9	59.9	+3.5	16.6	14.3	38.1	**74.8**	86.3	+4.5	0.57	0.23	62.9	88.7	96.8	+6.1	+4.7
AdaShare [22]	30.6	61.1	+7.5	**16.3**	14.2	39.3	**74.8**	87.2	+6.0	0.57	0.21	63.6	89.7	97.4	+8.0	+7.2
AuxiLearn [25]	24.9	56.5	-6.7	17.1	15.4	35.1	72.7	86.4	+0.4	1.08	0.37	18.4	46.4	76.3	-51.3	-19.2
MAMG(Ours)	**32.4**	**63.6**	**+12.9**	16.7	**13.0**	**44.4**	72.2	83.3	**+8.3**	**0.53**	**0.20**	**65.6**	**90.7**	**97.7**	**+10.3**	**+10.5**

but obtain poor performance in the Depth Prediction task. The huge gap in performance between tasks comes from task conflicts, and these methods cannot deal with conflicts well. PCGrad and AdaShare use specific strategies to mitigate gradient interference and thereby obtain higher performance, but the results are not satisfactory.

Table II shows the performance of the Semantic Segmentation and Surface Normal Estimation tasks on the NYUv2 dataset under ten metrics. Our proposed MAMG achieves the best performance in 8 out of 10 metrics. Compared with seven existing models, MAMG has about 4.6% improvements compared with the second best method in overall model performance Δ_{MTL}. Moreover, MAMG has a balanced performance in all learning tasks, which shows the effectiveness of the proposed approach to mitigate gradient interference.

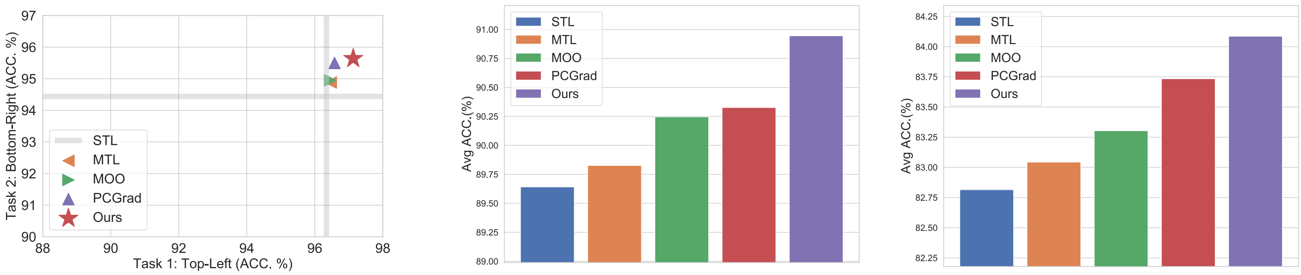
B. Effect of Multiple Task Sets

Our proposed MAMG is not only effective in the two-task learning scenario but is also applicable to multiple task sets, such as three-task learning and five-task learning scenarios. To show the effectiveness of MAMG in multiple tasks, we jointly train different task sets that include a different number of tasks from identical or different datasets.

We first construct a three-task learning scenario from the NYUv2 dataset by jointly training the Semantic Segmentation, Surface Normal Estimation, and Depth Prediction tasks. The results are shown in Table III, MAMG achieves the best performance in 13 out of 16 metrics. For each task, MAMG obtains the best overall performance. More concretely, methods, such as Cross-Stitch, NDDR-CNN, and AuxiLearn perform well in the Surface Normal task but obtain poor performance in the Semantic Segmentation and Depth Prediction tasks, especially in the Depth Prediction task. The huge gap in performance between tasks comes from the increase in task conflicts (an increase in the number of tasks brings more task conflicts). Therefore, these methods cannot deal with conflicts well when the number of tasks increases. PCGrad and AdaShare use specific strategies to improve the overall performance of the MTL model, but only obtain the second-best result. Therefore, the experimental results for the three-task learning scenario show that MAMG is superior in multiple learning task sets compared with existing methods.

Furthermore, we construct a five-task learning scenario by selecting five tasks from a larger real-world dataset, Taskonomy, with the aim of presenting the effectiveness of MAMG in the five-task learning scenario. We select the Surface Normal Estimation, Edge Detection, Keypoint Detection, Semantic Segmentation, and Depth Prediction tasks to construct a five-task learning experiment. We use the task-specific loss as the evaluation metrics to evaluate the performance of each task, such as cosine similarity for the Surface Normal task and errors for the Semantic Segmentation, Edge Detection, Keypoint Detection, and Depth Prediction

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c)

Fig. 3. Classification accuracy of different methods on three multitask datasets. We show classification accuracy (ACC.) on two tasks for MultiMNIST dataset. For CelebA and CIFAR-100 datasets, we report the average classification accuracy (Avg ACC.) of 40 tasks and 20 tasks, respectively. (a) MultiMNIST. (b) CelebA dataset. (c) CIFAR-100 dataset.

TABLE IV
Experimental Results on the Taskonomy Dataset

Model	Seg↓	Surface ↑	Depth↓	Keypoint ↓	Edge↓	Δ_{MTL} ↑
Single-Task	0.575	0.707	0.022	0.197	0.212	+0.0
Multi-Task	0.596	0.696	0.023	0.197	0.203	-1.1
Cross-Stitch	0.883	0.691	0.040	0.212	0.207	-29.4
NDDR-CNN	0.846	0.694	0.041	0.796	**0.196**	-25.0
PCGrad [15]	0.470	0.796	0.025	0.193	0.208	+4.2
AdaShare [22]	0.486	0.797	0.025	**0.190**	0.198	+5.1
MAMG (Ours)	**0.452**	**0.823**	**0.020**	0.194	0.202	**+9.7**

tasks. As shown in Table IV, MAMG achieves the best performance in 4 out of 6 metrics and obtains the best overall performance among the six comparison methods. According to the results reported in Tables II–IV, MAMG clearly obtains a better performance under different task sets compared with existing state-of-the-art methods. Moreover, MAMG has a more balanced performance across all training tasks, and it improves the performance in all tasks. Therefore, MAMG does not depend on manually selecting the learning task sets, and it is compatible with a wide range of tasks.

C. *Multilabel Classification Results*

In order to evaluate the performance of MAMG on Multilabel classification problems, we first conduct experiments on MultiMNIST [2], CelebA [2], and CIFAR-100 [15] datasets. We compare our MAMG with the vanilla MTL method, multiobjective optimization (MOO [2]), and PCGrad [15] to illustrate the effectiveness of MAMG. For the MultiMNIST experiments, we follow the same set-up of PCGrad [15] by using the LeNet network [59] as a task-shared network and using the fully connected layers as task-specific networks for each task. For the CelebA experiments, we follow [2] to use the ResNet-18 network [55] without the final layer as a task-shared network and employ 40 separate fully connected layers as task-specific networks for 40 tasks. In the CIFAR-100 experiments, we use the VGG-16 network [60] as the task-shared network and employ a fully connected layer for each task. For the fair comparison, all comparison methods use the same experimental setting.

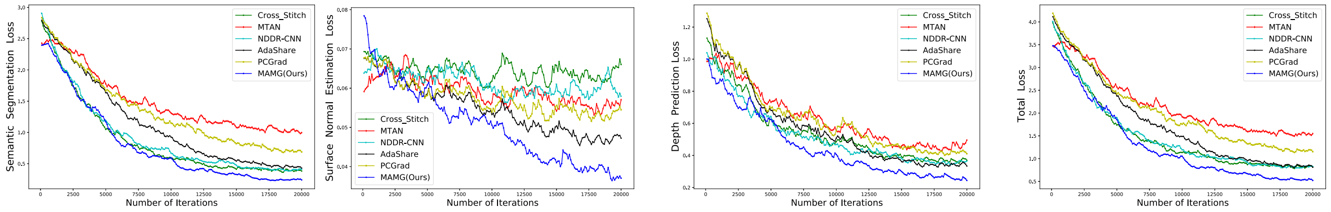
In Fig. 3(a), we show the classification accuracy of two tasks. We can observe that MAMG obtains the best performance on two tasks, which illustrates that our proposed MAMG can mitigate task conflicts and improve the performance of all tasks. As shown in Fig. 3(b), MAMG outperforms comparative methods across 40 binary classification tasks, which shows that MAMG is effective in multilabel classification problems and can improve the overall performance of the MTL model. Moreover, when the number of joint training tasks in the MTL CelebA dataset is high, our MAMG can also improve the performance of all tasks. Fig. 3(c) shows the average classification accuracy of 20 5-way classification tasks in the MTL CIFAR-100 dataset. It is clear that MAMG achieves the best performance on overall evaluation when the number of tasks is high and the tasks are more complex. This demonstrates the effectiveness of our proposed MAMG and can also mitigate the task conflicts in multilabel classification problems.

VI. Analysis

A. *Empirical Analysis on Convergence*

In the theoretical analysis part of Section III, we theoretically prove the superiority of our proposed method over existing MTL methods. Furthermore, we conduct extensive experiments about the convergence to better demonstrate the advantages of MAMG. Fig. 4(a) shows the training loss curves of the Semantic Segmentation task. It is clear that MAMG is constantly decreasing as the number of iterations increases. Fig. 4(b) shows the training loss curves of the Surface Normal Estimation task. We can find that MAMG is convergent, and loss value is constantly decreasing. Fig. 4(c) shows the training loss curves of the Depth Prediction task. We can find that MAMG has a similar convergent curve as the state-of-the-art methods, such as Cross-Stitch, NDDR-CNN, and AdaShare, which indicates that MAMG can guarantee the convergence of the objective. Fig. 4(d) shows the total training loss curves of these three learning tasks. MAMG is clearly superior to the existing methods because it can decrease faster than other comparison methods and obtains a lower objective value. Therefore, compared with other baselines, MAMG has a lower loss value in the three learning tasks, which shows that it can help multitask models converge to the minimizer of objective functions and obtain better performance.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c) (d)

Fig. 4. Learning curve of comparison methods about three learning tasks in the NYUv2 dataset. (a) Learning curve on Semantic Segmentation task. (b) Learning curve on the Surface Normal Estimation task. (c) Learning curve on the Depth Prediction task. (d) Total losses of the three learning tasks.

TABLE V
EXPERIMENTAL RESULTS ON THE NYUv2 DATASET WITH THREE LEARNING TASKS COMPARED WITH THE PCGRAD METHOD UNDER DIFFERENT STATE-OF-THE-ART METHODS

Architecture	Segmentation		Surface Normal					Depth				
	mIoU ↑	PixAcc ↑	Angle Distance ↓		Within $t°$ ↑			Errors ↓		Within σ ↑		
			Mean	**Median**	**11.25**	**22.5**	**30**	**Abs**	**Rel**	**1.25**	**1.25^2**	**1.25^3**
Cross-Stitch [16]	26.0	57.5	17.7	**15.1**	**31.3**	74.6	86.3	0.69	0.25	47.8	81.9	95.5
Cross-Stitch + PCGrad	**27.2**	**58.0**	17.8	15.4	29.3	74.6	**86.6**	0.78	0.26	41.9	78.0	94.6
Cross-Stitch + MAMG (Ours)	24.1	54.9	**17.6**	15.2	30.7	**75.0**	86.5	**0.66**	**0.25**	**51.7**	**84.3**	**96.1**
MTAN [39]	27.9	59.4	16.6	14.6	36.7	74.9	87.1	0.59	0.22	59.3	88.2	97.3
MTAN + PCGrad	28.5	60.6	16.6	14.7	36.7	**74.9**	**87.3**	0.58	0.21	60.1	88.7	97.3
MTAN + MAMG (Ours)	**31.1**	**63.2**	**16.5**	**14.4**	**37.9**	74.5	87.0	**0.58**	**0.21**	**60.6**	**89.0**	**97.3**
AdaShare [22]	30.6	61.1	**16.3**	14.2	39.3	**74.8**	**87.2**	0.57	0.21	63.6	89.7	97.4
AdaShare + PCGrad	30.1	60.7	16.6	14.6	37.7	74.3	86.9	0.57	0.22	63.3	89.6	97.2
AdaShare + MAMG (Ours)	**31.4**	**62.5**	16.6	**13.1**	**43.8**	72.6	83.9	**0.55**	**0.21**	**64.6**	**90.3**	**97.5**

B. Analysis of Model Independence

In order to better answer the question of whether MAMG is or is not a general and model-agnostic approach, we combine MAMG with several state-of-the-art MTL methods, namely, Cross-Stitch [16], MTAN [39], and AdaShare [22], and then evaluate the performance on the three challenging learning tasks in the challenging indoor scene dataset. The comparison results are reported in Table V. Applying MAMG can obtain better performance among all three tasks. More specifically, AdaShare with MAMG achieves the best performance in 9 out of 12 metrics across the Semantic Segmentation, Surface Normal Estimation, and Depth Prediction tasks. MTAN with MAMG achieves the best performance in 10 out of 12 metrics and is the second best in two metrics across three learning tasks. Therefore, MAMG is model-agnostic and can be plugged into most MTL networks to induce huge improvements in performance.

Furthermore, we conduct the experiments to analyze the convergence of these state-of-the-art methods with MAMG, such as MTAN and AdaShare, by plotting learning curves in Fig. 5 with respect to the gradient steps in the NYUv2 dataset. We can see that MAMG can indeed help MTL architectures converge faster and achieve a lower loss value compared with PCGrad methods [15]. In general, these results suggest that MAMG makes an improvement in performance and optimization speed. The improvement is caused by MAMG because it mitigates the gradient interference problem.

C. Ablation Studies on Extension to Other Architectures

To present that MAMG is model agnostic, we implement other network architectures, such as Wide ResNets (WRN) [61], and MobileNet-v2 [62], on the NYUv2 dataset. As shown in Table VI, MAMG outperforms the MTL baseline by 5.4% and 7.5% using MobileNet-v2 and WRN, respectively. This indicates the effectiveness of MAMG across different network architectures.

TABLE VI
DIFFERENT NETWORK ARCHITECTURES ON NYUV2 DATASET

Models	Architecture	$\Delta_{\mathcal{T}_{seg}}$ ↑	$\Delta_{\mathcal{T}_{Depth}}$ ↑	$\Delta_{\mathcal{T}_{MTL}}$ ↑
Multi-Task	**MobileNet-v2**	+0.69	+3.23	+1.96
MAMG(Ours)		**+8.85**	**+5.99**	**+7.42**
Multi-Task	**WRN**	-0.16	+6.99	+3.42
MAMG(Ours)		**+12.10**	**+9.80**	**+10.95**

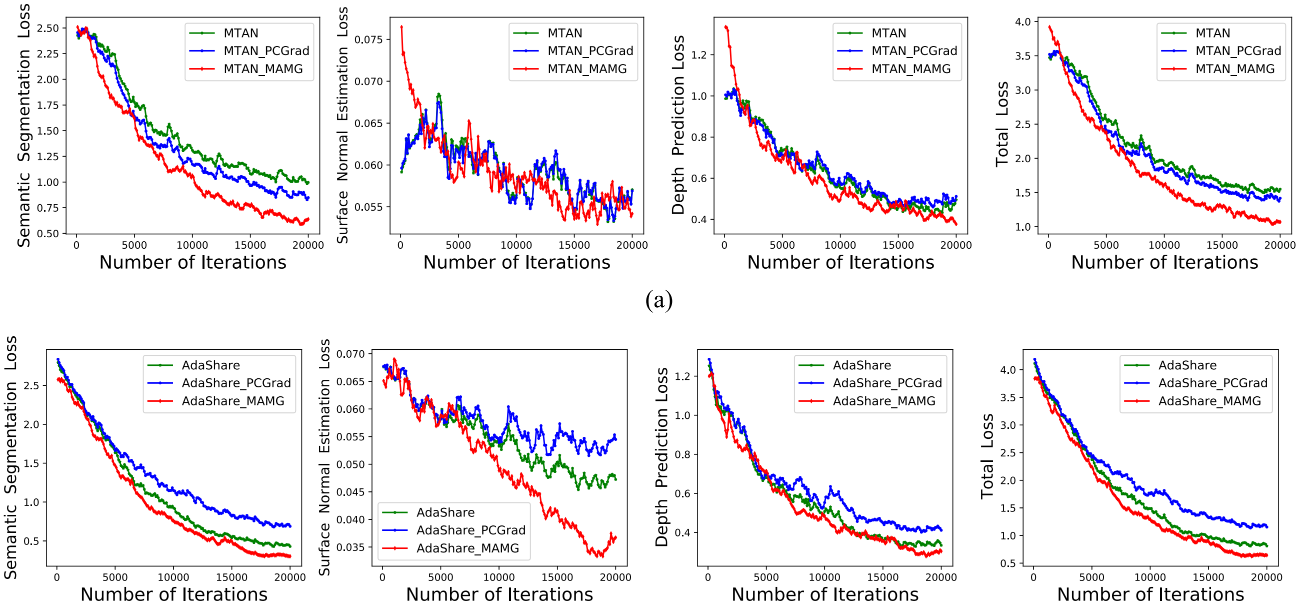
TABLE VII
EXPERIMENTAL RESULTS ON EXTENSION TO TEXT DATASET

Models	Architecture	Stance Task		Sentiment Task	
		$MacF_{avg}$	F_{avg}	ACC	F_{avg}
Multi-Task	**Bert**	0.536	0.646	0.783	0.796
MAMG(Ours)		**0.552**	**0.663**	**0.799**	**0.812**
Multi-Task	**GCN**	0.445	0.592	0.665	0.678
MAMG(Ours)		**0.456**	**0.603**	**0.674**	**0.684**

D. Ablation Studies on the Extension to Text Dataset

To further illustrate the effectiveness of MAMG, we extend MAMG to the text dataset, SemEval [63], which includes the stance detection task and the sentiment analysis task. Moreover, we implement MAMG using Bert and GCN [64].

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.



(a)

(b)

Fig. 5. Learning curves combining MAMG with two state-of-the-art networks. (a) MTAN network. (b) AdaShare network.

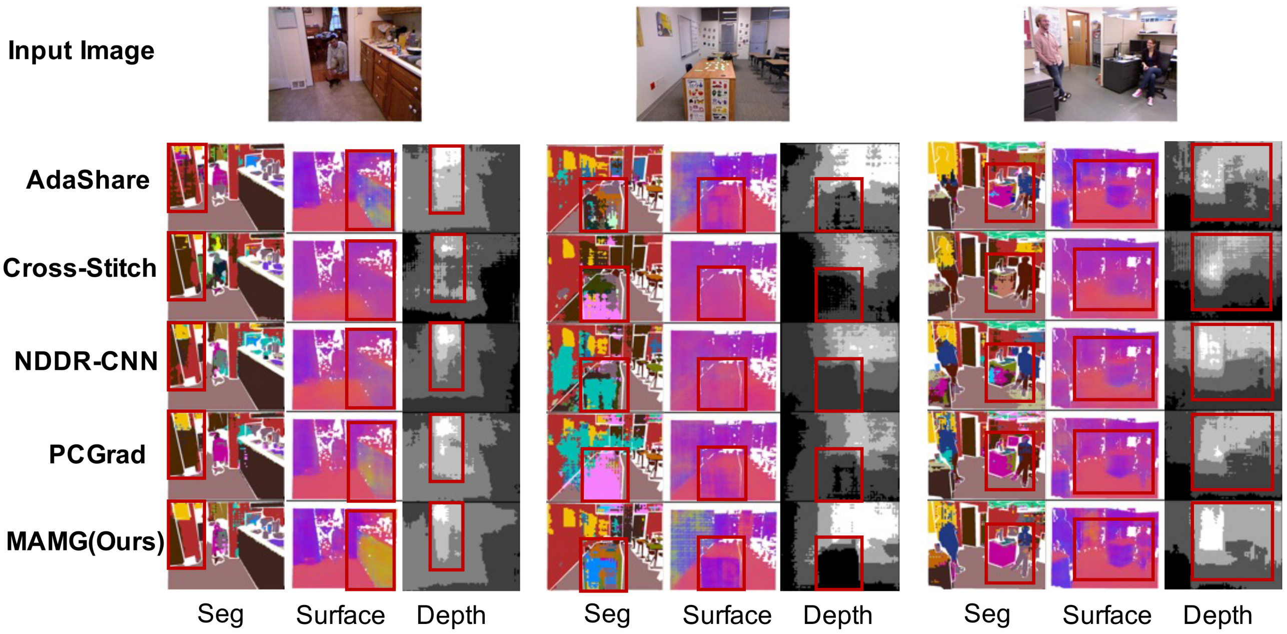


Fig. 6. Qualitative Visualization of AdaShare, Cross-Stitch, NDDR-CNN, PCGrad, and MAMG Performance in NYUv2 dataset. The red boxes are regions of interest, showing the effectiveness of the results provided from our method and other comparison methods. Our MAMG gives more accurate prediction and clearer contour in Semantic Segmentation (Seg), Surface Normal Prediction (Surface), and Depth Prediction (Depth). Best viewed in color.

As shown in Table VII, MAMG outperforms the MTL baseline overall metrics, which indicates that MAMG is practical in text datasets.

E. Qualitative Visualization

We visualize the results of Cross-Stitch, NDDR-CNN, AdaShare, PCGrad, and MAMG in a three-task learning scenario in NYUv2 dataset. As shown in Fig. 6, we can observe that MAMG can obtain the better predict results in Semantic Segmentation (Seg), Surface Normal Prediction (Surface), and Depth Prediction (Depth) tasks, where the edges of objects are clearly more pronounced. This implies the effectiveness of our proposed MAMG.

VII. Conclusion

In this article, we analyzed the gradient interference problem and provided formal definitions of the gradient interference. To alleviate these gradient interference issues, we proposed a novel approach (MAMG) that directly modifies the gradient of task. MAMG defines a gradient-interfering direction

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.

where different tasks may conflict with each other. According to the gradient component on conflict direction, we utilized the proposed *gradient clipping rule* to directly modify the task gradient to break the conditions of gradient interference. Moreover, we presented a series of theoretical proofs to illustrate the effectiveness of MAMG. Extensive experiments on six real-world datasets demonstrated the effectiveness of our approach in a wide range of task sets.

References

[1] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[2] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, 2018, pp. 525–536.

[3] Y. Lu, G. Lu, J. Li, Y. Xu, Z. Zhang, and D. Zhang, "Multiscale conditional regularization for convolutional neural networks," *IEEE Trans. Cybern.*, vol. 52, no. 1, pp. 444–458, Jan. 2022.

[4] J. Chen, B. Zhu, C. Ngo, T. Chua, and Y. Jiang, "A study of multi-task and region-wise deep learning for food ingredient recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 1514–1526, 2021.

[5] S. Chen, Z. Fang, S. Lu, and C. Gao, "Efficacy of regularized multitask learning based on SVM models," *IEEE Trans. Cybern.*, early access, Aug. 22, 2022, doi: 10.1109/TCYB.2022.3196308.

[6] G. Ma, J. Lu, F. Liu, Z. Fang, and G. Zhang, "Multiclass classification with fuzzy-feature observations: Theory and algorithms," *IEEE Trans. Cybern.*, early access, Jun. 27, 2022, doi: 10.1109/TCYB.2022.3181193.

[7] Y. Chen, D. Zhao, L. Lv, and Q. Zhang, "Multi-task learning for dangerous object detection in autonomous driving," *Inf. Sci.*, vol. 432, pp. 559–571, Mar. 2018.

[8] I. Leang, G. Sistu, F. Bürger, A. Bursuc, and S. K. Yogamani, "Dynamic task weighting methods for multi-task networks in autonomous driving systems," in *Proc. ITSC*, 2020, pp. 1–8.

[9] C. Zhang, E. Adeli, T. Zhou, X. Chen, and D. Shen, "Multi-layer multi-view classification for Alzheimer's disease diagnosis," in *Proc. AAAI*, 2018, pp. 4406–4413.

[10] M. Liu, J. Zhang, E. Adeli, and D. Shen, "Joint classification and regression via deep multi-task multi-channel learning for Alzheimer's disease diagnosis," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1195–1206, May 2019.

[11] S. Pang et al., "Beyond CNNs: Exploiting further inherent symmetries in medical image segmentation," *IEEE Trans. Cybern.*, early access, Aug. 31, 2022, doi: 10.1109/TCYB.2022.3195447.

[12] M. Yang, W. Huang, W. Tu, Q. Qu, Y. Shen, and K. Lei, "Multitask learning and reinforcement learning for Personalized dialog generation: An empirical study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 49–62, Jan. 2021.

[13] Q. Liao et al., "An integrated multi-task model for fake news detection," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5154–5165, Nov. 2022.

[14] J. Pang et al., "Fast supervised topic models for short text emotion detection," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 815–828, Feb. 2021.

[15] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," in *Proc. NeurIPS*, 2020, pp. 1–9.

[16] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proc. CVPR*, 2016, pp. 3994–4003.

[17] Y. Gao, J. Ma, M. Zhao, W. Liu, and A. L. Yuille, "NDDR-CNN: Layerwise feature fusing in multi-task CNNs by neural discriminative dimensionality reduction," in *Proc. CVPR*, 2019, pp. 3205–3214.

[18] C. Rosenbaum, T. Klinger, and M. Riemer, "Routing networks: Adaptive selection of non-linear functions for multi-task learning," in *Proc. ICLR*, 2018, pp. 1–6.

[19] A. Newell, L. Jiang, C. Wang, L. Li, and J. Deng, "Feature partitioning for efficient multi-task architectures," 2019, *arxiv.abs/1908.04339*.

[20] G. Strezoski, N. van Noord, and M. Worring, "Many task learning with task routing," in *Proc. ICCV*, 2019, pp. 1375–1384.

[21] T. Sun et al., "Learning sparse sharing architectures for multiple tasks," in *Proc. AAAI*, vol. 34, 2020, pp. 8936–8943.

[22] X. Sun, R. Panda, R. Feris, and K. Saenko, "AdaShare: Learning what to share for efficient deep multi-task learning," in *Proc. NeurIPS*, 2020, pp. 1–8.

[23] C. Liu, C. Zheng, S. Wu, Z. Yu, and H. Wong, "Multitask feature selection by graph-clustered feature sharing," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 74–86, Jan. 2020.

[24] H. Fei, S. Tan, and P. Li, "Hierarchical multi-task word embedding learning for synonym prediction," in *Proc. SIGKDD*, 2019, pp. 834–842.

[25] A. Navon, I. Achituve, H. Maron, G. Chechik, and E. Fetaya, "Auxiliary learning by implicit differentiation," in *Proc. Int. Conf. Learn. Rep. (ICLR)*, 2021, pp. 1–6.

[26] I. Kokkinos, "UberNet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse Datasets and limited memory," in *Proc. CVPR*, 2017, pp. 5454–5463.

[27] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. CVPR*, 2018, pp. 7482–7491.

[28] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Fast scene understanding for autonomous driving," in *Proc. DLVP*, 2017, pp. 1–5.

[29] M. Teichmann, M. Weber, J. M. Zöllner, R. Cipolla, and R. Urtasun, "MultiNet: Real-time joint semantic reasoning for autonomous driving," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2018, pp. 1013–1020.

[30] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. S. Feris, "Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification," in *Proc. CVPR*, 2017, pp. 1131–1140.

[31] S. Vandenhende, S. Georgoulis, L. V. Gool, and B. D. Brabandere, "Branched multi-task networks: Deciding what layers to share," in *Proc. BMVC*, 2020, pp. 1–2.

[32] D. Bruggemann, M. Kanakis, S. Georgoulis, and L. V. Gool, "Automated search for resource-efficient branched multi-task networks," in *Proc. BMVC*, 2020, pp. 1–8.

[33] M. Long, Z. Cao, J. Wang, and P. S. Yu, "Learning multiple tasks with multilinear relationship networks," in *Proc. NeurIPS*, 2017, pp. 1594–1603.

[34] P. Guo, C. Lee, and D. Ulbricht, "Learning to branch for multi-task learning," in *Proc. ICML*, vol. 119, 2020, pp. 3854–3863.

[35] F. J. S. Bragman, R. Tanno, S. Ourselin, D. C. Alexander, and M. J. Cardoso, "Stochastic filter groups for multi-task CNNs: Learning specialist and generalist convolution kernels," in *Proc. ICCV*, 2019, pp. 1385–1394.

[36] D. Xu, W. Ouyang, X. Wang, and N. Sebe, "PAD-Net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing," in *Proc. CVPR*, 2018, pp. 675–684.

[37] T. Sun et al., "Learning sparse sharing architectures for multiple tasks," in *Proc. AAAI*, 2020, pp. 8936–8943.

[38] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-Affinitive propagation across depth, surface normal and semantic segmentation," in *Proc. CVPR*, 2019, pp. 4106–4115.

[39] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proc. CVPR*, 2019, pp. 1871–1880.

[40] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," in *Proc. AAAI*, 2019, pp. 4822–4829.

[41] L. Zhou et al., "Pattern-structure diffusion for multi-task learning," in *Proc. CVPR*, 2020, pp. 4513–4522.

[42] Y. Du, W. M. Czarnecki, S. M. Jayakumar, R. Pascanu, and B. Lakshminarayanan. "Adapting auxiliary losses using gradient similarity." 2018. [Online]. Available: http://arxiv.org/abs/1812.02224

[43] D. Mahapatra and V. Rajan, "Multi-task learning with user preferences: Gradient descent with controlled ascent in Pareto optimization," in *Proc. ICML*, vol. 119, 2020, pp. 6597–6607.

[44] J. Lin, H. Liu, K. C. Tan, and F. Gu, "An effective knowledge transfer approach for multiobjective multitasking optimization," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3238–3248, 2021.

[45] Y. E. Nesterov and B. T. Polyak, "Cubic regularization of newton method and its global performance," *Math. Program.*, vol. 108, no. 1, pp. 177–205, 2006.

[46] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. CVPR*, 2016, pp. 3213–3223.

[47] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. ECCV*, 2012, pp. 746–760.

[48] A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proc. CVPR*, 2018, pp. 3712–3722.

[49] T. Standley, A. R. Zamir, D. Chen, L. J. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" in *Proc. ICML*, 2020, pp. 9120–9132.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.

[50] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3856–3866.

[51] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 3730–3738.

[52] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Handbook of Systemic Autoimmune Diseases*, vol. 1, 2009.

[53] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. ICCV*, 2015, pp. 2650–2658.

[54] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. NeurIPS*, 2014, pp. 2366–2374.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[56] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. ICCV*, 2015, pp. 1026–1034.

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–9.

[59] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Rep. (ICLR)*, 2015, pp. 1–9.

[61] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. BMVC*, 2016, pp. 1–9.

[62] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, 2018, pp. 4510–4520.

[63] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "SemEval-2016 task 6: Detecting stance in tweets," in *Proc. SemEval@NAACL-HLT*, 2016, pp. 31–41.

[64] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI*, 2019, pp. 7370–7377.

Heyan Chai received the M.S. degree in computer science and technology from the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China, in 2020, where he is currently pursuing the Ph.D. degree.

His research interests include data mining, multi-task learning, and text mining.

Zhe Yin received the B.S. degree in computer science and technology from the Department of Computer Science and Technology, Northeastern University, Shenyang, China, in 2019. He is currently pursuing the M.S. degree with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China.

His research interests include data mining and few-shot learning.

Ye Ding (Member, IEEE) received the Ph.D. degree in computer science and engineering from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2014 supervised by Prof. L. M. Ni.

He is currently an Associate Professor with the Dongguan University of Technology, Dongguan, China. His research interests are spatial-temporal data analytics and machine learning.

Li Liu (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2012.

She is currently a Full Professor with the College of System Engineering, NUDT. During her Ph.D. study, she spent more than two years as a visiting student with the University of Waterloo, Waterloo, ON, Canada, from 2008 to 2010. From 2015 to 2016, she spent ten months visiting the Multimedia Laboratory, The Chinese University of Hong Kong, Hong Kong. From December 2016 to November 2018, she worked as a Senior Researcher with the Machine Vision Group, University of Oulu, Oulu, Finland. Her current research interests include computer vision, pattern recognition, and machine learning.

Prof. Liu served as the Leading Guest Editor for special issues in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (IEEE TPAMI) and INTERNATIONAL JOURNAL OF COMPUTER VISION. She is serving as the Leading Guest Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE special issue on Learning with Fewer Labels in Computer Vision. She was a co-chair of nine International Workshops at CVPR, ICCV, and ECCV. Her papers have currently over 4000 citations in Google Scholar. She currently serves as an Associate Editor for *Pattern Recognition* and *Pattern Recognition Letter*. She serves as an Area Chair of ICME 2020, ICME 2021, and ACCV 2020.

Binxing Fang (Member, IEEE) received the M.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 1984, and the Ph.D. degree in computer science and technology from the Harbin Institute of Technology, Harbin, China, in 1989.

He is currently a Professor with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China. He is also a member of the Chinese Academy of Engineering, Beijing. His current research interests include computer networks, information and network security, and artificial intelligence security.

Qing Liao (Member, IEEE) received the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong, in 2016.

She is currently a Professor with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China, and also with the Department of New Networks, Peng Cheng Laboratory, Shenzhen. Her research interests include data mining, artificial intelligence, and information security.

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on December 06,2023 at 01:25:45 UTC from IEEE Xplore. Restrictions apply.