Beyond Losses Reweighting: Empowering Multi-Task Learning via the Generalization Perspective

Hoang Phan[1,*], Lam Tran[2,*], Quyen Tran[2], Ngoc Tran[3], Tuan Truong[2,†], Qi Lei[1], Nhat Ho[4], Dinh Phung[2,5], Trung Le[5]
[1]New York University, [2]Qualcomm AI Research[+], [3]Vanderbilt University, [4]University of Texas at Austin, [5]Monash University

Abstract

Multi-task learning (MTL) trains deep neural networks to optimize several objectives simultaneously using a shared backbone, which leads to reduced computational costs, improved data efficiency, and enhanced performance through cross-task knowledge sharing. Although recent gradient manipulation techniques aim to find a common descent direction that benefits all tasks, conventional empirical loss minimization still leaves models vulnerable to overfitting and gradient conflicts. To address this, we introduce a novel MTL framework that leverages weight perturbation to regulate gradient norms, thus improving generalization. By carefully modulating weight perturbations, our approach harmonizes task-specific gradients, reducing conflicts and encouraging more robust learning across tasks. Theoretical insights reveal that controlling the gradient norm through weight perturbation directly contributes to better generalization. Extensive experiments across diverse applications demonstrate that our method significantly outperforms existing gradient-based MTL techniques in terms of task performance and overall model robustness.

1. Introduction

Over the past few years, deep learning has emerged as a powerful tool for functional approximation, demonstrating superior performance and even surpassing human abilities in various applications. Despite these impressive achievements, training massive independent neural networks for individual tasks demands significant computational and storage resources, as well as extended runtime. Consequently, multi-task learning has become a preferable approach in many situations [47, 74, 80] as it aims to learn a shared network among tasks, reducing redundant feature calculations while promoting positive task transfer.

However, learning such a shared backbone faces performance degradation due to gradient conflict [78], where task-specific gradients may differ in direction and magnitude, resulting in tasks canceling each other and a subset of tasks being under-optimized. To tackle this, a common approach is to manipulate task gradients to find a better update direction so that all task losses decrease in a more balanced manner. This has been found to consistently exhibit improved performance [5, 45]. However, existing state-of-the-art methods in this vein [28, 44, 46, 62, 71, 78] often overlook the geometrical properties of the loss landscape, focusing solely on minimizing the empirical error in the optimization process, which can easily be prone to overfitting problems [30, 81].

Meanwhile, the overfitting problem in modern neural networks is often attributed to high-dimensional and nonconvex loss functions, which result in complex loss landscapes containing multiple local optima. Consequently, understanding the loss surface is crucial for training robust models, and developing flat minimizers remains one of the most effective strategies [30, 33, 40, 53]. Specifically, recent studies [24, 82] demonstrate that directly minimizing empirical risk often leads to a loss landscape with many sharp minima, resulting in poor generalization to unseen data. This issue is apparently further exacerbated when optimizing multiple objectives simultaneously, as in the context of multi-task learning. In fact, sharp minima of each constituent objective might appear at different locations, which can result in large generalization errors on the associated task. Therefore, finding a common flat and low-loss valued region for all tasks is desirable for improving the current methods of multi-task learning.

Contribution. To address the above desideratum, we propose a novel MTL training method that enhances existing gradient manipulation strategies by promoting the learning of flat loss landscapes across all tasks. Specifically, we penalize each task's sharpness, the gap between the largest and the empirical losses within a weight perturbation [20, 82], to improve the generalization of all tasks. This is theoretically

[*] Equal contributions.
[†] Work done while at Qualcomm AI Research.
[+] Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

supported by the generalization error in Theorem 1, which shows that our approach not only orients the model toward the joint low empirical loss value across tasks but also encourages the model to reach the task-based flat regions. Importantly, our approach is model-agnostic and compatible with current gradient-based MTL approaches (see Figure 1 for an overview of our approach). By using our proposed framework, gradient conflict across tasks is significantly mitigated, which is the goal of recent gradient-based MTL studies in alleviating negative transfer between tasks. Finally, we conduct comprehensive experiments on a variety of applications to demonstrate the merit of our approach for improving not only task performance but also model robustness and calibration. Last but not least, to the best of our knowledge, ours is the first work to improve multi-task learning by investigating the geometrical properties of the model loss landscape.

2. Related Work

2.1. Multi-task learning

In multi-task learning (MTL), we often aim to jointly train a single model to tackle multiple different but correlated tasks. It has been proven in prior work [8, 47, 48, 70] that it can not only enhance the overall performance but also reduce the memory usage and speed up the inference process. Previous studies on MTL often employ a hard parameter-sharing mechanism along with lightweight, task-specific modules to handle multiple tasks.

Pareto multi-task learning. Originating from Multiple-Gradient Descent Algorithm (MGDA), a popular line of gradient-based MTL methods aims to find Pareto stationary solutions, where it is impossible to improve model performance on any particular task without diminishing performance on another [71]. Moreover, recent studies suggest exploring the entire Pareto front by learning diverse solutions [42, 49, 54, 55, 68], or profiling the entire Pareto front with a hyper-network [43, 61]. While these methods are theoretically grounded and guaranteed to converge to Pareto-stationary points, the experimental results are often limited and lack comparisons in practical settings.

Loss and gradient balancing. Another branch of preliminary work in MTL capitalizes on the idea of dynamically reweighting loss functions based on gradient magnitudes [12], task homoscedastic uncertainty [31], or difficulty prioritization [23] to balance the gradients across tasks. More recently, PCGrad [78] introduces a gradient manipulation procedure to avoid conflicts among tasks by projecting random task gradients onto the normal plane of the other. Similarly, [44] proposes a provably convergent method to minimize the average loss, and [46] calculates loss-scaling coefficients such that the combined gradient has equal-length projections onto individual task gradients.

2.2. Flat minima

Flat minimizer has been found to improve the generalization ability of neural networks because it enables models to find wider local minima, by which they will be more robust against shifts between train and test losses [19, 29, 67]. This relationship between generalization ability and the width of minima is theoretically and empirically studied in many studies [17, 21, 26, 63], and subsequently, a variety of methods seeking flat minima have been proposed [10, 27, 32, 66].

Recently, SAM [20], which seeks flat regions by explicitly minimizing the worst-case loss around the current model, has received significant attention due to its effectiveness and scalability compared to previous methods. Particularly, it has been exploited in a variety of tasks and domains, such as domain generalization [9, 73, 79], federated learning [7, 69], Bayesian networks [59, 64], meta-learning [1]. In addition, SAM shows its generalization ability in vision models [11] and language models [4]. However, these studies have only focused on single-task problems. Closer to our setting are the works in [15, 77] that apply SAM to Continual Learning, but their focus is the relationship between flatness and catastrophic forgetting. In this work, we leverage SAM's principle to develop theory and devise practical methods, allowing for seeking flat minima in gradient-based multi-task learning models.

3. Methodology

This section outlines our proposed framework for enhancing gradient-based MTL methods. We begin by recalling the goal of multi-task learning and then establish upper bounds for the general loss of each task. Based on these bounds, we develop our framework to improve model generalization by guiding it toward flatter regions for each task.

3.1. Multi-task Learning and Gradient-based methods

In multi-task learning, we are given a data-label distribution $\mathcal{D}$ from which we can sample a training set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i^1, ..., y_i^m)_{i=1}^n\}$, where $\boldsymbol{x}_i$ is a data example and $y_i^1, ..., y_i^m$ are the labels of the tasks $1, 2, ..., m$ respectively.

The model for each task $\boldsymbol{\theta}^i = [\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}^i_{ns}]$ consists of the shared part $\boldsymbol{\theta}_{sh}$ and the individual non-shared part $\boldsymbol{\theta}^i_{ns}$. We denote the general loss for the task i by $\mathcal{L}^i_{\mathcal{D}}(\boldsymbol{\theta}^i)$, while its empirical loss over the training set $\mathcal{S}$ by $\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}^i)$. Existing work in MTL, typically MGDA [71], PCGrad [78], CAGrad [44], and IMTL [46], aim to find a model that simultaneously minimizes the empirical losses for all tasks:

$$\min_{\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}^{1:m}_{ns}} \left[\mathcal{L}^1_{\mathcal{S}}\left(\boldsymbol{\theta}^1\right), ..., \mathcal{L}^m_{\mathcal{S}}\left(\boldsymbol{\theta}^m\right)\right], \tag{1}$$

by calculating gradient $\boldsymbol{g}^i$ for i-th task ($i \in [m]$). The current model parameter is then updated using the unified

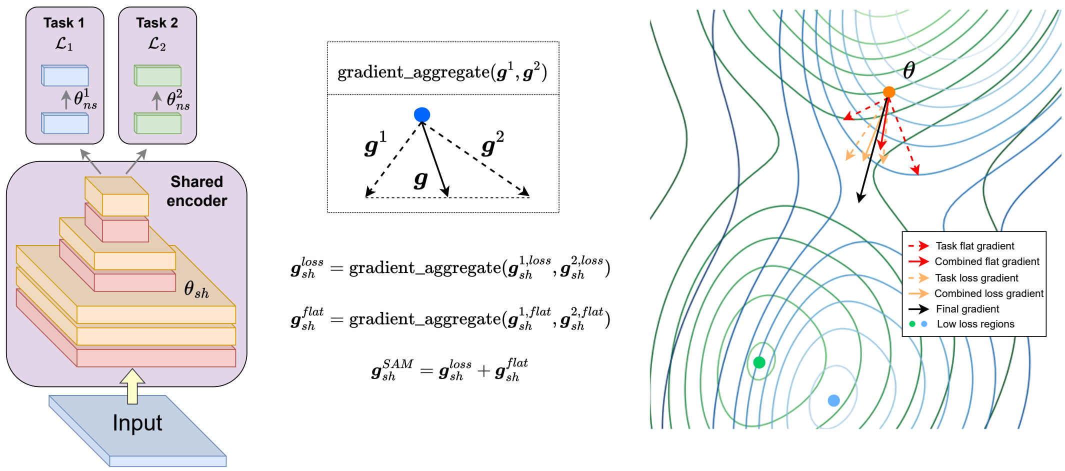

Figure 1. We demonstrate our framework in a 2-task problem. For the shared part, *task-based flat gradients* (red dashed arrows) steer the model away from sharp areas, while *task-based loss gradients* (orange dashed arrows) lead the model into their corresponding low-loss regions. In our method, we aggregate them to find the combined flat gradient $\boldsymbol{g}_{sh}^{flat}$ and combined loss gradient $\boldsymbol{g}_{sh}^{loss}$, respectively. Finally, we add these two output gradients to target the joint low-loss and flat regions across the tasks. Conversely, updating task-specific non-shared parts is straightforward and much easier as only one objective is involved.

gradient $\boldsymbol{g} = \text{gradient\_aggregate}(\boldsymbol{g}^1, \boldsymbol{g}^2, \ldots, \boldsymbol{g}^m)$, where the generic operation gradient_aggregate combines multiple task gradients, as proposed in gradient-based MTL studies. Details on this operation can be found in the Appendix.

However, prior works only focused on minimizing the empirical losses and tend to be overfitting. To alleviate this, inspired by [20, 35, 75, 82], it is desirable to develop sharpness-aware MTL approaches wherein the task models simultaneously seek low loss and flat regions, which is discussed below.

3.2. Sharpness minimization for MTL

Intuitively, flat minima are those where neighboring points also exhibit low loss values. One effective way to find such minima is to minimize the worst-case perturbation loss, as demonstrated in [20, 35]. Here, we propose applying this concept to each task objective in MTL. Formally, the worst-case loss for each task is defined as follows:

$$\max_{||\boldsymbol{\epsilon}_{sh}||_2 \leq \rho_{sh}} \left[\max_{||\boldsymbol{\epsilon}^i_{ns}||_2 \leq \rho_{ns}} \mathcal{L}^i_{\mathcal{S}} \left(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}^i_{ns} + \boldsymbol{\epsilon}^i_{ns} \right) \right]_{i=1}^m \tag{2}$$

where $|| \cdot ||_2$ denotes the l_2 norm, ρ_{sh} and ρ_{ns} represent the radii of the neighborhoods for the shared and non-shared parts, respectively.

The formulation of the worst-case loss in Eq. (2) differs from that in the single-task setting, as it involves multiple objective functions, each consisting of shared and individual non-shared parameters. This complexity makes extending the generalization error bound in [20] non-trivial. In the next sub-section, we provide such bounds for the true risks in the context of MTL, highlighting the concept of sharpness for the shared and non-shared parts.

3.3. Theoretical development

We informally state our main theorem that bounds the generalization performance of individual tasks by the empirical error on the training set:

Theorem 1. *For any perturbation radius $\rho_{sh}, \rho_{ns} > 0$, under the bounded-loss and mild assumptions, with probability $1 - \delta$ (over the choice of training set $\mathcal{S} \sim \mathcal{D}$) we obtain*

$$\left[\mathcal{L}^i_{\mathcal{D}} \left(\boldsymbol{\theta}^i\right)\right]_{i=1}^m \leq \max_{\|\boldsymbol{\epsilon}_{sh}\|_2 \leq \rho_{sh}} \Bigg[\max_{\|\boldsymbol{\epsilon}^i_{ns}\|_2 \leq \rho_{ns}} \mathcal{L}^i_{\mathcal{S}} \left(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}^i_{ns} + \boldsymbol{\epsilon}^i_{ns}\right) + f^i\left(\|\boldsymbol{\theta}^i\|_2^2\right) \Bigg]_{i=1}^m, \tag{3}$$

where $f^i : \mathbb{R}_+ \to \mathbb{R}_+, i \in [m]$ are strictly increasing functions.

Theorem 1 establishes the connection between the generalization error of each task and its empirical training error via worst-case perturbation on the parameter space. The formally stated theorem and proof are provided in the Appendix. We note that the worst-case shared perturbation $\boldsymbol{\epsilon}_{sh}$ is common for all tasks, while the worst-case non-shared perturbation $\boldsymbol{\epsilon}^i_{ns}$ is tailored for each task i. This requires addressing multiple objectives with both non-shared and shared components in our theory development.

Additionally and importantly, the proof in [20] invokes the PAC-Bayesian generalization bound [58]; hence, it only applies to the 0-1 loss in the binary classification setting. In contrast, as a theoretical contribution, we employ a more general PAC-Bayesian generalization bound [2], which only requires the loss to be bounded, thus tackling a notably wider range of losses in MTL. Hence, our theory development is not a trivial extension of prior works due to the nature of multi-objective optimization.

3.4. Practical method

Guided by Theorem 1, we first aim to solve the bi-level maximization problem for each task loss as follows:

$$\max_{\|\boldsymbol{\epsilon}_{sh}\|_2\le\rho_{sh}}\left[\max_{\|\boldsymbol{\epsilon}^i_{ns}\|_2\le\rho_{ns}}\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}_{sh}+\boldsymbol{\epsilon}_{sh},\boldsymbol{\theta}^i_{ns}+\boldsymbol{\epsilon}^i_{ns}\right)\right] \tag{4}$$

$$\approx\max_{\|\boldsymbol{\epsilon}_{sh}\|_2\le\rho_{sh}}\Bigg[\max_{\|\boldsymbol{\epsilon}^i_{ns}\|_2\le\rho_{ns}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})+ (\boldsymbol{\epsilon}^i_{ns})^{\mathbb{T}}\nabla_{\boldsymbol{\theta}^i_{ns}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})+(\boldsymbol{\epsilon}_{sh})^{\mathbb{T}}\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\Bigg], \tag{5}$$

$$=\max_{\|\boldsymbol{\epsilon}_{sh}\|_2\le\rho_{sh}}\Bigg[\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})+(\boldsymbol{\epsilon}_{sh})^{\mathbb{T}}\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns}) + \max_{\|\boldsymbol{\epsilon}^i_{ns}\|_2\le\rho_{ns}}(\boldsymbol{\epsilon}^i_{ns})^{\mathbb{T}}\nabla_{\boldsymbol{\theta}^i_{ns}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\Bigg] \tag{6}$$

where approximation is the first order Taylor expansion with a note that $\boldsymbol{\epsilon}_{sh}$ and $\boldsymbol{\epsilon}^i_{ns}$ are independent. Now following the dual norm problem as in [20], the solution for the inner maximization is

$$\boldsymbol{\epsilon}^{i,*}_{ns}=\frac{\rho_{ns}\nabla_{\boldsymbol{\theta}^i_{ns}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})}{\|\nabla_{\boldsymbol{\theta}^i_{ns}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\|_2} \tag{7}$$

Next, our goal is to find $\boldsymbol{\epsilon}_{sh}$ that simultaneously maximizes the following objectives

$$\max_{\|\boldsymbol{\epsilon}_{sh}\|_2\le\rho_{sh}}\Bigg[\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})+(\boldsymbol{\epsilon}_{sh})^{\mathbb{T}}\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns}) + \rho_{ns}\|\nabla_{\boldsymbol{\theta}^i_{ns}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\|_2\Bigg]_{i=1}^m \tag{8}$$

$$=\max_{\|\boldsymbol{\epsilon}_{sh}\|_2\le\rho_{sh}}\left[(\boldsymbol{\epsilon}_{sh})^{\mathbb{T}}\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\right]_{i=1}^m+const \tag{9}$$

$$\le\left[\max_{\|\boldsymbol{\epsilon}^i_{sh}\|_2\le\rho_{sh}}(\boldsymbol{\epsilon}^i_{sh})^{\mathbb{T}}\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\right]_{i=1}^m+const, \tag{10}$$

where $const$ is the constant independent of $\boldsymbol{\epsilon}_{sh}$. It is non-trivial to find the closed-form solution for this problem because the worst-cased perturbation $\boldsymbol{\epsilon}_{sh}$ is shared among the tasks (cf. Eq. (9)). We hence relax it by separately finding $\boldsymbol{\epsilon}^i_{sh}$ for each task (cf. Eq. (10)). Similarly to the inner maximization, we now have

$$\boldsymbol{\epsilon}^{i,*}_{sh}=\frac{\rho_{sh}\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})}{\|\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\|_2}. \tag{11}$$

Note that this relaxation gives us an upper bound for the optimal value of the maximization problem in Eq. (9) because $(\boldsymbol{\epsilon}^*_{sh})^{\mathbb{T}}\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\le(\boldsymbol{\epsilon}^{i,*}_{sh})^{\mathbb{T}}\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns}),\forall i=1,\cdots m$. Therefore, we aim to minimize this upper-bound multi-objectives in practice. Finally, substituting Eq. (7) and Eq. (11) back to Eq. (6), the bi-level maximization in Eq. (4) has the following approximate solution:

$$\left[\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}_{sh}+\boldsymbol{\epsilon}^{i,*}_{sh},\boldsymbol{\theta}^i_{ns}+\boldsymbol{\epsilon}^{i,*}_{ns}\right)\right]_{i=1}^m \tag{12}$$

$$\approx\left[\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\right]_{i=1}^m+\rho_{sh}\left[\|\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\|_2\right]_{i=1}^m + \rho_{ns}\left[\|\nabla_{\boldsymbol{\theta}^i_{ns}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns})\|_2\right]_{i=1}^m \tag{13}$$

The above equation shows that sharpness-aware minimization for MTL requires us to minimize the following sub-objectives for each task: (i) the conventional task-loss, (ii) the norm of gradient w.r.t the task-specific head, and (iii) the norm of gradient w.r.t the shared backbone. Minimizing the first objective leads the model toward local, but possibly sharp minima, which causes overfitting and severe task conflict. Hence, to reduce this, the last two objectives help steer the model away from such sharp minima, favoring flatter ones, i.e. gradient norm minimization seeks flat minima. However, in comparison with traditional MTL, we are tasked with more objectives, possibly leading to more conflicts between all of them. In the following, we will present how we solve this problem by separately treating each sub-objectives for all tasks.

3.4.1. Update the non-shared parts

Since each task i has its own task-specific head $\boldsymbol{\theta}^i_{ns}$, we can find flat minima for $\boldsymbol{\theta}^i_{ns}$ as follows:

$$\boldsymbol{g}^{i,\mathrm{SAM}}_{ns}=\nabla_{\boldsymbol{\theta}^i_{ns}}\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}_{sh}+\boldsymbol{\epsilon}^{i,*}_{sh},\boldsymbol{\theta}^i_{ns}+\boldsymbol{\epsilon}^{i,*}_{ns}\right) \approx\nabla_{\boldsymbol{\theta}^i_{ns}}\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}_{sh},\boldsymbol{\theta}^i_{ns}\right)\Big|_{\boldsymbol{\theta}_{sh}=\boldsymbol{\theta}_{sh}+\boldsymbol{\epsilon}^{i,*}_{sh},\boldsymbol{\theta}^i_{ns}=\boldsymbol{\theta}^i_{ns}+\boldsymbol{\epsilon}^{i,*}_{ns}},$$
$$\boldsymbol{\theta}^i_{ns}=\boldsymbol{\theta}^i_{ns}-\eta\boldsymbol{g}^{i,\mathrm{SAM}}_{ns} \tag{14}$$

Note that computing gradient directly on the gradient norm sub-objective requires the heavy computation of Hessian matrix. Instead, we resort to the perturbed loss (12) and approximate its gradient w.r.t $\boldsymbol{\theta}^i_{ns}$ as in Eq. (14). This procedure is similar to single-task SAM.

3.4.2. Update the shared part

This is a challenging task since as shown in Eq. (13), we have to find a common $\boldsymbol{\theta}_{sh}$ to not only reduce losses of all tasks, but also to reduce their gradient norms. Specifically, define $\mathcal{L}^i_{loss} := \mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}^i_{ns})$ and $\mathcal{L}^i_{flat} := \rho_{sh}\|\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}^i_{ns})\|_2 + \rho_{ns}\|\nabla_{\boldsymbol{\theta}^i_{ns}}\mathcal{L}^i_{\mathcal{S}}(\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}^i_{ns})\|_2$, we have $2 \times m$ objectives in total.

It has been shown in [73, 85] that there may exist conflict between $\mathcal{L}^i_{loss}$ and $\mathcal{L}^i_{flat}$, leading to a risk of increasing the loss when minimizing sharpness. This problem can worsen in the scope of MTL where conflicts can arise not only between task objectives (inter-conflict) but also between the two purposes of each task (intra-conflict). Inspired by this evidence and the inherently different goals between these two types of losses, we propose to consider them individually. Conceptually, we decompose the original MTL problem, MTL$[\mathcal{L}^i_{loss} + \mathcal{L}^i_{flat}]^m_{i=1}$, into two sub-MTLs, MTL$[\mathcal{L}^i_{loss}]^m_{i=1}$ + MTL$[\mathcal{L}^i_{flat}]^m_{i=1}$.

To solve each sub-MTL problem, we have to compute gradients of $\mathcal{L}^i_{loss}$ and $\mathcal{L}^i_{flat}$ w.r.t $\boldsymbol{\theta}_{sh}$. The former is straightforward, but the latter requires heavy Hessian computation. We bypass this by noticing from Eq. (13) that the perturbed loss is the sum of $\mathcal{L}^i_{loss}$ and $\mathcal{L}^i_{flat}$. Hence, the gradient of $\mathcal{L}^i_{flat}$ can be approximated by the difference between the gradient of the perturbed loss and that of $\mathcal{L}^i_{loss}$. Formally,

$$\boldsymbol{g}^{i,\text{SAM}}_{sh} = \nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}^{i,*}_{sh}, \boldsymbol{\theta}^i_{ns} + \boldsymbol{\epsilon}^{i,*}_{ns}\right)$$

$$\approx \nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}^i_{ns}\right)\Big|_{\boldsymbol{\theta}_{sh}=\boldsymbol{\theta}_{sh}+\boldsymbol{\epsilon}^{i,*}_{sh}, \boldsymbol{\theta}^i_{ns}=\boldsymbol{\theta}^i_{ns}+\boldsymbol{\epsilon}^{i,*}_{ns}}$$

$$\boldsymbol{g}^{i,loss}_{sh} = \nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}^i_{ns}\right), \tag{15}$$

$$\boldsymbol{g}^{i,flat}_{sh} = \boldsymbol{g}^{i,SAM}_{sh} - \boldsymbol{g}^{i,loss}_{sh}, \tag{16}$$

The purpose of the negative gradient $-\boldsymbol{g}^{i,\text{loss}}_{sh}$ is to orient the model to minimize the loss of the task i, while $-\boldsymbol{g}^{i,\text{flat}}_{sh}$ navigates the model to the task i's flatter region. Therefore, the gradients $\boldsymbol{g}^{i,\text{loss}}_{sh}$ share a similar nature, making them likely congruent. A similar relationship holds for $\boldsymbol{g}^{i,\text{flat}}_{sh}$. To solve each sub-MTL, following gradient-based MTL methods that aggregate gradients such that their conflict is reduced, we aim to find a common direction that leads the joint low-valued losses for all tasks and the joint flatter region for them as:

$$\boldsymbol{g}^{\text{loss}}_{sh} = \text{gradient\_aggregate}(\boldsymbol{g}^{1,\text{loss}}_{sh}, \ldots, \boldsymbol{g}^{m,\text{loss}}_{sh}),$$
$$\boldsymbol{g}^{\text{flat}}_{sh} = \text{gradient\_aggregate}(\boldsymbol{g}^{1,\text{flat}}_{sh}, \ldots, \boldsymbol{g}^{m,\text{flat}}_{sh}).$$

Finally, to combine two sub-MTL problems, we can similarly aggregate $\boldsymbol{g}^{\text{loss}}_{sh}$ and $\boldsymbol{g}^{\text{flat}}_{sh}$ based on gradient-based MTL methods. However, in practice, we find that simply adding them can work well in most cases (e.g., Ours vs Second-aggre in Table 6), so we adopt this strategy to save computation: $\boldsymbol{g}^{\text{SAM}}_{sh} = \boldsymbol{g}^{\text{loss}}_{sh} + \boldsymbol{g}^{\text{flat}}_{sh}$; $\boldsymbol{\theta}_{sh} = \boldsymbol{\theta}_{sh} - \eta\boldsymbol{g}^{\text{SAM}}_{sh}$.

The key steps of our proposed framework are summarized in Algorithm 1, and the overall schema of our proposed method is demonstrated in Figure 1.

Note that one can apply gradient-based methods to remove intra-conflict between $\boldsymbol{g}^{i,loss}_{sh}$ and $\boldsymbol{g}^{i,flat}_{sh}$ for each task to obtain $\boldsymbol{g}^i_{sh}$, then aggregate them once more time. This strategy, however, is extremely computationally expensive as we have to use gradient-aggregator $m+1$ times, and still results in similar performance compared to our method, i.e., Each-aggre vs Ours in Table 6.

Another approach is to directly aggregate $\boldsymbol{g}^{i,SAM}_{sh}$ of each task, i.e, ignoring the intra-conflict. This might still result in a higher level of gradient conflict than in our method which considers intra-conflict. We empirically demonstrate this in Figure 2 where the our strategy gains lower loss values and gradient norms than the direct strategy, and the effectiveness of our method in Table 5.

Algorithm 1 Sharpness minimization for multi-task learning

Input: Model parameter $\boldsymbol{\theta} = [\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}^{1:m}_{ns}]$, perturbation radius $\rho = [\rho_{sh}, \rho_{ns}]$, step size η and a list of m differentiable loss functions $\{\mathcal{L}^i\}^m_{i=1}$.
Output: Updated parameter $\boldsymbol{\theta}^*$

1: **for** task $i \in [m]$ **do**
2: Compute gradient $\boldsymbol{g}^{i,\text{loss}}_{sh}, \boldsymbol{g}^i_{ns} \leftarrow \nabla_{\boldsymbol{\theta}}\mathcal{L}^i(\boldsymbol{\theta})$
3: Worst-case perturbation direction
$\boldsymbol{\epsilon}^i_{sh} = \rho_{sh} \cdot \boldsymbol{g}^{i,\text{loss}}_{sh} / \left\|\boldsymbol{g}^{i,\text{loss}}_{sh}\right\|$, $\quad \boldsymbol{\epsilon}^i_{ns} = \rho_{ns} \cdot \boldsymbol{g}^i_{ns} / \left\|\boldsymbol{g}^i_{ns}\right\|$
4: Approximate SAM's gradient
$\boldsymbol{g}^{i,\text{SAM}}_{sh}, \boldsymbol{g}^{i,\text{SAM}}_{ns} = \nabla\mathcal{L}^i(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}^i_{sh}, \boldsymbol{\theta}^i_{ns} + \boldsymbol{\epsilon}^i_{ns})$
5: Compute flat gradient
$\boldsymbol{g}^{i,\text{flat}}_{sh} = \boldsymbol{g}^{i,\text{SAM}}_{sh} - \boldsymbol{g}^{i,\text{loss}}_{sh}$
6: **end for**
7: Calculate combined update gradients:
$\boldsymbol{g}^{\text{loss}}_{sh} = \text{gradient\_aggregate}(\boldsymbol{g}^{1,\text{loss}}_{sh}, \boldsymbol{g}^{2,\text{loss}}_{sh}, \ldots, \boldsymbol{g}^{m,\text{loss}}_{sh})$
$\boldsymbol{g}^{\text{flat}}_{sh} = \text{gradient\_aggregate}(\boldsymbol{g}^{1,\text{flat}}_{sh}, \boldsymbol{g}^{2,\text{flat}}_{sh}, \ldots, \boldsymbol{g}^{m,\text{flat}}_{sh})$
8: Calculate shared gradient update $\boldsymbol{g}^{\text{SAM}}_{sh} = \boldsymbol{g}^{\text{loss}}_{sh} + \boldsymbol{g}^{\text{flat}}_{sh}$
9: Update model parameter
$\boldsymbol{\theta}^* = [\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}^{1:m}_{ns}] - \eta[\boldsymbol{g}^{\text{SAM}}_{sh}, \boldsymbol{g}^{1:m,\text{SAM}}_{ns}]$

4. Experiments

Datasets and Baselines. Our proposed method is evaluated on four MTL benchmarks, including Multi-MNIST [42], CelebA [51] for visual classification, and NYUv2 [72], CityScapes [14] for scene understanding. Their descriptions can be found in Appendix C. We show how our framework boosts performance of gradient-based MTL methods by comparing *vanilla* MGDA [71], PCGrad [78], CAGrad [44], IMTL [46], NashMTL [62], FairGrad [5] to their flat-based versions F-MGDA, F-PCGrad, F-CAGrad, F-IMTL,

Method	MultiFashion			MultiMNIST			MultiFashion+MNIST		
	Task 1 ↑	Task 2 ↑	Average ↑	Task 1 ↑	Task 2 ↑	Average ↑	Task 1 ↑	Task 2 ↑	Average ↑
STL	87.10 ± 0.09	86.20 ± 0.06	86.65 ± 0.02	95.33 ± 0.08	94.16 ± 0.04	94.74 ± 0.06	98.40 ± 0.02	89.42 ± 0.03	93.91 ± 0.02
MGDA	86.76 ± 0.09	85.78 ± 0.36	86.27 ± 0.22	95.62 ± 0.02	94.49 ± 0.10	95.05 ± 0.06	97.24 ± 0.04	88.19 ± 0.13	92.72 ± 0.07
F-MGDA	**88.12 ± 0.11**	**87.35 ± 0.11**	**87.73 ± 0.09**	**96.37 ± 0.06**	**94.99 ± 0.06**	**95.68 ± 0.00**	**97.30 ± 0.09**	**89.26 ± 0.14**	**93.28 ± 0.03**
PCGrad	86.93 ± 0.17	86.20 ± 0.14	86.57 ± 0.12	95.71 ± 0.03	94.41 ± 0.02	95.06 ± 0.02	97.12 ± 0.16	88.45 ± 0.08	92.78 ± 0.11
F-PCGrad	**88.17 ± 0.14**	**87.35 ± 0.27**	**87.76 ± 0.07**	**96.49 ± 0.05**	**95.34 ± 0.10**	**95.92 ± 0.07**	**97.65 ± 0.06**	**89.35 ± 0.07***	**93.50 ± 0.01**
CAGrad	86.99 ± 0.17	86.04 ± 0.15	86.51 ± 0.16	95.62 ± 0.05	94.39 ± 0.04	95.01 ± 0.04	97.19 ± 0.06	88.18 ± 0.14	92.68 ±0.04
F-CAGrad	**88.19 ± 0.19**	**87.45 ± 0.13**	**87.82 ± 0.10***	**96.54 ± 0.02**	**95.36 ± 0.04**	**95.95 ± 0.01***	**97.82 ± 0.05**	**89.26 ± 0.22**	**93.54 ± 0.13***
IMTL	87.35 ± 0.22	86.45 ± 0.09	86.90 ± 0.15	95.93 ± 0.09	94.63 ± 0.13	95.28 ± 0.02	97.47 ± 0.06	88.46 ± 0.11	92.97 ± 0.03
F-IMTL	**88.1 ± 0.10**	**87.50 ± 0.04***	**87.80 ± 0.06**	**96.55 ± 0.07***	**95.16 ± 0.05**	**95.85 ± 0.05**	**97.59 ± 0.12**	**88.99 ± 0.08**	**93.29 ± 0.02**
NashMTL	86.91 ± 0.09	86.17 ± 0.03	86.54 ± 0.04	95.54 ± 0.00	94.49 ± 0.09	95.01 ± 0.05	97.00 ± 0.18	88.39 ± 0.16	92.70 ± 0.02
F-NashMTL	**88.21 ± 0.20***	**87.40 ± 0.03**	**87.81 ± 0.11**	**96.47 ± 0.03**	**95.40 ± 0.10***	**95.94 ± 0.05**	**97.63 ± 0.12**	**89.33 ± 0.12**	**93.48 ± 0.07**
FairGrad	86.85 ± 0.15	86.17 ± 0.07	86.51 ± 0.04	95.55 ± 0.15	94.29 ± 0.09	94.92 ± 0.11	97.02 ± 0.06	88.37 ± 0.16	92.70 ± 0.10
F-FairGrad	**88.05 ± 0.08**	**87.41 ± 0.14**	**87.73 ± 0.04**	**96.48 ± 0.07**	**95.34 ± 0.04**	**95.91 ± 0.05**	**97.94 ± 0.02***	**88.99 ± 0.18**	**93.46 ± 0.10**

Table 1. Evaluation of different methods on three Multi-MNIST datasets. Rows with flat-based minimizers are shaded. Bold numbers denote higher accuracy between flat-based methods and their baselines. * denotes the highest accuracy (except for STL as it unfairly exploits multiple neural networks). We also use arrows to indicate that the higher is the better (↑) or vice-versa (↓).

F-NashMTL and F-FairGrad. We also add a single-task learning (STL) baseline for each dataset.

4.1. Image classification

Multi-MNIST. Following [71], we set up three Multi-MNIST experiments with ResNet18 [25], namely: MultiFashion, MultiMNIST and MultiFashion+MNIST.

As summarized in Table 1, we can see that seeking flatter regions for all tasks can improve the performance of all the baselines across all three datasets. Especially, flat-based methods achieve the highest score for each task and for the average, outperforming STL by 1.2% on MultiFashion and MultiMNIST. We conjecture that the discrepancy between gradient update trajectories to classify digits from MNIST and fashion items from FashionMNIST has resulted in the fruitless performance of baselines, compared to STL on MultiFashion+MNIST. Even if there exists dissimilarity between tasks, our best obtained average accuracy when applying our method to CAGrad is just slightly lower than STL ($< 0.4\%$) while employing a single model only.

CelebA. CelebA [52] is a face dataset with 200K images and 40 attributes, forming a 40-class multi-label classification problem. Table 2 presents the average errors over 40 tasks, with Linear Scalarization (LS) and Uncertainty Weighting (UW) (Kendall, Gal, and Cipolla 2018) as additional baselines. The best results in each pair and overall are highlighted in bold and *, respectively. Even with a large number of tasks, flat region seeking consistently shows its advantages, with F-CAGrad achieving the lowest average error. Notably, when the optimizer considers flat minima, the performance gaps between PCGrad, IMTL, and CAGrad (8.23, 8.24 vs. 8.22) are smaller than those under conventional ERM training (8.69, 8.88, and 8.52). This suggests that better aggregation of task gradients, and thus reduced conflict, occurs when shared parameters approach a common flat region.

Method	STL	LS	UW	MGDA	PCGrad	CAGrad	IMTL
Vanilla	8.77	9.99	9.66	9.96	8.69	8.52	8.88
Flat-based	-	-	-	**9.22**	**8.23**	**8.22***	**8.24**

Table 2. Mean of error per category of MTL algorithms in multi-label classification on CelebA dataset.

4.2. Scene Understanding

Two datasets used here are NYUv2 [72] and CityScapes [14]. For these two experiments, we additionally include several recent MTL methods, namely, scale-invariant (SI), random loss weighting (RLW), Dynamic Weight Average (DWA) [47], GradDrop [13], and Nash-MTL [62] whose results are taken from [62] and details are in Appendix C. Also following the standard protocol used in [44, 47, 62], Multi-Task Attention Network [47] is employed on top of the SegNet architecture [3], our results are averaged over the last 10 epochs to align with previous work.

Evaluation metric. In this experiment, we handle different task types, each with its own metrics. We report the relative task improvement [57] to compare overall performance. Let M_i and S_i be the metrics obtained by the main and the single-task learning (STL) model, respectively. The relative task improvement on i-th task is mathematically given by: $\Delta_i := 100 \cdot (-1)^{l_i} \, (M_i - S_i)/S_i$, where $l_i = 1$ if a lower value for the i-th criterion is better and 0 otherwise. We depict our results by the average relative task improvement $\boldsymbol{\Delta m\%} = \frac{1}{m} \sum_{i=1}^{m} \Delta_i$.

NYUv2. Table 3 presents the results and relative improvements of each task over STL for different methods. Generally, the flat-based versions achieve comparable or

	Segmentation		Depth		Surface Normal					
					Angle Distance ↓		Within t° ↑			$\Delta m\%$ ↓
	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓	Mean	Median	11.25	22.5	30	
STL	38.30	63.76	0.6754	0.2780	25.01	19.21	30.14	57.20	69.15	0.00
LS	39.29	65.33	0.5493	0.2263	28.15	23.96	22.09	47.50	61.08	5.59
SI	38.45	64.27	0.5354	0.2201	27.60	23.37	22.53	48.57	62.32	4.39
RLW	37.17	63.77	0.5759	0.2410	28.27	24.18	22.26	47.05	60.62	7.78
DWA	39.11	65.31	0.5510	0.2285	27.61	23.18	24.17	50.18	62.39	3.57
UW	36.87	63.17	0.5446	0.2260	27.04	22.61	23.54	49.05	63.65	4.05
GradDrop	39.39	65.12	0.5455	0.2279	27.48	22.96	23.38	49.44	62.87	3.58
Nash-MTL	40.13	65.93	0.5261*	0.2171	25.26	20.08	28.4	55.47	68.15	−4.04
MGDA	**30.47**	**59.90**	**0.6070**	0.2555	24.88	19.45	29.18	56.88	69.36	1.38
F-MGDA	26.42	58.78	0.6078	**0.2353**	**24.34***	**18.45***	**31.64***	**58.86***	**70.50***	**−0.33**
PCGrad	38.06	64.64	0.5550	0.2325	27.41	**22.80**	**23.86**	**49.83**	**63.14**	3.97
F-PCGrad	**40.05**	**65.42**	**0.5429**	**0.2243**	**27.38**	23.00	23.47	49.35	62.74	**3.14**
CAGrad	39.79	65.49	0.5486	0.2250	26.31	21.58	25.61	52.36	65.58	0.20
F-CAGrad	**40.93***	**66.68***	**0.5285**	**0.2162**	**25.43**	**20.39**	**27.99**	**54.82**	**67.56**	**−3.78**
IMTL	39.35	65.60	0.5426	0.2256	26.02	21.19	26.2	53.13	66.24	−0.76
F-IMTL	**40.42**	**65.61**	**0.5389**	**0.2121***	**25.03**	**19.75**	**28.90**	**56.19**	**68.72**	**−4.77***

Table 3. Test performance for three-task NYUv2 of Segnet [3]: semantic segmentation, depth estimation, and surface normal. Using the proposed procedure with gradient-based multi-task learning methods consistently improves their overall performance.

higher results on most metrics, except for F-MGDA in the segmentation task, where it notably decreases the $mIoU$ score. However, F-MGDA significantly boosts performance in other tasks, raising MGDA's overall relative improvement from -1.38% to +0.33% above STL. Notably, F-CAGrad and F-IMTL outperform competitors by large margins across all tasks, with top relative improvements of 3.78% and 4.77%, respectively.

	Segmentation		Depth		
Method	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err↓	$\Delta m\%$ ↓
STL	74.01	93.16	0.0125	27.77	
LS	75.18	93.49	0.0155	46.77	22.60
SI	70.95	91.73	0.0161	33.83	14.11
RLW	74.57	93.41	0.0158	47.79	24.38
DWA	75.24	93.52	0.0160	44.37	21.45
UW	72.02	92.85	0.0140	30.13*	5.89
GradDrop	75.27	93.53	0.0157	47.54	23.73
Nash-MTL	75.41	93.66	0.0129	35.02	6.82
MGDA	68.84	91.54	0.0309	33.50	44.14
F-MGDA	**73.77**	**93.12**	**0.0129**	**27.44***	**0.67***
PCGrad	75.13	93.48	0.0154	42.07	18.29
F-PCGrad	**75.77**	**93.67**	**0.0144**	**39.60**	**13.65**
CAGrad	75.16	93.48	0.0141	37.60	11.64
F-CAGrad	**76.02**	**93.72**	**0.0134**	**34.64**	**7.25**
IMTL	75.33	93.49	0.0135	38.41	11.10
F-IMTL	**76.63***	**93.76***	**0.0124***	**31.17**	**1.87**

Table 4. Test performance for two-task CityScapes: semantic segmentation and depth estimation.* denotes the best score for each task's metrics.

CityScapes. In Table 4, the positive impact of seeking flat regions is evident across all metrics and baselines. Notably, MGDA and IMTL show significant relative improvements, achieving the highest and second-best $\Delta m\%$ scores, respectively. PCGrad, CAGrad, and IMTL even surpass STL in segmentation scores. Interestingly, MGDA biases to the depth estimation objective, leading to the predominant performance of F-MGDA on that task, consistent with patterns observed in [46] and the NYUv2 experiment.

4.3. Ablation study

Here, we provide experimental justification for our gradient decomposition, and our method's impact on conventional MTL training by examining task conflict. Appendix D provides additional results for model calibration D.2, model robustness D.5, D.8, gradient norms D.9, loss landscape visualization D.4, training curves D.7, and hyper-param sensitivity D.3.

Directly aggregating SAM gradients neglects intra-conflict. Table 5 compares between the direct aggregation on $\{\boldsymbol{g}_{sh}^{i,\text{SAM}}\}_{i=1}^{m}$ and our individual aggregation on $\{\boldsymbol{g}_{sh}^{i,\text{flat}}\}_{i=1}^{m}$, and $\{\boldsymbol{g}_{sh}^{i,\text{loss}}\}_{i=1}^{m}$.

	Segmentation		Depth		
Method	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err↓	$\Delta m\%$ ↓
ERM	68.84	91.54	0.0309	33.50	44.14
Ours (**direct**)	68.93	91.41	0.0130	31.37	6.43
Ours (**individual**)	**73.77**	**93.12**	**0.0129**	**27.44***	**0.67**

Table 5. Direct SAM gradients aggregation vs our proposed gradient aggregation strategy on CityScapes.

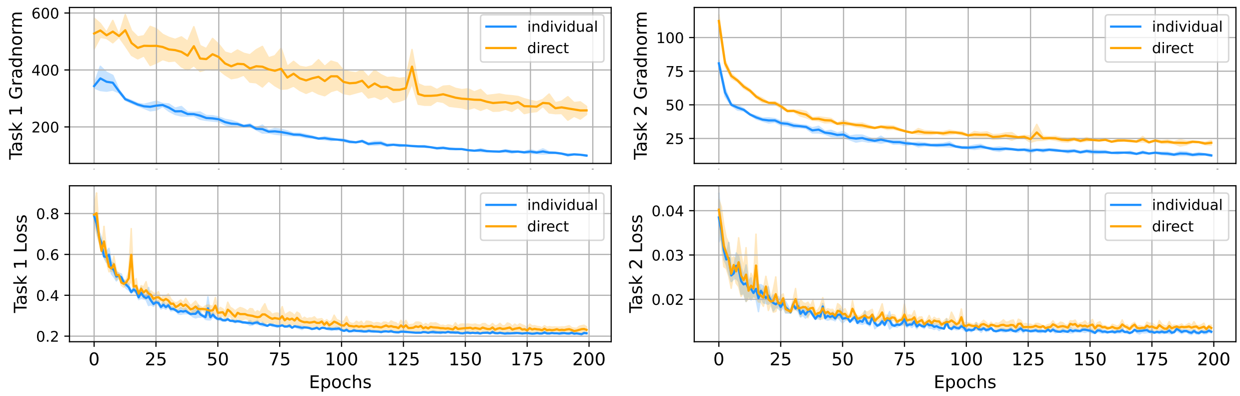

Figure 2. Evolution of gradient norms and task losses across aggregation strategies.

Compared to the naive approach, in which per-task SAM gradients are directly aggregated, our decomposition approach consistently improves performance by a large margin across all tasks. Moreover, using our decomposed SAM yields flatter minima and lower loss values throughout the whole training process, as shown in Figure 2. These results reinforce the rationale behind separately aggregating low-loss directions and flat directions.

Other approaches to apply gradient aggregation. Table 6 summarizes the performance of Second-aggre, Each-aggre and Our aggregation strategies: Second-aggre performs an additional step to aggregate $\boldsymbol{g}_{sh}^{loss}$ and $\boldsymbol{g}_{sh}^{flat}$; Each-aggre iteratively aggregates $\boldsymbol{g}_{sh}^{i,loss}$ and $\boldsymbol{g}_{sh}^{i,flat}$ to obtain $\boldsymbol{g}_{sh}^{i}$ for each task, then aggregate $\boldsymbol{g}_{sh}^{i}$ one more time. Our strategy requires less aggregation steps yet still achieves comparable performance.

Method	MultiFashion			MultiMNIST			MultiFashion+MNIST		
	Task 1 ↑	Task 2 ↑	Avg ↑	Task 1 ↑	Task 2 ↑	Avg ↑	Task 1 ↑	Task 2 ↑	Avg ↑
Second-aggre	88.10	87.70	87.90	96.48	95.24	95.86	97.87	89.02	93.44
Each-aggre	87.51	87.68	87.59	96.45	95.40	95.92	97.61	89.17	93.39
Ours	88.19	87.45	87.82	96.54	95.36	95.95	97.82	89.26	93.54

Table 6. Different aggregation strategies applied on CAGrad, on Multi-MNIST datasets.

Our improvement is not a mere result of single-task SAM. To show this point, we provide the results of STL and linear scalarization (LS), casting MTL as a single objective, trained with and without SAM on NYUv2 in Table 7. When equipped with SAM, F-STL and F-LS improve almost all scores of their counterparts. However, they can not consistently exceed flat gradient-based MTL baselines, which take gradient conflict into account. Particularly, on Segmentation and Depth tasks, F-IMTL achieves the highest scores, while on the Surface Normal task, F-MGDA is the best method. Overall, F-IMTL obtains the best $\boldsymbol{\Delta m\%}$.

Task conflict. To empirically confirm reduced gradient conflict in flat regions, we measured the proportion of minibatches with gradient conflict at each epoch, and present

	Segmentation		Depth		Surface Normal					
	mIoU ↑	Pix Acc ↑	Abs Err ↓	Rel Err ↓	Angle Distance ↓ Mean	Median	Within t° ↑ 11.25	22.5	30	$\Delta m\%$ ↓
STL	38.30	63.76	0.6754	0.2780	25.01	19.21	30.14	57.20	69.15	0.00
F-STL	39.07	64.21	0.6183	0.2514	25.08	18.72	30.99	58.29	69.76	−3.17
LS	39.29	65.33	0.5493	0.2263	28.15	23.96	22.09	47.50	61.08	5.59
F-LS	40.28	65.30	0.5360	0.2173	27.14	22.56	24.45	50.29	63.57	1.65
F-MGDA	26.42	58.78	0.6078	0.2353	24.34	18.45	31.64	58.86	70.50	−0.33
F-IMTL	40.42	65.61	0.5389	0.2121	25.03	19.75	28.90	56.19	68.72	−4.77

Table 7. Test performance for three-task **NYUv2** of Segnet.

the results in Figure 3. While ERM's gradient conflict rises above 50%, ours decreases and approaches 0%. This reduction is also a key objective of recent gradient-based MTL methods aimed at mitigating negative transfer between tasks [74, 78, 84].

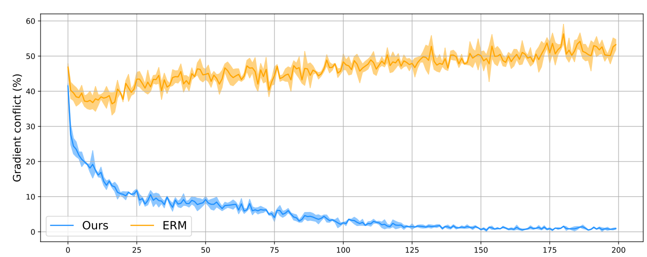

Figure 3. Proportion of conflict between per-task gradients ($\boldsymbol{g}^{1,\text{loss}} \cdot \boldsymbol{g}^{2,\text{loss}} < 0$) on Multi-MNIST.

5. Conclusion

In this work, we have presented a general framework that can be incorporated into current multi-task learning methods following the gradient balancing mechanism. The core ideas of our proposed method are the employment of flat minimizers in the context of MTL and proving that they can help enhance previous works both theoretically and empirically. Concretely, our method goes beyond optimizing per-task objectives solely to yield models that have both low errors and high generalization capabilities.

6. Acknowledgements

We thank Chau Pham for his contributions to the scene understanding experiments. This work was supported by ARC DP23 grant DP230101176 and by the Air Force Office of Scientific Research under award number FA2386-23-1-4044.

References

[1] Momin Abbas, Quan Xiao, Lisha Chen, Pin-Yu Chen, and Tianyi Chen. Sharp-maml: Sharpness-aware model-agnostic meta learning. *arXiv preprint arXiv:2206.03996*, 2022. 2

[2] Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of gibbs posteriors. *Journal of Machine Learning Research*, 17(236):1–41, 2016. 4, 1

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 6, 7, 5

[4] Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7360–7371, Dublin, Ireland, 2022. Association for Computational Linguistics. 2

[5] Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. In *Forty-first International Conference on Machine Learning*, 2024. 1, 5

[6] Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950. 7

[7] Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *European Conference on Computer Vision*, pages 654–672. Springer, 2022. 2

[8] Rich Caruana. Multitask learning. *Machine learning*, 28(1): 41–75, 1997. 2

[9] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021. 2

[10] Pratik Chaudhari, Anna Choromańska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019, 2017. 2

[11] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021. 2

[12] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018. 2

[13] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020. 6, 5

[14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5, 6

[15] Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. Flattening sharpness for dynamic gradient projection memory benefits continual learning. *Advances in Neural Information Processing Systems*, 34: 18710–18721, 2021. 2

[16] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012. 4

[17] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017. 2

[18] Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, R. Goh, and Vincent Y. F. Tan. Efficient sharpness-aware minimization for improved training of neural networks. *International Conference on Learning Representations*, 2021. 11

[19] Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *UAI*. AUAI Press, 2017. 2

[20] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 4, 5

[21] Stanislav Fort and Surya Ganguli. Emergent properties of the local geometry of neural loss landscapes. *arXiv preprint arXiv:1910.05929*, 2019. 2

[22] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 7

[23] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 270–287, 2018. 2

[24] Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. *Advances in neural information processing systems*, 32, 2019. 1

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 5, 7

[26] Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. In *NIPS*, pages 529–536. MIT Press, 1994. 2

[27] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, pages 876–885. AUAI Press, 2018. 2, 9

[28] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. In *International Conference on Learning Representations*, 2021. 1

[29] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *ICLR*. OpenReview.net, 2020. 2

[30] Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J Kusner. A fair comparison of two popular flat minima optimizers: Stochastic weight averaging vs. sharpness-aware minimization. *arXiv preprint arXiv:2202.00661*, 1, 2022. 1, 9

[31] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018. 2

[32] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*. OpenReview.net, 2017. 2

[33] Nitish Shirish Keskar, Jorge Nocedal, Ping Tak Peter Tang, Dheevatsa Mudigere, and Mikhail Smelyanskiy. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017*, 2017. 1

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[35] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks, 2021. 3, 5

[36] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 7

[37] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000. 2, 3

[38] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[39] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018. 9

[40] Zhouzi Li, Zixuan Wang, and Jian Li. Analyzing sharpness along gd trajectory: Progressive sharpening and edge of stability. *arXiv preprint arXiv:2207.12678*, 2022. 1

[41] Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor W Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *arXiv preprint arXiv:2111.10603*, 2021. 5

[42] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in neural information processing systems*, 32, 2019. 2, 5

[43] Xi Lin, Zhiyuan Yang, Qingfu Zhang, and Sam Kwong. Controllable pareto multi-task learning. *arXiv preprint arXiv:2010.06313*, 2020. 2

[44] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34: 18878–18890, 2021. 1, 2, 5, 6, 4

[45] Bo Liu, Yihao Feng, Peter Stone, and qiang liu. FAMO: Fast adaptive multitask optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1

[46] Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2020. 1, 2, 5, 7, 4

[47] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019. 1, 2, 6, 5

[48] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496. Association for Computational Linguistics, 2019. 2

[49] Xingchao Liu, Xin Tong, and Qiang Liu. Profiling pareto front with multi-objective stein variational gradient descent. *Advances in Neural Information Processing Systems*, 34: 14721–14733, 2021. 2

[50] Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12360–12370, 2022. 11

[51] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 5

[52] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018. 6, 5

[53] Kaifeng Lyu, Zhiyuan Li, and Sanjeev Arora. Understanding the generalization benefit of normalization layers: Sharpness reduction. *arXiv preprint arXiv:2206.07085*, 2022. 1

[54] Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *International Conference on Machine Learning*, pages 6597–6607. PMLR, 2020. 2

[55] Debabrata Mahapatra and Vaibhav Rajan. Exact pareto optimal search for multi-task learning: Touring the pareto front. *arXiv preprint arXiv:2108.00597*, 2021. 2

[56] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018. 7

[57] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1851–1860, 2019. 6

[58] David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999. 3

[59] Thomas Möllenhoff and Mohammad Emtiyaz Khan. SAM as an optimal relaxation of bayes. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[60] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. 7

[61] Aviv Navon, Aviv Shamsian, Gal Chechik, and Ethan Fetaya. Learning the pareto front with hypernetworks. In *International Conference on Learning Representations*, 2021. 2

[62] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022. 1, 5, 6

[63] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017. 2

[64] Van-Anh Nguyen, Tung-Long Vuong, Hoang Phan, Thanh-Toan Do, Dinh Phung, and Trung Le. Flat seeking bayesian neural networks. *Advances in Neural Information Processing Systems*, 2023. 2

[65] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019. 7

[66] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR (Workshop)*. OpenReview.net, 2017. 2

[67] Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. In *NeurIPS*, pages 18420–18432, 2021. 2

[68] Hoang Phan, Ngoc Tran, Trung Le, Toan Tran, Nhat Ho, and Dinh Phung. Stochastic multiple target sampling gradient descent. *Advances in neural information processing systems*, 35:22643–22655, 2022. 2

[69] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. *arXiv preprint arXiv:2206.02618*, 2022. 2

[70] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2

[71] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018. 1, 2, 5, 6, 4

[72] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 5, 6

[73] Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. Sharpness-aware gradient matching for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3769–3778, 2023. 2, 5

[74] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2020. 1, 8

[75] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in neural information processing systems*, 33:2958–2969, 2020. 3

[76] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[77] Enneng Yang, Li Shen, Zhenyi Wang, Shiwei Liu, Guibing Guo, and Xingwei Wang. Data augmented flatness-aware gradient projection for continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5630–5639, 2023. 2

[78] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020. 1, 2, 5, 8, 4

[79] Xingxuan Zhang, Renzhe Xu, Han Yu, Yancheng Dong, Pengfei Tian, and Peng Cui. Flatness-aware minimization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5189–5202, 2023. 2

[80] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European conference on computer vision*, pages 94–108. Springer, 2014. 1

[81] Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. *arXiv preprint arXiv:2202.03599*, 2022. 1

[82] Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8156–8165, 2021. 1, 3

[83] Zhanpeng Zhou, Mingze Wang, Yuchen Mao, Bingrui Li, and Junchi Yan. Sharpness-aware minimization efficiently selects flatter minima late in training. *arXiv preprint arXiv:2410.10373*, 2024. 11

[84] Shijie Zhu, Hui Zhao, Pengjie Wang, Hongbo Deng, Jian Xu, and Bo Zheng. Gradient deconfliction via orthogonal projections onto subspaces for multi-task learning. 2022. 8

[85] Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, sekhar tatikonda, James s Duncan, and Ting Liu. Surrogate gap minimization improves