# Fair Resource Allocation in Multi-Task Learning

**Hao Ban** [1] **Kaiyi Ji** [1]

## Abstract

By jointly learning multiple tasks, multi-task learning (MTL) can leverage the shared knowledge across tasks, resulting in improved data efficiency and generalization performance. However, a major challenge in MTL lies in the presence of conflicting gradients, which can hinder the fair optimization of some tasks and subsequently impede MTL's ability to achieve better overall performance. Inspired by fair resource allocation in communication networks, we formulate the optimization of MTL as a utility maximization problem, where the loss decreases across tasks are maximized under different fairness measurements. To address the problem, we propose FairGrad, a novel optimization objective. FairGrad not only enables flexible emphasis on certain tasks but also achieves a theoretical convergence guarantee. Extensive experiments demonstrate that our method can achieve state-of-the-art performance among gradient manipulation methods on a suite of multi-task benchmarks in supervised learning and reinforcement learning. Furthermore, we incorporate the idea of $\alpha$-fairness into the loss functions of various MTL methods. Extensive empirical studies demonstrate that their performance can be significantly enhanced. Code is available at `https://github.com/OptMN-Lab/fairgrad`.

## 1. Introduction

By aggregating labeled data for various tasks, multi-task learning (MTL) can not only capture the latent relationship across tasks but also reduce the computational overhead compared to training individual models for each task (Caruana, 1997; Evgeniou & Pontil, 2004; Thung & Wee, 2018). As a result, MTL has been successfully applied in various fields like natural language processing (Liu et al., 2016a; Zhang et al., 2023; Radford et al., 2019), computer vision (Zhang et al., 2014; Dai et al., 2016; Vandenhende et al., 2021), autonomous driving (Chen et al., 2018a; Ishihara et al., 2021; Yu et al., 2020a), and recommendation systems (Bansal et al., 2016; Li et al., 2020; Wang et al., 2020a). Research has shown that MTL is capable of learning robust representations, which in turn helps avoid overfitting certain individual tasks (Lounici et al., 2009; Zhang & Yang, 2021; Ruder, 2017; Liu et al., 2016b), and hence often achieves better generalization than the single-task counterparts.

MTL often solves the average loss across tasks in many real-world scenarios. However, it has been shown that there may exist conflicting gradients (Yu et al., 2020b; Liu et al., 2021; Wang et al., 2020b; Sener & Koltun, 2018) among tasks that exhibit different directions and magnitudes. If directly optimizing the average loss, the final update direction will often be dominated by the largest gradient, which can degrade the overall performance of MTL. To alleviate this negative impact, a series of gradient manipulation methods have been proposed to find a compromised direction (Désidéri, 2012; Chen et al., 2018b; Yu et al., 2020b; Liu et al., 2021; Navon et al., 2022; Xiao et al., 2023; Liu et al., 2023). In this paper, we view these methods from a novel fairness perspective. For example, MGDA (Désidéri, 2012) and its variants such as (Xiao et al., 2023; Fernando et al., 2022; Liu et al., 2021; 2023) tends to strike a max-min fairness among tasks, where the least-fortune tasks (i.e., with the lowest progress) are the most important. Nash-MTL (Navon et al., 2022) aims to achieve proportional fairness among tasks by formulating the problem as a bargaining game, attaining a balanced solution that is not dominated by any single large gradient.

However, different applications may favor different types of task fairness, and there is currently no unified framework in MTL that allows for the incorporation of diverse fairness concepts beyond those previously mentioned. To fill this gap, we propose a novel fair MTL framework, as well as efficient algorithms with performance guarantee. Our specific contributions are summarized below.

- We first draw an important connection between MTL and fair resource allocation in communication networks (Jain et al., 1984; Kelly, 1997; Mo & Walrand, 2000; Radunovic & Le Boudec, 2007; Srikant & Ying, 2013; Ju & Zhang, 2014; Liu & Xia, 2015), where we

[1]Department of Computer Science and Engineering, University at Buffalo, New York, United States. Correspondence to: Kaiyi Ji <kaiyiji@buffalo.edu>.

*Proceedings of the $41^{st}$ International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

think of the common search direction $d$ shared by all tasks as a resource to minimize their losses, and the service quality is measure by the loss decrease after performing a gradient descent along $d$. Inspired by this connection, we model MTL as a utility maximization problem, where each task is associated with $\alpha$-fair utility function and different $\alpha$ yields different ideas of fairness including max-min, proportional, minimum potential delay fairness, etc.

- We propose a novel algorithm named **FairGrad** to solve the $\alpha$-fair MTL utility maximization problem. FairGrad is easy to implement, allows for a flexible selection of $\alpha$, and guarantees convergence to a Parato stationary point under mild assumptions.
- Extensive experiments show that our FairGrad method can achieve state-of-the-art overall performance among gradient manipulation methods on 5 benchmarks in supervised learning and reinforcement learning with the number of tasks from 2 to 40.
- Finally, we incorporate our idea of $\alpha$-fairness into the loss functions of existing methods including Linear Scalarization, RLW, DWA, UW, MGDA, PCGrad, and CAGrad, and demonstrate that it can significantly improve their overall performance.

## 2. Related Work

**Multi-Task Learning.** MTL has drawn significant attention both in theory and practice. One class of studies is designing sophisticated model architectures. These studies can be mainly divided into two categories, hard parameter sharing where task-specific layers are built on a common feature space (Liu et al., 2019; Kokkinos, 2017), and soft parameter sharing which couples related parameters through certain constraints (Ruder et al., 2019; Gao et al., 2020). Another line of research aims to capture the relationship among tasks to guide knowledge transfer effectively (Zhao et al., 2020; Ciliberto et al., 2017). Additionally, the magnitudes of losses for different tasks may vary, posing challenges to the optimization of MTL. A group of studies seeks to balance tasks through heuristic re-weighting rules such as task-dependent uncertainty (Kendall et al., 2018), gradient magnitudes (Chen et al., 2018b), and the rate of change of loss for each task (Liu et al., 2019).

As MTL is one of the important applications of multi-objective optimization (MOO), several MOO-based gradient manipulation methods have been explored recently to address the challenge of conflicting gradients. (Désidéri, 2012) proposed MGDA, and show it guarantees the convergence to the Pareto front under certain assumptions. (Sener & Koltun, 2018) cast MTL as a MOO problem and refined MGDA for optimization in the context of deep neural networks. (Yu et al., 2020b) determined the update by projecting a task's gradient onto the normal plane of other conflicting gradients. (Liu et al., 2021) limited the update to a neighborhood of the average gradient. (Navon et al., 2022) considered finding the update as a bargaining game across all tasks. (Liu et al., 2023) searched for the update with the largest worst-case loss improvement rate to ensure that all tasks are optimized with approximately similar progress.

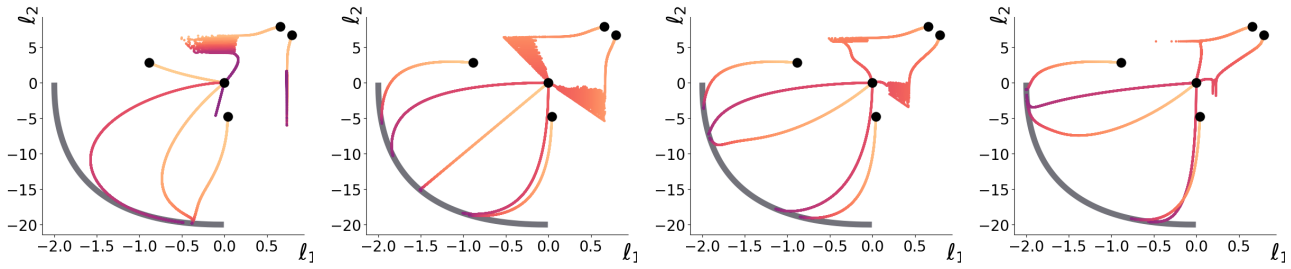
Theoretically, (Hu et al., 2023) showed that Linear Scalarization cannot fully explore the Pareto front compared with MGDA-variant methods. (Zhou et al., 2022) proposed a correlation-reduced stochastic gradient manipulation method to address the non-convergence issue of MGDA, CAGrad, and PCGrad in the stochastic setting. (Fernando et al., 2022) introduced a stochastic variant of MGDA with guaranteed convergence. (Xiao et al., 2023) proposed a simple and provable SGD-type method that benefits from direction-oriented improvements like (Liu et al., 2021). (Chen et al., 2023) offered a framework for analyzing stochastic MOO algorithms, considering the trade-off among optimization, generalization, and conflict-avoidance.

**Fairness in Resource Allocation.** Fair resource allocation has been studied for decades in wireless communication (Nandagopal et al., 2000; Eryilmaz & Srikant, 2006; Lan et al., 2010; Huaizhou et al., 2013; Noor-A-Rahim et al., 2020; Xu et al., 2021), where limited resources such as power and communication bandwidth need to be fairly allocated to users of the networks. Various fairness criteria have been proposed to improve the service quality for all users without sacrificing the overall network throughput. For example, Jain's fairness index (Jain et al., 1984) prefers all users to share the resources equally. Proportional fairness (Kelly, 1997) distributes resources proportional to user demands or priorities. Max-min fairness (Radunovic & Le Boudec, 2007) attempts to protect the user who receives the least amount of resources by providing them with the maximum possible allocation. The $\alpha$-fairness framework was proposed to unify multiple fairness criteria, where different choices of $\alpha$ lead to different ideas of fairness (Mo & Walrand, 2000; Lan et al., 2010). Recent research has explored the application of fair resource allocation in federated learning (Li et al., 2019; Zhang et al., 2022). In this paper, we connect MTL with fair resource allocation and further propose an $\alpha$-fair utility maximization problem as well as an efficient algorithm to solve it.

## 3. Preliminaries

### 3.1. Multiple Objectives and Pareto Concepts

MTL involves multiple objective functions, denoted as $L(\theta) = (l_1(\theta), \cdots, l_K(\theta))$, where $\theta$ represents model pa-



(a) FairGrad (without fairness) (b) FairGrad (proportionally fair) (c) FairGrad (MPD fair) (d) FairGrad (max-min fair)

*Figure 1.* An illustrative two-task example from (Navon et al., 2022) to show the convergence of FairGrad to Pareto front from different initialization points (black dots •). The optimization trajectories are colored from orange to purple. The bold gray line represents the Pareto front. The illustration showcases four fairness concepts (from left to right): simple average (i.e., Linear Scalarization (LS)), proportional fairness, minimum potential delay (MPD) fairness, and max-min fairness. It can be seen that LS is inclined towards the task 2 with a larger gradient. FairGrad with proportional fairness resembles Nash-MTL (Navon et al., 2022), and can find more balanced solutions along the Pareto front. MPD fairness aims to minimize the overall time for all tasks to converge, and shifts slightly more attention to some struggling tasks with smaller gradients. Max-min fairness emphasizes more on the less-fortune task with a smaller gradient magnitude. Also, observe that our FairGrad ensures the convergence to the Pareto front from all different initialization points.

rameters, $K$ is the number of tasks, and $l_i$ refers to the loss function of the $i$-th task.

Given two points $\theta_1, \theta_2 \in \mathbb{R}^m$, we say that $\theta_1$ *dominates* $\theta_2$ if $l_i(\theta_1) \leq l_i(\theta_2)$ for all $i \in [K]$ and $L(\theta_1) \neq L(\theta_2)$. A point is considered *Pareto optimal* if it is not dominated by any other point. This means that no improvement can be made in any one objective without negatively affecting at least one other objective. The set of all Pareto optimal points form into the *Pareto front*. A point $\theta \in \mathbb{R}^m$ is *Pareto stationary* if $\min_{w \in \mathcal{W}} \|G(\theta)w\| = 0$, where $G(\theta) = [g_1(\theta), \cdots, g_K(\theta)] \in \mathbb{R}^{m \times K}$ is the matrix with each column $g_i(\theta)$ denoting the gradient of $i$-th objective, and $\mathcal{W}$ is the probability simplex defined on $[K]$. Pareto stationarity is a necessary condition for Pareto optimality.

### 3.2. $\alpha$-Fair Resource Allocation

In the context of fair resource allocation in communication networks with $K$ users, the goal is to properly allocate resources (e.g., channel bandwidth, transmission rate) to maximize the total user utility (e.g., throughput) under the link capacity constraints. A generic overall objective for the network is given by

$$\max_{x_1,\ldots,x_K \in \mathcal{D}} \quad \sum_{i \in [K]} u(x_i) := \frac{x_i^{1-\alpha}}{1-\alpha}, \tag{1}$$

where $\frac{x_i^{1-\alpha}}{1-\alpha}$ is the $\alpha$-fair function with $\alpha \in [0,1) \cup (1, +\infty)$, and $x_i$ denotes the packet transmission rate, and $\mathcal{D}$ denotes the convex link capacity constraints. Different $\alpha$ implements different ideas of fairness, as elaborated below. Let $x_i^*$ be the solution of the utility maximization problem.

**Proportional Fairness**. This type of fairness is achieved when $\alpha \to 1$. To see this, the user utility function then becomes $\log x_i$, and it can be shown (see Appendix B.1) that $\sum_i \frac{x_i - x_i^\star}{x_i^\star} \leq 0$ for any $x_i \in \mathcal{D}$. This inequality indicates that if the amount of resource assigned to one user is increased, then the sum of the proportional changes of all other users is non-positive and hence there is at least one other user with a **negative** proportional change. Thus, $x_i^*, i = 1, ..., K$ are called proportionally fair.

**Minimum Potential Delay Fairness**. When $\alpha = 2$, the utility function is $-\frac{1}{x_i}$ and hence the overall objective is to minimize $\sum_i \frac{1}{x_i}$. Since $x_i$ is the transmission rate in networks, $\frac{1}{x_i}$ can be reviewed as the delay of transferring a file with unit size, and hence this case is called minimum potential delay fairness.

**Max-Min Fairness**. When $\alpha \to +\infty$, it is shown from Appendix B.1 that for any feasible allocation $x_i, i = 1, ..., K \in \mathcal{D}$, if $x_i > x_i^\star$ for some user $i$ then there exists another user $j$ such that $x_j^\star \leq x_i^\star$ and $x_j < x_j^\star$, which further indicates that $\min_i x_i^\star \geq \min_i x_i$. Thus, max-min fairness tends to protect the user who receives the least amount of resources by providing them with the maximum possible allocation.

## 4. FairGrad: Fair Resource Allocation in MTL

Inspired by the fair resource allocation over networks in Section 3.2, we now provide an $\alpha$-fair framework for MTL. Let $d$ be the updating direction for all tasks within the ball $B_\epsilon$ centered at 0 with a radius of $\epsilon$, and $g_i$ be the gradient for task $i$. Then, based on the first-order Taylor approximation

of $l_i(\theta)$, we have for a small stepsize $\eta$

$$\frac{1}{\eta}[l_i(\theta) - l_i(\theta - \eta d)] \approx g_i^\top d,$$

and hence $g_i^\top d$ can be regarded as the loss decreasing rate that plays a role similar to the transmission rate $x_i$ in communication networks. Towards this end, we define the $\alpha$-fair utility for each user $i$ as $\frac{(g_i^\top d)^{1-\alpha}}{1-\alpha}$, and hence the overall objective is to maximize the following total utilities of all tasks:

$$\begin{aligned} \max_{d \in B_\epsilon} & \sum_{i \in [K]} \frac{(g_i^\top d)^{1-\alpha}}{1-\alpha} \\ \text{s.t.} \quad & g_i^\top d \geq 0, \end{aligned} \tag{2}$$

where $\alpha \in [0, 1) \cup (1, +\infty)$. Note that our $\alpha$-fair framework takes the same spirit as Linear Scalarization (LS), Nash-MTL, and MGDA when we take $\alpha \to 0, 1, \infty$, respectively, and provides the coverage over other fairness ideas.

### 4.1. Analogy

**Utility**. For each user $i$ in a communication network, it usually holds that the larger the allocated transmission rate $x_i$, the higher the user's level of satisfaction. However, the total link capacity in a communication network is constrained. For each task $i$ in MTL scenarios, the larger the loss decreasing rate $g_i^\top d$, the more optimized the task becomes. As we consider the update direction $d$ within a ball $B_\epsilon$ centered at 0 with a radius of $\epsilon$, the feasible update progress for each task is also constrained. In a communication network, increasing the allocated transmission rate for one user may decrease the rate of other users. Similarly, conflicting gradients may occur in MTL.

**Capacity Constraint**. In the network resource allocation, the convex link capacity constraint refers to the constraints imposed on individual network links to ensure the packet transmission rate across each link does not exceed its capacity. It can be formulated as follows:

$$\sum_{i \in L} x_i < C,$$

where $x_i \geq 0$, $C$ represents the capacity of the network link, and $L$ denotes the users who transmit their packets on this link. In Equation (2), the loss decrease rate $g_i^\top d$ is modeled as a utility. The feasible update direction $d$ in the ball $B_\epsilon$. This is similar to the capacity constraint in network resource allocation because $d$ here cannot be arbitrarily large and then has capacity in MTL. In addition, the constraint on $d$ in our case is: $g_i^\top d \geq 0$ and $d \in B_\epsilon$ for all $i$, which turns out to be convex, as in network resource allocation where the capacity constraint is convex.

**Algorithm 1** FairGrad for MTL

**Input:** Model parameters $\theta_0$, $\alpha$, learning rate $\{\eta_t\}$
**for** $t = 1$ **to** $T - 1$ **do**
  Compute gradients $G(\theta_t) = [g_1(\theta_t), \cdots, g_K(\theta_t)]$
  Solve Equation (4) to obtain $w_t$
  Compute $d_t = G(\theta_t) w_t$
  Update the parameters $\theta_{t+1} = \theta_t - \eta d_t$
**end for**

### 4.2. Method

We next take the following steps to solve the problem in Equation (2). First note that the objective function is non-decreasing with respect to any feasible $d$. Thus, if $d$ lies in the interior of $B_\epsilon$, then there must exist a point along the same direction but on the boundary of $B_\epsilon$, which achieves a larger overall utility. Thus, it can be concluded that the optimal $d^*$ lies on the boundary, and the gradient of the overall objective is aligned with $d^*$, i.e.,

$$\sum_i g_i (g_i^\top d)^{-\alpha} = cd \tag{3}$$

for some constant $c > 0$. Following Nash-MTL (Navon et al., 2022), we take $c = 1$ for simplicity, and assume that the gradients of tasks are linearly independent when not at a Pareto stationary point $\theta$ such that $d$ can be represented as a linear combination of task gradients: $d = \sum_i w_i g_i$, where $w := (w_1, ..., w_K)^\top \in \mathbb{R}_+^K$ denotes the weights. Then, we obtain from Equation (3) that $(g_i^\top d)^{-\alpha} = w_i$, which combined with $d = \sum_i w_i g_i$, implies that

$$G^\top G w = w^{-1/\alpha}, \tag{4}$$

where $\alpha \neq 0$ and the power $-1/\alpha$ is applied elementwisely. It is evident that we have $w_i = 1$ for all $i \in [K]$ when $\alpha = 0$. Differently from Nash-MTL that approximates the solution using a sequence of convex optimization problems, we treat Equation (4) as a simple constrained nonlinear least square problem

$$\begin{aligned} \min_w & \sum_i f(w)_i^2 \\ \text{s.t.} \quad & f(w) = G^\top G w - w^{-1/\alpha} \quad w \in \mathbb{R}_+^K, \end{aligned}$$

which is solved by *scipy.optimize.least_squares* efficiently. The complete procedure of our algorithm is summarized in Algorithm 1.

## 5. Empirical Results

We first use a toy example to elaborate how FairGrad balances the tasks by incorporating different fairness criteria. Then we conduct extensive experiments under both supervised learning and reinforcement learning settings to demon-

*Table 1.* Results on CelebA (40-task) and QM9 (11-task) datasets. Each experiment is repeated 3 times with different random seeds and the average is reported.

| METHOD | CELEBA | | QM9 | |
|---|---|---|---|---|
| | MR ↓ | $\Delta m\%$ ↓ | MR ↓ | $\Delta m\%$ ↓ |
| LS | 6.53 | 4.15 | 8.18 | 177.6 |
| SI | 8.00 | 7.20 | 4.82 | 77.8 |
| RLW | 5.40 | 1.46 | 9.55 | 203.8 |
| DWA | 7.23 | 3.20 | 7.82 | 175.3 |
| UW | 6.00 | 3.23 | 6.18 | 108.0 |
| MGDA | 11.05 | 14.85 | 7.73 | 120.5 |
| PCGRAD | 6.98 | 3.17 | 6.36 | 125.7 |
| CAGRAD | 6.53 | 2.48 | 7.18 | 112.8 |
| IMTL-G | **4.95** | 0.84 | 6.09 | 77.2 |
| NASH-MTL | 5.38 | 2.84 | **3.64** | 62.0 |
| FAMO | 5.03 | 1.21 | 4.73 | 58.5 |
| FAIRGRAD | **4.95** | **0.37** | 3.82 | **57.9** |

strate the effectiveness of our proposed method. Full experimental details and more empirical studies can be found in Appendix A.

### 5.1. Toy Example

We adopt the 2-task toy example introduced in (Navon et al., 2022), where the objectives of the 2 tasks, denoted as $L_1$ and $L_2$, have different scales. More details are provided in Appendix A.1. We select 5 starting points and illustrate the optimization trajectories of FairGrad with different fairness criteria in Figure 1.

Obviously, without fairness (Linear Scalarization), the algorithm may not converge to a Pareto stationary point. However, in other cases where fairness is involved, the algorithm can converge. Furthermore, in the experiment setting, objective $L_2$ exhibits a larger scale than objective $L_1$, resulting in a larger gradient magnitude. If there is no fairness, task 2 will dominate the optimization process. When the algorithm converges, it will always converge to a stationary point where $L_2$ is smaller than $L_1$, as shown in Figure 1. On the other hand, with max-min fairness, the least fortunate task will be prioritized. The algorithm tends to converge to a stationary point with a smaller $L_1$. Proportional fairness and MPD fairness will lead to a more balanced solution.

### 5.2. Supervised Learning

We evaluate the performance of our method in three different supervised learning scenarios described as follows.

**Image-Level Classification.** CelebA (Liu et al., 2015) is a large-scale face attributes dataset, containing over 200K celebrity images. Each image is annotated with 40 attributes, such as smiling, wavy hair, mustache, etc. We can consider the dataset as an image-level 40-task MTL classification problem, with each task predicting the presence of a specific attribute. This setting assesses the capability of MTL methods in handling a large number of tasks. We follow the experiment setup in (Liu et al., 2023). We employ a network containing a 9-layer convolutional neural network (CNN) as the backbone and a specific linear layer for each task. We train our method for 15 epochs, using Adam optimizer with learning rate 3e-4. The batch size is 256.

**Regression.** QM9 (Ramakrishnan et al., 2014) is a widely-used benchmark in graph neural networks. It comprises over 130k organic molecules, which are organized as graphs with annotated node and edge features. The goal of predicting 11 properties with different measurement scales is to see if MTL methods can effectively balance the variations present across these tasks. Following (Navon et al., 2022; Liu et al., 2023), we use the example provided in Pytorch Geometric (Fey & Lenssen, 2019), and use 110k molecules for training, 10k for validation, and the rest 10k for testing. We train our method for 300 epochs with a batch size of 120. The initial learning rate is 1e-3, and a scheduler is used to reduce the learning rate once the improvement of validation stagnates.

**Dense Prediction.** NYU-v2 (Silberman et al., 2012) contains 1449 densely annotated images that have been collected from video sequences of various indoor scenes. It involves one pixel-level classification task and two pixel-level regression tasks, which correspond to 13-class semantic segmentation, depth estimation, and surface normal prediction, respectively. Similarly, Cityscapes (Cordts et al., 2016) contains 5000 street-scene images with two tasks: 7-class semantic segmentation and depth estimation. This scenario evaluates the effectiveness of MTL methods in tackling complex situations. We follow (Liu et al., 2021; Navon et al., 2022; Liu et al., 2023) and adopt the backbone of MTAN (Liu et al., 2019), which adds task-specific attention modules on SegNet (Badrinarayanan et al., 2017). We train our method for 200 epochs with batch size 2 for NYU-v2 and 8 for Cityscapes. The learning rate is 1e-4 for the first 100 epochs, then decayed by half for the rest.

**Evaluation.** For image-level classification and regression, we compare our FairGrad with Linear Scalarization (LS) which minimizes the sum of task losses, Scale-Invariant (SI) which minimizes the sum of logarithmic losses, Random Loss Weighting (RLW) (Lin et al., 2021), Dynamic Weigh Average (DWA) (Liu et al., 2019), Uncertainty weighting (UW) (Kendall et al., 2018), MGDA (Sener & Koltun, 2018), PCGrad (Yu et al., 2020b), CAGrad (Liu et al., 2021), IMTL-G (Liu et al., 2020), Nash-MTL (Navon et al., 2022), and FAMO (Liu et al., 2023). For dense prediction, we also compare with GradDrop (Chen et al., 2020), and MoCo (Fernando et al., 2022). We consider two metrics to represent the overall performance of the MTL method $m$. **(1)**

*Table 2.* Results on NYU-v2 (3-task) dataset. Each experiment is repeated 3 times with different random seeds and the average is reported.

| METHOD | SEGMENTATION | | DEPTH | | SURFACE NORMAL | | | | | MR ↓ | $\Delta m\%$ ↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | ANGLE DISTANCE ↓ | | WITHIN $t^\circ$ ↑ | | | | |
| | 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 | | |
| LS | 39.29 | 65.33 | 0.5493 | 0.2263 | 28.15 | 23.96 | 22.09 | 47.50 | 61.08 | 10.67 | 5.59 |
| SI | 38.45 | 64.27 | 0.5354 | 0.2201 | 27.60 | 23.37 | 22.53 | 48.57 | 62.32 | 9.44 | 4.39 |
| RLW | 37.17 | 63.77 | 0.5759 | 0.2410 | 28.27 | 24.18 | 22.26 | 47.05 | 60.62 | 13.11 | 7.78 |
| DWA | 39.11 | 65.31 | 0.5510 | 0.2285 | 27.61 | 23.18 | 24.17 | 50.18 | 62.39 | 9.44 | 3.57 |
| UW | 36.87 | 63.17 | 0.5446 | 0.2260 | 27.04 | 22.61 | 23.54 | 49.05 | 63.65 | 9.22 | 4.05 |
| MGDA | 30.47 | 59.90 | 0.6070 | 0.2555 | 24.88 | **19.45** | 29.18 | **56.88** | **69.36** | 7.11 | 1.38 |
| PCGRAD | 38.06 | 64.64 | 0.5550 | 0.2325 | 27.41 | 22.80 | 23.86 | 49.83 | 63.14 | 9.78 | 3.97 |
| GRADDROP | 39.39 | 65.12 | 0.5455 | 0.2279 | 27.48 | 22.96 | 23.38 | 49.44 | 62.87 | 8.78 | 3.58 |
| CAGRAD | 39.79 | 65.49 | 0.5486 | 0.2250 | 26.31 | 21.58 | 25.61 | 52.36 | 65.58 | 5.78 | 0.20 |
| IMTL-G | 39.35 | 65.60 | 0.5426 | 0.2256 | 26.02 | 21.19 | 26.20 | 53.13 | 66.24 | 5.11 | -0.76 |
| MOCO | **40.30** | **66.07** | 0.5575 | **0.2135** | 26.67 | 21.83 | 25.61 | 51.78 | 64.85 | 5.44 | 0.16 |
| NASH-MTL | 40.13 | 65.93 | **0.5261** | 0.2171 | 25.26 | 20.08 | 28.40 | 55.47 | 68.15 | 3.11 | -4.04 |
| FAMO | 38.88 | 64.90 | 0.5474 | 0.2194 | 25.06 | 19.57 | 29.21 | 56.61 | 68.98 | 4.44 | -4.10 |
| FAIRGRAD | 39.74 | 66.01 | 0.5377 | 0.2236 | **24.84** | 19.60 | **29.26** | 56.58 | 69.16 | **2.67** | **-4.66** |

$\mathbf{\Delta m}\%$, the average per-task performance drop against the single-task (STL) baseline $b$:

$$\Delta m\% = \frac{1}{K}\sum_{i=1}^{K}(-1)^{\delta_k}(M_{m,k} - M_{b,k})/M_{b,k} \times 100,$$

where $M_{b,k}$ denotes the value of metric $M_k$ from baseline $b$, $M_{m,k}$ denotes the value of metric $M_k$ from the compared method $m$, and $\delta_k = 1$ if metric $M_k$ prefers a higher value. **(2) Mean Rank (MR)**, the average rank of each metric across tasks.

**Results.** The experiment results are shown in Table 1, Table 2, and Table 3. Each experiment is repeated 3 times with different random seeds and the average is computed. It can be seen from Table 1 that the proposed FairGrad outperforms existing methods on the CelebA dataset with 40 tasks, indicating that it performs effectively when faced with a substantial number of tasks. Table 1 shows that FairGrad also achieves the best overall performance drop $\Delta m\%$ on the QM9 dataset, while attaining a mean rank of 3.82 comparable to the best 3.64 of Nash-MTL. In addition, Table 2 and Table 3 show that FairGrad outperforms all the baselines on the NYU-v2 and Cityscapes datasets w.r.t. MR and $\Delta m\%$, demonstrating its effectiveness in learning from scene understanding scenarios.

Furthermore, there are some other interesting findings from the results presented in Table 2. LS performs poorly in the surface normal prediction (SNP) task compared to the other two tasks. This is because LS does not take the fairness among tasks into consideration, and hence the gradient of the SNP task is dominated by the others. On the contrary, MGDA (Sener & Koltun, 2018) obtains the best performance in the SNP task among all three tasks by enforcing the max-min fairness. Meanwhile, the performance of Nash-MTL (Navon et al., 2022), which embodies proportional fairness, is more balanced across all tasks. As a comparison, our FairGrad can find a more balanced solution than LS and MGDA, while placing greater emphasis on the challenging SNP tasks than Nash-MTL.

### 5.3. Reinforcement Learning

We further evaluate our method on the MT10, a benchmark including 10 robotic manipulation tasks from the MetaWorld environment (Yu et al., 2020c), where the objective is to learn one policy that generalizes to different tasks such as pick and place, open door, etc. We follow (Liu et al., 2021; Navon et al., 2022; Liu et al., 2023) and adopt Soft Actor-Critic (SAC) (Haarnoja et al., 2018) as the underlying algorithm. We implement with MTRL codebase (Shagun Sodhani, 2021) and train our method for 2 million steps with a batch size of 1280.

**Evaluation.** We compare our FairGrad with Multi-task SAC (MTL SAC) (Yu et al., 2020c), Multi-task SAC with task encoder (MTL SAC + TE) (Yu et al., 2020c), Multi-headed SAC (MH SAC) (Yu et al., 2020c), PCGrad (Yu et al., 2020b), CAGrad (Liu et al., 2021), MoCo (Fernando et al., 2022), Nash-MTL (Navon et al., 2022), and FAMO (Liu et al., 2023).

**Results.** The results are shown in Table 4. Each method is evaluated once every 10,000 steps, and the best average success rate over 10 random seeds throughout the entire training course is reported. We could not reproduce the MTRL result in the original paper of Nash-MTL exactly,

*Table 3.* Results on Cityscapes (2-task) dataset. Each experiment is repeated 3 times with different random seeds and the average is reported.

| METHOD | SEGMENTATION | | DEPTH | | MR ↓ | $\Delta m\%$ ↓ |
|---|---|---|---|---|---|---|
| | MIOU ↑ | PIX ACC ↑ | ABS ERR ↓ | REL ERR ↓ | | |
| STL | 74.01 | 93.16 | 0.0125 | 27.77 | | |
| LS | 75.18 | 93.49 | 0.0155 | 46.77 | 8.50 | 22.60 |
| SI | 70.95 | 91.73 | 0.0161 | 33.83 | 10.50 | 14.11 |
| RLW | 74.57 | 93.41 | 0.0158 | 47.79 | 10.75 | 24.38 |
| DWA | 75.24 | 93.52 | 0.0160 | 44.37 | 8.50 | 21.45 |
| UW | 72.02 | 92.85 | 0.0140 | **30.13** | 6.75 | 5.89 |
| MGDA | 68.84 | 91.54 | 0.0309 | 33.50 | 11.00 | 44.14 |
| PCGRAD | 75.13 | 93.48 | 0.0154 | 42.07 | 8.50 | 18.29 |
| GRADDROP | 75.27 | 93.53 | 0.0157 | 47.54 | 8.00 | 23.73 |
| CAGRAD | 75.16 | 93.48 | 0.0141 | 37.60 | 7.00 | 11.64 |
| IMTL-G | 75.33 | 93.49 | 0.0135 | 38.41 | 5.50 | 11.10 |
| MOCO | 75.42 | 93.55 | 0.0149 | 34.19 | 4.50 | 9.90 |
| NASH-MTL | 75.41 | 93.66 | **0.0129** | 35.02 | 3.25 | 6.82 |
| FAMO | 74.54 | 93.29 | 0.0145 | 32.59 | 7.25 | 8.13 |
| FAIRGRAD | **75.72** | **93.68** | 0.0134 | 32.25 | **1.50** | **5.18** |

*Table 4.* Results on MT10 benchmark. Average over 10 random seeds. Nash-MTL$^\star$ denotes the result reported in the original paper (Navon et al., 2022). While Nash-MTL (reproduced) denotes the reproduced result in (Liu et al., 2023).

| METHOD | SUCCESS RATE (MEAN ± STDERR) |
|---|---|
| STL | $0.90 \pm 0.03$ |
| MTL SAC | $0.49 \pm 0.07$ |
| MTL SAC + TE | $0.54 \pm 0.05$ |
| MH SAC | $0.61 \pm 0.04$ |
| PCGRAD | $0.72 \pm 0.02$ |
| CAGRAD | $0.83 \pm 0.05$ |
| MOCO | $0.75 \pm 0.05$ |
| NASH-MTL$^\star$ | $\mathbf{0.91} \pm 0.03$ |
| NASH-MTL (REPRODUCED) | $0.80 \pm 0.13$ |
| FAMO | $0.83 \pm 0.05$ |
| FAIRGRAD | $0.84 \pm 0.07$ |

and hence we adopt the reproduced result of Nash-MTL in (Liu et al., 2023). It is evident that our method performs competitively when compared to other methods.

### 5.4. Effect of Different Fairness Criteria

We investigate the effect of different fairness criteria on NYU-v2 and Cityscapes datasets by setting $\alpha \to [1, 2, 5, 10]$, which corresponds to the proportional fairness, minimum potential delay fairness, and approximate max-min fairness. The results are presented in Table 5. The results show that different fairness criteria prioritize different tasks, and thus lead to different overall performance. In particular, the minimum potential delay fairness with $\alpha = 2$ achieves the best $\Delta m\%$ among all fairness criteria.

Also note that although the best $\Delta m\%$ reported in Table 5 is better than that reported in Table 2 and Table 3, their results w.r.t. MR are worse than those in Table 2 and Table 3. This is because an improved $\Delta m\%$ may result in a lower rank for certain tasks, causing a significant degradation in the average rank. See Appendix A.3 for more details.

*Table 5.* $\Delta m\%$ of different fairness criteria on NYU-v2 (3-task) and Cityscapes (2-task) datasets.

| METHOD | NYU-V2 | CITYSCAPES |
|---|---|---|
| FAIRGRAD ($\alpha = 1$) | -2.79 | 6.73 |
| FAIRGRAD ($\alpha = 2$) | **-4.96** | **3.90** |
| FAIRGRAD ($\alpha = 5$) | -3.03 | 6.87 |
| FAIRGRAD ($\alpha = 10$) | -1.00 | 10.54 |

### 5.5. Discussion on Practical Implementation

**Supervised Learning.** For experiments on QM9, NYU-v2, and Cityscapes, we implement our method based on the codes released by (Navon et al., 2022). For experiments on CelebA, our implementation is based on the codes provided by (Liu et al., 2023), consistent with all the baselines presented in Table 1.

**Reinforcement Learning.** We find it time-consuming to solve the constrained nonlinear least square problem discussed in Section 4 under the reinforcement learning setting. Therefore, we use SGD to approximately solve the problem and accelerate the training process. Specifically, we use SGD optimizer with a learning rate of 0.1, momentum of 0.5, and train 20 epochs.

**Choice of $\alpha$.** We first search with $\alpha \in [1, 2, 5, 10]$ and evaluate which choice is better. Then we narrow down the search space and continue to execute a grid search with a step size of 0.1 until we determine an appropriate value.

In practice, the choice depends on the specific needs or preferences. If there are no requirements for fairness and tasks with larger gradients are allowed to finish first, we can simply set $\alpha = 0$ to allow for quick training. If tasks with struggling progress are prioritized (e.g., in some meta-learning setups, some harder-to-train tasks may play more important roles in deciding final test accuracy), then the max-min fairness (with a larger $\alpha$) is desired. From our observation, if aiming to achieve the most balanced overall performance, MPD fairness with $\alpha = 2$ is preferable. After fairness criteria are selected, some slight finetuning on $\alpha$ can also be conducted to further improve the overall accuracy.

## 6. Applying $\alpha$-Fairness to Existing Methods

In MTL, tasks often exhibit variations in difficulty, resulting in losses that may vary in scale. Since the idea of $\alpha$-fairness provides a framework unifying different fairness criteria, we argue that it can be directly applied in many MTL methods to mitigate the problem of varying loss scales by replacing the task losses $(l_1, \cdots, l_K)$ with

$$\Big(\frac{l_1^{1-\alpha}}{1-\alpha}, \cdots, \frac{l_K^{1-\alpha}}{1-\alpha}\Big), \tag{5}$$

where $\alpha \in (-\infty, 1)$ and $i \in [K]$. Note that the meaning of $\alpha$ here differs from that used in Section 4. FairGrad aims to address the issue of varying loss decreasing rates, while applying $\alpha$-fairness to existing methods tries to deal with the issue of varying loss scales. Although the ideas of $\alpha$-fairness are the same, the goals are different.

Here we omit the model parameter $\theta$ for simplicity. It can be observed that the gradient changes from $g_i$ to $g_i/l_i^\alpha$, where $\alpha$ controls the emphasis placed on tasks with different levels of difficulty. Take the example of simply summing $\alpha$-fair losses of all tasks

$$\min \sum_{i\in[K]} \frac{l_i^{1-\alpha}}{1-\alpha}.$$

If we choose $\alpha = 0$, the objective is reduced to Linear Scalarization (LS) which minimizes the sum of all losses. If $\alpha \to 1$, the objective tends to minimize the sum of the logarithmic losses, which shares similarity with Scale-Invariant (SI). If $\alpha \to -\infty$, the objective exhibits the notion of the minimax fairness (Radunovic & Le Boudec, 2007), which aims to minimize the maximum loss among all tasks.

**Proposition 6.1.** *The Pareto front of the $\alpha$-fair loss functions in Equation (5) is the same as that of original loss functions $(l_1, ..., l_K)$.*

According to Proposition 6.1, transforming each $l_i$ to its $\alpha$-fair counterpart does not change the Pareto front, and allows us to find an improved solution along this front under a proper selection of fairness.

We then apply this $\alpha$-fair loss transformation to a series of MTL methods including LS, RLW (Lin et al., 2021), DWA (Liu et al., 2019), UW (Kendall et al., 2018), MGDA (Sener & Koltun, 2018), PCGrad (Yu et al., 2020b), CAGrad (Liu et al., 2021), and test the performance on NYU-v2 and Cityscapes datasets. We simply choose $\alpha = 0.5$ for all the experiments. Other experiment settings remain the same with Section 5.2. The results presented in Table 6 and Table 7 clearly demonstrate that the $\alpha$-fair loss transformation improves the performance of these MTL methods via a large margin.

Additionally, we also test the applicability of $\alpha$-fair loss transformation to FairGrad. It can be seen from Table 7 that compared to other MTL methods, applying this transformation to FairGrad provides only a marginal improvement. This shows that FairGrad can mitigate the issue of varying loss scales by incorporating fairness-based utility functions.

## 7. Theoretical Analysis

In this section, we provide a theoretical analysis of our method on the convergence to a Pareto stationary point, at which some convex combination of task gradients is 0. As mentioned before, we assume that the gradients of different tasks are linearly independent when not reaching a Pareto stationary point. Formally, we make the following assumption, as also adopted by (Navon et al., 2022).

**Assumption 7.1.** For the output sequence $\{\theta_t\}$ generated by the proposed method, the gradients of all tasks are linearly independent while not at a Pareto stationary point.

The following assumption imposes differentiability and Lipschitz continuity on the loss functions, as also adopted by (Liu et al., 2021; Navon et al., 2022).

**Assumption 7.2.** For each task, the loss function $l_i(\theta)$ is differentiable and $L$-smooth such that $\|\nabla l_i(\theta_1) - \nabla l_i(\theta_2)\| \leq L\|\theta_1 - \theta_2\|$ for any two points $\theta_1, \theta_2$.

Then, we obtain the following convergence theorem.

**Theorem 7.3.** *Suppose Assumptions 7.1-7.2 are satisfied. Set the stepsize $\eta_t = \frac{\sum_i w_{t,i}^{-1/\alpha}}{LK\sum_i w_{t,i}^{1-1/\alpha}}$, Then, there exists a subsequence $\{\theta_{t_j}\}$ of the output sequence $\{\theta_t\}$ that converges to a Pareto stationary point $\theta^\star$.*

*Proof skecth.* We first show that the average loss $\mathcal{L}(\theta_t) = \frac{1}{K}\sum_i l_i(\theta_t)$ is monotonically decreasing. Then, we show that the smallest singular value of the Gram matrix, denoted as $\sigma_K(G(\theta_t)^\top G(\theta_t))$, is upper bounded and approaches 0

*Table 6.* Results of $\alpha$-fair loss transformation on NYU-v2 (3-task) dataset. Each experiment is repeated 3 times with different random seeds and the average is reported. We simply choose $\alpha = 0.5$.

| Method | Segmentation | | Depth | | Surface Normal | | | | | $\Delta m\% \downarrow$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Angle Distance ↓ | | Within $t^\circ$ ↑ | | | |
| | mIoU ↑ | Pix Acc ↑ | Abs Err ↓ | Rel Err ↓ | Mean | Median | 11.25 | 22.5 | 30 | |
| LS | 39.29 | 65.33 | 0.5493 | 0.2263 | 28.15 | 23.96 | 22.09 | 47.50 | 61.08 | 5.59 |
| Fair-LS | 38.64 | 64.96 | 0.5422 | 0.2255 | 27.14 | 22.64 | 24.05 | 50.14 | 63.53 | **2.85** |
| RLW | 37.17 | 63.77 | 0.5759 | 0.2410 | 28.27 | 24.18 | 22.26 | 47.05 | 60.62 | 7.78 |
| Fair-RLW | 37.29 | 63.58 | 0.5481 | 0.2263 | 27.67 | 23.33 | 23.38 | 48.72 | 62.12 | **5.00** |
| DWA | 39.11 | 65.31 | 0.5510 | 0.2285 | 27.61 | 23.18 | 24.17 | 50.18 | 62.39 | 3.57 |
| Fair-DWA | 39.03 | 65.18 | 0.5404 | 0.2266 | 27.20 | 22.63 | 24.31 | 50.14 | 63.45 | **2.65** |
| UW | 36.87 | 63.17 | 0.5446 | 0.2260 | 27.04 | 22.61 | 23.54 | 49.05 | 63.65 | 4.05 |
| Fair-UW | 38.51 | 64.56 | 0.5423 | 0.2274 | 27.23 | 22.92 | 23.62 | 49.52 | 63.23 | **3.56** |
| MGDA | 30.47 | 59.90 | 0.6070 | 0.2555 | 24.88 | 19.45 | 29.18 | 56.88 | 69.36 | 1.38 |
| Fair-MGDA | 35.91 | 63.19 | 0.5646 | 0.2260 | 24.75 | 19.24 | 30.04 | 57.30 | 69.55 | **-3.26** |
| PCGrad | 38.06 | 64.64 | 0.5550 | 0.2325 | 27.41 | 22.80 | 23.86 | 49.83 | 63.14 | 3.97 |
| Fair-PCGrad | 39.26 | 65.08 | 0.5257 | 0.2177 | 26.88 | 22.26 | 24.74 | 50.85 | 64.18 | **1.23** |
| CAGrad | 39.79 | 65.49 | 0.5486 | 0.2250 | 26.31 | 21.58 | 25.61 | 52.36 | 65.58 | 0.20 |
| Fair-CAGrad | 39.32 | 65.36 | 0.5290 | 0.2221 | 25.50 | 20.32 | 28.06 | 54.94 | 67.65 | **-2.91** |

*Table 7.* Results of $\alpha$-fair loss transformation on Cityscapes (2-task) dataset. We simply choose $\alpha = 0.5$.

| Method | Segmentation | | Depth | | $\Delta m\% \downarrow$ |
|---|---|---|---|---|---|
| | mIoU ↑ | Pix Acc ↑ | Abs Err ↓ | Rel Err ↓ | |
| LS | 75.18 | 93.49 | 0.0155 | 46.77 | 22.60 |
| Fair-LS | 74.91 | 93.48 | 0.0137 | 37.51 | **10.86** |
| RLW | 74.57 | 93.41 | 0.0158 | 47.79 | 24.38 |
| Fair-RLW | 74.32 | 93.36 | 0.0140 | 37.46 | **11.64** |
| DWA | 75.24 | 93.52 | 0.0160 | 44.37 | 21.45 |
| Fair-DWA | 75.06 | 93.46 | 0.0147 | 35.34 | **10.74** |
| MGDA | 68.84 | 91.54 | 0.0309 | 33.50 | 44.14 |
| Fair-MGDA | 74.45 | 93.50 | 0.0131 | 37.64 | **9.91** |
| PCGrad | 75.13 | 93.48 | 0.0154 | 42.07 | 18.29 |
| Fair-PCGrad | 75.25 | 93.51 | 0.0140 | 37.00 | **10.71** |
| CAGrad | 75.16 | 93.48 | 0.0141 | 37.60 | 11.64 |
| Fair-CAGrad | 74.74 | 93.39 | 0.0134 | 33.04 | **6.23** |
| FairGrad | 75.72 | 93.68 | 0.0134 | 32.25 | 5.18 |
| Fair-FairGrad | 75.50 | 93.51 | 0.0131 | 32.54 | **4.92** |

as the number of training steps increases. Consequently, the output sequence $\{\theta_t\}$ has a subsequence converging to a point $\theta^\star$, where the matrix $G(\theta^\star)^\top G(\theta^\star)$ has a zero singular value and hence the gradients of all tasks are linearly dependent. This immediately indicates the attainment of a Pareto stationary point. □

## 8. Conclusion

We first discuss the connection between MTL and fair resource allocation in communication networks and model the optimization of MTL as a utility maximization problem by leveraging the concept of $\alpha$-fairness. Then, we introduce FairGrad, a novel MTL method offering the flexibility to balance different tasks through different selections of $\alpha$, and provide it with a theoretical convergence analysis. Our extensive experiments demonstrate not only the promising performance of FairGrad, but also the power of the $\alpha$-fairness idea in enhancing existing MTL methods.

For future studies, we will explore the performance of FairGrad in more challenging MTL settings with significantly diverse tasks. Theoretically, we will study the impact of varying levels of difficulty across tasks on the final convergence and generalization performance.

## Impact Statement

This paper discusses the fairness in optimization methods for multi-task learning (MTL). There are some potential societal consequences, none of which we feel must be specifically highlighted here.

## References

Badrinarayanan, V., Kendall, A., and Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

Bansal, T., Belanger, D., and McCallum, A. Ask the gru: Multi-task learning for deep text recommendations. In *proceedings of the 10th ACM Conference on Recom-*

*mender Systems*, pp. 107–114, 2016.

Caruana, R. Multitask learning. *Machine learning*, 28: 41–75, 1997.

Chen, L., Fernando, H., Ying, Y., and Chen, T. Three-way trade-off in multi-objective learning: Optimization, generalization and conflict-avoidance. *arXiv preprint arXiv:2305.20057*, 2023.

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

Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pp. 794–803. PMLR, 2018b.

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

Ciliberto, C., Rudi, A., Rosasco, L., and Pontil, M. Consistent multitask learning with nonlinear output relations. *Advances in Neural Information Processing Systems*, 30, 2017.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Dai, J., He, K., and Sun, J. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3150–3158, 2016.

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

Eryilmaz, A. and Srikant, R. Joint congestion control, routing, and mac for stability and fairness in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1514–1524, 2006.

Evgeniou, T. and Pontil, M. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 109–117, 2004.

Fernando, H. D., Shen, H., Liu, M., Chaudhury, S., Murugesan, K., and Chen, T. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *The Eleventh International Conference on Learning Representations*, 2022.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Gao, Y., Bai, H., Jie, Z., Ma, J., Jia, K., and Liu, W. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11543–11552, 2020.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870. PMLR, 2018.

Hu, Y., Xian, R., Wu, Q., Fan, Q., Yin, L., and Zhao, H. Revisiting scalarization in multi-task learning: A theoretical perspective. *arXiv preprint arXiv:2308.13985*, 2023.

Huaizhou, S., Prasad, R. V., Onur, E., and Niemegeers, I. Fairness in wireless networks: Issues, measures and challenges. *IEEE Communications Surveys & Tutorials*, 16(1):5–24, 2013.

Ishihara, K., Kanervisto, A., Miura, J., and Hautamaki, V. Multi-task learning with attention for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2902–2911, 2021.

Jain, R. K., Chiu, D.-M. W., Hawe, W. R., et al. A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 21, 1984.

Ju, H. and Zhang, R. Optimal resource allocation in full-duplex wireless-powered communication network. *IEEE Transactions on Communications*, 62(10):3528–3540, 2014.

Kelly, F. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):33–37, 1997.

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

Kokkinos, I. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6129–6138, 2017.

Lan, T., Kao, D., Chiang, M., and Sabharwal, A. *An axiomatic theory of fairness in network resource allocation*. IEEE, 2010.

Li, P., Li, R., Da, Q., Zeng, A.-X., and Zhang, L. Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2605–2612, 2020.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2019.

Lin, B., Ye, F., Zhang, Y., and Tsang, I. W. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *arXiv preprint arXiv:2111.10603*, 2021.

Liu, B., Liu, X., Jin, X., Stone, P., and Liu, Q. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.

Liu, B., Feng, Y., Stone, P., and Liu, Q. Famo: Fast adaptive multitask optimization. *arXiv preprint arXiv:2306.03792*, 2023.

Liu, H. and Xia, Y. Optimal resource allocation in complex communication networks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(7):706–710, 2015.

Liu, L., Li, Y., Kuang, Z., Xue, J.-H., Chen, Y., Yang, W., Liao, Q., and Zhang, W. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2020.

Liu, P., Qiu, X., and Huang, X. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016a.

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

Liu, T., Tao, D., Song, M., and Maybank, S. J. Algorithm-dependent generalization bounds for multi-task learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):227–241, 2016b.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Lounici, K., Pontil, M., Tsybakov, A. B., and Van De Geer, S. Taking advantage of sparsity in multi-task learning. *arXiv preprint arXiv:0903.1468*, 2009.

Mo, J. and Walrand, J. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, 2000.

Nandagopal, T., Kim, T.-E., Gao, X., and Bharghavan, V. Achieving mac layer fairness in wireless packet networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 87–98, 2000.

Navon, A., Shamsian, A., Achituve, I., Maron, H., Kawaguchi, K., Chechik, G., and Fetaya, E. Multi-task learning as a bargaining game. In *International Conference on Machine Learning*, pp. 16428–16446. PMLR, 2022.

Noor-A-Rahim, M., Liu, Z., Lee, H., Ali, G. M. N., Pesch, D., and Xiao, P. A survey on resource allocation in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):701–721, 2020.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Radunovic, B. and Le Boudec, J.-Y. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transactions on Networking*, 15(5): 1073–1083, 2007.

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Ruder, S., Bingel, J., Augenstein, I., and Søgaard, A. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4822–4829, 2019.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pp. 746–760. Springer, 2012.

Srikant, R. and Ying, L. *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

Thung, K.-H. and Wee, C.-Y. A brief review on multi-task learning. *Multimedia Tools and Applications*, 77:29705–29725, 2018.

Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., and Van Gool, L. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (7):3614–3633, 2021.

Wang, M., Lin, Y., Lin, G., Yang, K., and Wu, X.-m. M2grl: A multi-task multi-view graph representation learning framework for web-scale recommender systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2349–2358, 2020a.

Wang, Z., Tsvetkov, Y., Firat, O., and Cao, Y. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *arXiv preprint arXiv:2010.05874*, 2020b.

Xiao, P., Ban, H., and Ji, K. Direction-oriented multi-objective learning: Simple and provable stochastic algorithms. *arXiv preprint arXiv:2305.18409*, 2023.

Xu, Y., Gui, G., Gacanin, H., and Adachi, F. A survey on resource allocation for 5g heterogeneous networks: Current research, future trends, and challenges. *IEEE Communications Surveys & Tutorials*, 23(2):668–695, 2021.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2636–2645, 2020a.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33: 5824–5836, 2020b.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pp. 1094–1100. PMLR, 2020c.

Zhang, G., Malekmohammadi, S., Chen, X., and Yu, Y. Proportional fairness in federated learning. *Transactions on Machine Learning Research*, 2022.

Zhang, Y. and Yang, Q. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021.

Zhang, Z., Luo, P., Loy, C. C., and Tang, X. Facial landmark detection by deep multi-task learning. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pp. 94–108. Springer, 2014.

Zhang, Z., Yu, W., Yu, M., Guo, Z., and Jiang, M. A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 943–956, 2023.

Zhao, H., Stretcu, O., Smola, A. J., and Gordon, G. J. Efficient multitask feature and relationship learning. In *Uncertainty in Artificial Intelligence*, pp. 777–787. PMLR, 2020.

Zhou, S., Zhang, W., Jiang, J., Zhong, W., Gu, J., and Zhu, W. On the convergence of stochastic multi-objective gradient manipulation and beyond. *Advances in Neural Information Processing Systems*, 35:38103–38115, 2022.

# A. Experiments

## A.1. Toy Example

Following (Navon et al., 2022; Liu et al., 2023), we use a slightly modified version of the 2-task toy example provided in (Liu et al., 2021). The two tasks $L_1(x)$ and $L_2(x)$ are defined on $x = (x_1, x_2)^\top \in \mathbb{R}^2$,

$$L_1(x) = 0.1 \cdot (f_1(x)g_1(x) + f_2(x)h_1(x))$$
$$L_2(x) = f_1(x)g_2(x) + f_2(x)h_2(x),$$

where the functions are given by

$$f_1(x) = \max\big(\tanh(0.5x_2), 0\big)$$
$$f_2(x) = \max\big(\tanh(-0.5x_2), 0\big)$$
$$g_1(x) = \log\Big(\max\big(|0.5(-x_1 - 7) - \tanh(-x_2)|, 0.000005\big)\Big) + 6$$
$$g_2(x) = \log\Big(\max\big(|0.5(-x_1 + 3) - \tanh(-x_2) + 2|, 0.000005\big)\Big) + 6$$
$$h_1(x) = \big((-x_1 + 7)^2 + 0.1(-x_1 - 8)^2\big)/10 - 20$$
$$h_2(x) = \big((-x_1 - 7)^2 + 0.1(-x_1 - 8)^2\big)/10 - 20.$$

The magnitude of the gradient of $L_2(x)$ is larger than $L_1(x)$, posing challenges to the optimization of MTL methods. By choosing different values of $\alpha$, our method covers different ideas of fairness. We use five different starting points $\{(-8.5, 7.5), (0, 0), (9.0, 9.0), (-7.5, -0.5), (9.0, -1.0)\}$. We use Adam optimizer with a learning rate of 1e-3. The training process stops when the Pareto front is reached. The optimization trajectories illustrated in Figure 1 demonstrate that the proposed FairGrad can not only converge to the Pareto front but also exhibit different types of fairness under different choices of $\alpha$.

## A.2. Detailed Results on Multi-Task Regression

We provide more details about per-task results on the QM9 dataset in Table 8. Our FairGrad obtains the best $\Delta m\%$. In addition, as a special case of $\alpha$-fair loss transformation, SI outperforms other methods in 7 tasks, indicating the effectiveness of the transformation.

*Table 8.* Detailed results of on QM9 (11-task) dataset. Each experiment is repeated 3 times with different random seeds and the average is reported.

| METHOD | $\mu$ | $\alpha$ | $\epsilon_{HOMO}$ | $\epsilon_{LUMO}$ | $\langle R^2 \rangle$ | ZPVE | $U_0$ | $U$ | $H$ | $G$ | $c_v$ | MR↓ | $\Delta m\%\downarrow$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MAE ↓ | | | | | | | | | | | | |
| STL | 0.067 | 0.181 | 60.57 | 53.91 | 0.502 | 4.53 | 58.8 | 64.2 | 63.8 | 66.2 | 0.072 | | |
| LS | 0.106 | 0.325 | **73.57** | 89.67 | 5.19 | 14.06 | 143.4 | 144.2 | 144.6 | 140.3 | 0.128 | 8.18 | 177.6 |
| SI | 0.309 | 0.345 | 149.8 | 135.7 | **1.00** | **4.50** | **55.3** | **55.75** | **55.82** | **55.27** | 0.112 | 4.82 | 77.8 |
| RLW | 0.113 | 0.340 | 76.95 | 92.76 | 5.86 | 15.46 | 156.3 | 157.1 | 157.6 | 153.0 | 0.137 | 9.55 | 203.8 |
| DWA | 0.107 | 0.325 | 74.06 | 90.61 | 5.09 | 13.99 | 142.3 | 143.0 | 143.4 | 139.3 | 0.125 | 7.82 | 175.3 |
| UW | 0.386 | 0.425 | 166.2 | 155.8 | 1.06 | 4.99 | 66.4 | 66.78 | 66.80 | 66.24 | 0.122 | 6.18 | 108.0 |
| MGDA | 0.217 | 0.368 | 126.8 | 104.6 | 3.22 | 5.69 | 88.37 | 89.4 | 89.32 | 88.01 | 0.120 | 7.73 | 120.5 |
| PCGRAD | 0.106 | 0.293 | 75.85 | 88.33 | 3.94 | 9.15 | 116.36 | 116.8 | 117.2 | 114.5 | 0.110 | 6.36 | 125.7 |
| CAGRAD | 0.118 | 0.321 | 83.51 | 94.81 | 3.21 | 6.93 | 113.99 | 114.3 | 114.5 | 112.3 | 0.116 | 7.18 | 112.8 |
| IMTL-G | 0.136 | 0.287 | 98.31 | 93.96 | 1.75 | 5.69 | 101.4 | 102.4 | 102.0 | 100.1 | 0.096 | 6.09 | 77.2 |
| NASH-MTL | **0.102** | **0.248** | 82.95 | **81.89** | 2.42 | 5.38 | 74.5 | 75.02 | 75.10 | 74.16 | **0.093** | **3.64** | 62.0 |
| FAMO | 0.15 | 0.30 | 94.0 | 95.2 | 1.63 | 4.95 | 70.82 | 71.2 | 71.2 | 70.3 | 0.10 | 4.73 | 58.5 |
| FAIRGRAD | 0.117 | 0.253 | 87.57 | 84.00 | 2.15 | 5.07 | 70.89 | 71.17 | 71.21 | 70.88 | 0.095 | 3.82 | **57.9** |

## A.3. Detailed Results on Effect of Different Fairness Criteria

We provide additional results of different fairness criteria discussed in Section 5.4 in Table 9 and Table 10. As $\alpha$ changes, the algorithm will prioritize certain tasks over others. Hence, an overall performance drop may be observed. However, tasks with lower priority may get higher ranks, then leading to a higher MR value.

*Table 9.* Results of Different Fairness Criteria on Cityscapes (2-task) dataset. Each experiment is repeated 3 times with different random seeds and the average is reported.

| METHOD | SEGMENTATION | | DEPTH | | MR ↓ | $\Delta m\%$ ↓ |
|---|---|---|---|---|---|---|
| | MIOU ↑ | PIX ACC ↑ | ABS ERR ↓ | REL ERR ↓ | | |
| FAIRGRAD ($\alpha = 1$) | **75.94** | **93.65** | 0.0138 | 33.13 | **2.25** | 6.73 |
| FAIRGRAD ($\alpha = 2$) | 74.10 | 93.03 | 0.0135 | 29.92 | 5.75 | **3.90** |
| FAIRGRAD ($\alpha = 5$) | 67.30 | 90.23 | **0.0134** | 30.01 | 7.25 | 6.87 |
| FAIRGRAD ($\alpha = 10$) | 62.77 | 88.00 | 0.0151 | **28.05** | 8.50 | 10.54 |

*Table 10.* Results of Different Fairness Criteria on NYU-v2 (3-task) dataset. Each experiment is repeated 3 times with different random seeds and the average is reported.

| METHOD | SEGMENTATION | | DEPTH | | SURFACE NORMAL | | | | | MR ↓ | $\Delta m\%$ ↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | ANGLE DISTANCE ↓ | | WITHIN $t^\circ$ ↑ | | | | |
| | MIOU ↑ | PIX ACC ↑ | ABS ERR ↓ | REL ERR ↓ | MEAN | MEDIAN | 11.25 | 22.5 | 30 | | |
| FAIRGRAD ($\alpha = 1$) | **40.64** | **67.20** | 0.5671 | 0.2434 | 25.18 | 20.05 | 28.35 | 55.61 | 68.37 | **4.78** | -2.79 |
| FAIRGRAD ($\alpha = 2$) | 38.80 | 65.29 | **0.5572** | **0.2322** | 24.55 | 18.97 | 30.50 | 57.94 | 70.14 | **4.78** | **-4.96** |
| FAIRGRAD ($\alpha = 5$) | 34.05 | 62.82 | 0.5853 | 0.2375 | **24.40** | **18.70** | **30.96** | **58.48** | **70.45** | 6.22 | -3.03 |
| FAIRGRAD ($\alpha = 10$) | 31.45 | 61.43 | 0.5948 | 0.2341 | 24.80 | 19.08 | 30.10 | 57.58 | 69.69 | 6.22 | -1.00 |

*Table 11.* Results of Different Methods for Solving Sub-problems of Nash-MTL on Cityscapes (2-task) dataset. Each experiment is repeated 3 times with different random seeds and the average is reported.

| METHOD | SEGMENTATION | | DEPTH | | $\Delta m\%$ ↓ |
|---|---|---|---|---|---|
| | MIOU ↑ | PIX ACC ↑ | ABS ERR ↓ | REL ERR ↓ | |
| NASH-MTL (ORIGINAL) | 75.41 | 93.66 | 0.0129 | 35.02 | 6.82 |
| NASH-MTL (OURS) | 75.26 | 93.71 | 0.0129 | 34.45 | **6.28** |

### A.4. Difference Between FairGrad and Nash-MTL

Although FairGrad with proportional fairness shares similarities with Nash-MTL, there are many differences. The high-level ideas are different. Nash-MTL is developed from the perspective of game theory, whereas our FairGrad is inspired by fair resource allocation in communication networks. Thus, this allows us to incorporate the advances from network resource allocation into MTL. FairGrad incorporates other different notions of fairness that Nash-MTL cannot cover. From our empirical studies on Cityscapes and NYUv2 datasets, the performance of MDP fairness is significantly better than proportional fairness. This indicates that proportional fairness may not always be the most suitable choice for different applications and scenarios. This greatly highlights the importance of incorporating other fairness ideas. Unlike Nash-MTL, our proposed FairGrad offers the flexibility to explore different fairness criteria.

Technically, algorithmic designs are different. We propose to solve weights $w_1, ..., w_K$ from our objective through a simple nonlinear least square problem. Solving this problem is efficient, and it turns out that the results are good enough. As a comparison, Nash-MTL solves weights from a different constrained objective via a variation of concave-convex-procedure (CCP) that solves a sequence of simple constrained problems. Our approach that solves a nonlinear least square equation can also be applied to Nash-MTL.

We experiment on the Cityscapes dataset using the codes from the Nash-MTL paper. The results are presented in Table 11, where Nash-MTL (original) denotes the original method, and Nash-MTL (ours) denotes the method using our approach to solve the sub-problems. The results demonstrated that our approach can not only be applied to Nash-MTL, but also achieve slightly better performance.

## B. Proofs

### B.1. $\alpha$-fairness

Different values of $\alpha$ yield different ideas of fairness. Recall from Equation (1) the following utilization maximization objective

$$\max_{x_1,\dots,x_K\in\mathcal{D}} \quad \sum_{i\in[K]} u(x_i) := \frac{x_i^{1-\alpha}}{1-\alpha},$$

where $x_i$ denotes the transmission rate of user $i$, $u(x_i) = \frac{x_i^{1-\alpha}}{1-\alpha}$ is a concave utility function with $\alpha \in [0,1)\cup(1,+\infty)$, and $\mathcal{D}$ is the convex link capacity constraints.

When $\alpha = 0$, the objective is

$$\max_{x_1,\dots,x_K\in\mathcal{D}} \quad \sum_{i\in[K]} u(x_i) := x_i,$$

Note that in our MTL setting, it is similar to the Linear Scalarization.

When $\alpha \to 1$, the utilization maximization objective Equation (1) captures the proportional fairness. First note that

$$\max_{x_1,\dots,x_K\in\mathcal{D}} \quad \sum_{i\in[K]} \frac{x_i^{1-\alpha}}{1-\alpha} = \max_{x_1,\dots,x_K\in\mathcal{D}} \quad \sum_{i\in[K]} \frac{x_i^{1-\alpha}-1}{1-\alpha}.$$

By applying L'Hospital's rule, we have

$$\lim_{\alpha\to 1} \frac{x_i^{1-\alpha}-1}{1-\alpha} = \log x_i.$$

Then, the current objective is

$$\max_{x_1,\dots,x_K\in\mathcal{D}} \quad \sum_{i\in[K]} \log x_i.$$

For a concave function $f(x)$ over a domain $\mathcal{D}$, it is shown in (Srikant & Ying, 2013) that

$$\nabla f(x^\star)(x - x^\star) \le 0 \quad \forall x \in \mathcal{D}. \tag{6}$$

Clearly, since the objective $\sum_{i\in[K]} \log x_i$ is concave, applying Equation (6) yields

$$\sum_{i\in[K]} \frac{x_i - x_i^\star}{x_i^\star} \le 0.$$

If the proportion of one user increases, then there will be at least one other user whose proportional change decreases. The allocation $\{x^\star\}$ captures the proportional fairness. In the MTL setting, the objective becomes

$$\begin{aligned} \max_{d\in B_\epsilon} \quad & \sum_{i\in[K]} \log g_i^\top d \\ \text{s.t.} \quad & g_i^\top d \ge 0, \end{aligned}$$

which corresponds to Nash-MTL (Navon et al., 2022).

When $\alpha \to \infty$, the utilization maximization objective Equation (1) yields the max-min fairness. The following proofs follow Section 2.2.1 in (Srikant & Ying, 2013). Let $x^\star(\alpha)$ be the $\alpha$-fair allocation. Assume $x_i^\star(\alpha) \to x^\star$ as $\alpha \to \infty$ and $x_1^\star < x_2^\star < \cdots < x_K^\star$. Let $\epsilon$ be the minimum difference of $\{x^\star\}$. That is, $\epsilon = \min_i |x_{i+1}^\star - x_i^\star|$, $i \in [K-1]$. When $\alpha$ is

sufficiently large, we then have $|x_i^\star(\alpha) - x_i^\star| \leq \epsilon/4$, which also implies $x_1^\star(\alpha) < x_2^\star(\alpha) < \cdots < x_K^\star(\alpha)$. According to Equation (6), we have

$$\sum_{i\in[K]} \frac{x_i - x_i^\star(\alpha)}{x_i^{\star\alpha}(\alpha)} \leq 0.$$

For any $j \in [K]$, the following inequality always holds

$$\sum_{i=1}^{j}(x_i - x_i^\star(\alpha))\frac{x_j^{\star\alpha}(\alpha)}{x_i^{\star\alpha}(\alpha)} + (x_j - x_j^\star(\alpha)) + \sum_{i=j+1}^{K}(x_i - x_i^\star(\alpha))\frac{x_j^{\star\alpha}(\alpha)}{x_i^{\star\alpha}(\alpha)} \leq 0.$$

Since we have $|x_i^\star(\alpha) - x_i^\star| \leq \epsilon/4$, we then get

$$\sum_{i=1}^{j}(x_i - x_i^\star(\alpha))\frac{x_j^{\star\alpha}(\alpha)}{x_i^{\star\alpha}(\alpha)} + (x_j - x_j^\star(\alpha)) - \sum_{i=j+1}^{K}|x_i - x_i^\star(\alpha)|\frac{(x_j^\star + \epsilon/4)^\alpha}{(x_i^\star - \epsilon/4)^\alpha} \leq 0,$$

where $(x_i - \epsilon/4) - (x_j^\star + \epsilon/4) \geq \epsilon/2$ for any $i > j$. Therefore, when $\alpha$ becomes large enough, the last term in the above inequality will be negligible. Consequently, if $x_j > x_j^\star(\alpha)$, then the allocation for at least one user $i < j$ will decrease. That is, the allocation approaches the max-min fairness when $\alpha \to \infty$. In the context of MTL, this takes the same spirit as MGDA and its variants that aim to maximize the loss decrease for the least-fortune task.

### B.2. Convergence

**Theorem B.1** (Restatement of Theorem 7.3)**.** *Suppose Assumptions 7.1-7.2 are satisfied. Set the stepsize* $\eta_t = \frac{\sum_i w_{t,i}^{-1/\alpha}}{LK\sum_i w_{t,i}^{1-1/\alpha}}$*, Then, there exists a subsequence* $\{\theta_{t_j}\}$ *of the output sequence* $\{\theta_t\}$ *that converges to a Pareto stationary point* $\theta^\star$*.*

*Proof.* Since $(g_i^\top d)^{-\alpha} = w_i$ and $d = \sum_i w_i g_i$ in each iteration, we have the norm $\|d\|^2 = \sum_i w_i g_i^\top d = \sum_i w_i^{1-\frac{1}{\alpha}}$. Each loss function $l_i(\theta)$ is $L$-smooth. Then, we have

$$\begin{aligned} l_i(\theta_{t+1}) &\leq l_i(\theta_t) - \eta_t g_{t,i}^\top d_t + \frac{L}{2}\|\eta_t d_t\|^2 \\ &= l_i(\theta_t) - \eta_t w_{t,i}^{-\frac{1}{\alpha}} + \frac{L}{2}\eta_t^2\|d_t\|^2 \\ &= l_i(\theta_t) - \eta_t w_{t,i}^{-\frac{1}{\alpha}} + \frac{L\eta_t^2}{2}(\sum_{j=1}^{K} w_{t,j}^{1-\frac{1}{\alpha}}). \end{aligned}$$

Set the learning rate $\eta_t = \frac{\sum_{i=1}^K w_{t,i}^{-1/\alpha}}{LK\sum_{i=1}^K w_{t,i}^{1-1/\alpha}}$. Consider the averaged loss function $\mathcal{L}(\theta) = \frac{1}{K}\sum_i l_i(\theta)$, we have

$$\begin{aligned} \mathcal{L}(\theta_{t+1}) &\leq \mathcal{L}(\theta_t) - \eta_t \frac{1}{K}\sum_{i=1}^{K} w_{t,i}^{-\frac{1}{\alpha}} + \frac{L\eta_t^2}{2}(\sum_{i=1}^{K} w_{t,i}^{1-\frac{1}{\alpha}}) \\ &= \mathcal{L}(\theta_t) - L\eta_t^2(\sum_{i=1}^{K} w_{t,i}^{1-\frac{1}{\alpha}}) + \frac{L\eta_t^2}{2}(\sum_{i=1}^{K} w_{t,i}^{1-\frac{1}{\alpha}}) \\ &= \mathcal{L}(\theta_t) - \frac{L\eta_t^2}{2}(\sum_{i=1}^{K} w_{t,i}^{1-\frac{1}{\alpha}}). \end{aligned}$$

It can be observed that $\sum_{\tau=0}^{t}\frac{L\eta_\tau^2}{2}(\sum_{i=1}^K w_{\tau,i}^{1-\frac{1}{\alpha}}) \leq \mathcal{L}(\theta_0) - \mathcal{L}(\theta_{t+1})$. Then, we get

$$\sum_{\tau=0}^{\infty}\frac{L\eta_\tau^2}{2}(\sum_{i=1}^{K} w_{\tau,i}^{1-\frac{1}{\alpha}}) = \frac{1}{2LK^2}\sum_{\tau=0}^{\infty}\frac{(\sum_{i=1}^K w_{\tau,i}^{-\frac{1}{\alpha}})^2}{\sum_{i=1}^K w_{\tau,i}^{1-\frac{1}{\alpha}}} < \infty.$$

Then, it can be obtained that

$$\lim_{\tau\to\infty}\frac{(\sum_{i=1}^{K} w_{\tau,i}^{-\frac{1}{\alpha}})^2}{\sum_{i=1}^{K} w_{\tau,i}^{1-\frac{1}{\alpha}}}=0. \tag{7}$$

From Equation (4), we get

$$\|w_t^{-\frac{1}{\alpha}}\|\geq\sigma_K(G_t^\top G_t)\|w_t\|,$$

where $\sigma_K(G_t^\top G_t)$ is the smallest singular value of matrix $G_t^\top G_t$. Denote $\mathbf{1}=[1,\cdots,1]^\top$ as the length-$K$ vector whose elements are all 1. Note that we have

$$\|w\|^2=\sum_{i=1}^{K} w_i^2\leq\sum_{i=1}^{K} w_i\cdot\sum_{i=1}^{K} w_i=\|w\|_1^2,$$

$$\|w\|_1=\mathbf{1}^\top w\leq\|1\|\cdot\|w\|=\sqrt{K}\|w\|.$$

Combine the above inequalities, we get

$$\|w_t^{-\frac{1}{\alpha}}\|_1\geq\|w_t^{-\frac{1}{\alpha}}\|\geq\sigma_K(G_t^\top G_t)\|w_t\|\geq\frac{1}{\sqrt{K}}\sigma_K(G_t^\top G_t)\|w_t\|_1.$$

Then, we have

$$\frac{\sum_{i=1}^{K} w_{t,i}^{-\frac{1}{\alpha}}}{\sum_{i=1}^{K} w_{t,i}}\geq\frac{1}{\sqrt{K}}\sigma_K(G_t^\top G_t). \tag{8}$$

Furthermore,

$$\begin{aligned}\frac{\sum_{i=1}^{K} w_{t,i}^{-\frac{1}{\alpha}}}{\sum_{i=1}^{K} w_{t,i}}&=\frac{(\sum_{i=1}^{K} w_{t,i}^{-\frac{1}{\alpha}})^2}{(\sum_{i=1}^{K} w_{t,i})\cdot(\sum_{i=1}^{K} w_{t,i}^{-\frac{1}{\alpha}})}\\&=\frac{(\sum_{i=1}^{K} w_{t,i}^{-\frac{1}{\alpha}})^2}{\sum_{i=1}^{K} w_{t,i}^{1-\frac{1}{\alpha}}+\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K} w_{t,i}w_{t,j}^{-\frac{1}{\alpha}}}\\&\leq\frac{(\sum_{i=1}^{K} w_{t,i}^{-\frac{1}{\alpha}})^2}{\sum_{i=1}^{K} w_{t,i}^{1-\frac{1}{\alpha}}}.\end{aligned} \tag{9}$$

For any fixed $K$, it can be concluded from Equation (7), Equation (8), and Equation (9) that

$$\lim_{\tau\to\infty}\sigma_K(G_\tau^\top G_\tau)=0.$$

Since the sequence $\mathcal{L}(\theta_t)$ is monotonically decreasing, we know the sequence $\theta_t$ is in the compact sublevel set $\{\theta|\mathcal{L}(\theta)\leq\mathcal{L}(\theta_0)\}$. Then, there exists a subsequence $\theta_{t_j}$ that converges to $\theta^\star$ where we have $\sigma_K(G_\star^\top G_\star)=0$ and $G_\star$ denotes the matrix of multiple gradients at $\theta^\star$. Therefore, the gradients at $\theta^\star$ are linearly dependent, and $\theta^\star$ is Pareto stationary. □

### B.3. $\alpha$-fair loss transformation

**Proposition B.2** (Restatement of Proposition 6.1)**.** *The Pareto front of the $\alpha$-fair loss functions in Equation (5) is the same as that of original loss functions $(l_1,...,l_K)$.*

*Proof.* If $\theta^\star$ is a Pareto optimal point of $L(\theta)$, then there exists no point $\theta$ dominating $\theta^\star$. That is, we have $l_i(\theta^\star)\leq l_i(\theta)$ for all $i\in[K]$ and $L(\theta^\star)\neq L(\theta)$. Note that the function $f(x)=\frac{x^{1-\alpha}}{1-\alpha}$ with $x>0$ and $\alpha\in[0,1)\cup(1,+\infty)$ is monotonically increasing. It is evident that $\frac{l_i^{1-\alpha}(\theta^\star)}{1-\alpha}\leq\frac{l_i^{1-\alpha}(\theta)}{1-\alpha}$ for all $i\in[K]$. Thus, $\theta^\star$ is also a Pareto optimal point of $\frac{L^{1-\alpha}(\theta)}{1-\alpha}$. Similarly, it can be shown that if $\theta^\star$ is a Pareto optimal point of $\frac{L^{1-\alpha}(\theta)}{1-\alpha}$, it is also a Pareto optimal point of $L(\theta)$. □