FairBranch: Mitigating Bias Transfer in Fair Multi-task Learning

Arjun Roy[†§] arjun.roy@unibw.de

Christos Koutlis[‡] ckoutlis@iti.gr

Symeon Papadopoulos[‡] papadop@iti.gr

Eirini Ntoutsi[§] eirini.ntoutsi@unibw.de

[†]Dept. Math & CSc., Free University of Berlin; [‡]Information Technologies Institute, CERTH

[§]RI CODE, Bundeswehr University, Munich.

***Abstract*—The generalisation capacity of Multi-Task Learning (MTL) suffers when unrelated tasks negatively impact each other by updating shared parameters with conflicting gradients. This is known as *negative transfer* and leads to a drop in MTL accuracy compared to single-task learning (STL). Lately, there has been a growing focus on the fairness of MTL models, requiring the optimization of both accuracy and fairness for individual tasks. Analogously to negative transfer for accuracy, task-specific fairness considerations might adversely affect the fairness of other tasks when there is a conflict of fairness loss gradients between the jointly learned tasks - we refer to this as *bias transfer*. To address both negative- and bias-transfer in MTL, we propose a novel method called *FairBranch*, which branches the MTL model by assessing the similarity of learned parameters, thereby grouping related tasks to alleviate negative transfer. Moreover, it incorporates fairness loss gradient conflict correction between adjoining task-group branches to address bias transfer within these task groups. Our experiments on tabular and visual MTL problems show that *FairBranch* outperforms state-of-the-art MTLs on both fairness and accuracy. Our code is available on github.com/arjunroyihrpa/FairBranch**

***Index Terms*—multitasking, fairness, negative-transfer, bias-transfer, task-grouping**

I. Introduction

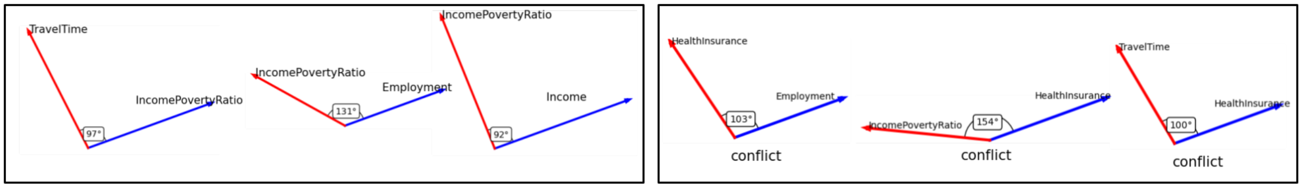
Multi-Task Learning (MTL) traditionally involves deep neural networks trained with fully shared representation layers (parameters) common to all tasks followed by individual task-specific layers to improve model performance across multiple tasks [1]. However, when tasks do not align in their optimisation directions, conflicting updates to the shared parameters may occur, i.e., they may attempt to update the shared parameters with gradients pulling in conflicting directions [2], resulting in performance degradation of the MTL model on specific tasks compared to STL models [3], a phenomenon commonly known as *negative transfer* of knowledge [4].

Lately, there has been a growing focus on the fairness of MTL models [5]–[7], and it is shown that such models can make biased predictions for specific demographic groups characterized by a protected attribute, such as gender or race, across multiple tasks. Fair-MTL methods try to optimize for both accuracy and fairness [5]–[7], by incorporating, for example, a fairness loss alongside the accuracy loss for each task [7], [8]. Analogously to negative transfer for accuracy, bias transfer may occur in fair-MTL, where task-specific fairness considerations could negatively affect the fairness of other tasks, when conflicting fairness loss gradients emerge among jointly learned tasks.

In our paper, we aim to tackle the intertwined challenges of negative transfer and bias transfer in Multi-Task Learning (MTL). Negative transfer in vanilla MTL has been addressed through various methods, including balancing task-specific weights [9], [10], gradient conflict correction [2], [11]–[13], employing branching model architectures [12], [14], [15], and learning separate models for each task-group [16]. While using task-specific weights is cost-effective, determining them poses a significant challenge, especially when considering fairness-accuracy trade-offs for each task. Moreover, methods solely relying on balancing task-weights, correcting gradients, or learning fixed task-group models are constrained by their fixed architecture [17]. Approaches addressing gradient conflicts can be computationally slow, as they necessitate computing and comparing conflicts for every possible task pair in each epoch, a challenge compounded in fair-MTL due to increased possibilities of conflict [12]. Notably, state-of-the-art methods in mitigating negative transfer fail to address fairness conflict issues, leading to bias transfer. In our experiments on the ACS-PUMS dataset (Fig. 1), we illustrate the shortcomings of two prominent MTL methods: TAG [16], which employs task grouping, and Recon [12], which uses gradient correction. These results underscore the inability of negative transfer correction alone to resolve fairness conflicts.

Our proposed solution, *FairBranch*, addresses both *negative transfer* and *bias transfer* by mitigating *negative transfer* through accuracy conflict-aware task grouping and countering *bias transfer* through fairness gradient conflict correction. We create task-group branches based on parameter similarity and correct fairness conflicts within each branch. This branching strategy helps mitigate accuracy loss gradient conflicts, as tasks with similar parameters exhibit similar loss gradient directions. By limiting fairness conflict correction to within task-group branches, our method scales effectively to a large number of tasks.

Our key contributions can be summarised as follows: i) We introduce the study of *bias transfer* (negative transfer of fairness) to enable bias-aware sharing of information among the tasks in MTL. ii) We propose *FairBranch* MTL that



(a) TAG (b) Recon

Fig. 1: Fairness loss gradient conflicts observed in state-of-the-art MTLs addressing negative transfer of accuracy: (a) TAG [16] using task-grouping and (b) Recon [12] using gradient correction on the ACS-PUMS Census Data 2018.

leverages parameter similarity to branch the network, and performs fairness loss gradient correction within each branch to mitigate bias transfer within the task groups. iii) We show that *FairBranch* outperforms state-of-the-art `MTL` methods in addressing negative and bias transfer.

II. Related Works

Related work can be categorized into two broad categories: MTL methods that tackle negative transfer (*negative transfer*) and fairness-aware MTL methods.

Various methods have been proposed to address negative transfer in vanilla MTL, including balancing task-specific weights [9], [10], gradient conflict correction [2], [11]–[13], employing branching model architectures [12], [14], [15], and learning separate models for each task-group [16]. Among these, methods that utilize task-grouping (e.g., TAG [16] and FAFS [14]) or are gradient conflict-aware (e.g., PCGrad [18] and Recon [12]) emerge as direct competitors to our approach. While task-grouping methods compare evaluated task loss output to compute groups, our approach groups tasks based on the learned parameter space, which we show is more effective in addressing the *negative transfer* problem. Our strategy for *negative transfer* is inspired by PCGrad but resolves conflicts only within task branches, requiring fewer conflict corrections and scaling better with a large number of tasks.

Fairness-aware MTL methods can be categorized into in-processing approaches like L2TFMT [6] and WB-fair [7], which modify the objective function by incorporating fairness losses alongside accuracy losses for each task, and post-processing approaches [5] that learn data-driven distance estimators to adjust learned class boundaries. However, none of these prior works explicitly studied the problem of *negative transfer*. Our work falls under the in-processing category of fairness-aware learning, where we branch the model architecture based on parameter similarity in task-groups and then correct fairness conflicts within each task-group to address the joint problem of negative and bias transfer.

We provide a comparative overview of the various methods most relevant to our work in Table I. The evaluation dimensions include whether they address negative transfer, consider fairness, and incorporates dynamic architecture adaptation. Our method is the only one addressing all dimensions.

TABLE I: A comparative overview of SOTA

Methods	Negative Transfer	Fairness	Dynamic Architecture
FAFS [14]	✓	-	✓
TAG [16]	✓	-	-
PCGrad [18]	✓	-	-
Recon [12]	✓	-	✓
L2TFMT [6]	-	✓	-
WB-fair [5]	-	✓	-
FairBranch	✓	✓	✓

III. Background And Motivation

A. Background Setup

We assume a dataset $D = X \times S \times Y$ consisting of m-dimensional *non-protected attributes* $X \in \mathbb{R}^m$, *protected attribute* $S \in \mathbb{S}$, and an output part $Y = Y_1 \times \cdots \times Y_T$ referring to the associated class labels for the output tasks $1, \cdots, T$. For simplicity, we assume binary tasks, i.e., $Y_t \in \{0, 1\}$, $t = 1, \cdots, T$; with 1 representing a positive (e.g., "granted") and 0 representing a negative (e.g.,"rejected") class, and a binary protected attribute: $\mathbb{S} = \{g, \overline{g}\}$, where g and $\overline{g}$ represent demographic groups like "female", and "male".

Let $\mathcal{M}$ be a deep `MTL` model with (d+1) layers, parameterized by the set $\theta \in \Theta$ of parameters, which includes: d layers of shared parameters θ_{sh} (i.e., weights of layers shared by all tasks) connected in order of depth from 1 to d, and for every task t a single layer of task-specific parameters θ_t^{d+1} (i.e., weights of the task specific layers) connected to the topmost shared layer θ_{sh}^d. Formally, we describe the parameters of $\mathcal{M}$ as $\theta = \theta_{sh}^{1,\cdots,d} \times \theta_1^{d+1} \times \cdots \times \theta_T^{d+1}$, where $\theta_{sh}^{1,\cdots,d}$ indicates that shared parameters extends from depth 1 to d, we use θ_α^b to indicate any parameters θ_α at a certain depth b.

Typically, in fair-`MTL` for every task t, $t = 1, \cdots, T$, the goal is to minimize an accuracy loss function $\mathcal{L}_t()$, and a fairness loss function $\mathcal{F}_t()$. In this work, for $\mathcal{L}_t()$ we use the negative log likelihood, and for $\mathcal{F}_t()$ the robust log [6], [19]:

$$\begin{aligned}
\mathcal{L}_t(\theta, X) &= -Y_t \log \mathcal{M}^t(X, \theta) - (1 - Y_t) \log(1 - \mathcal{M}^t(X, \theta))) \\
\mathcal{F}_t(\theta, X, S) &= \sum_{y \in \{1,0\}} \max\big(\mathcal{L}_t(\theta, X \mid Y_t = y, S = g), \\
&\qquad \mathcal{L}_t(\theta, X \mid Y_t = y, S = \overline{g})\big)
\end{aligned} \tag{1}$$

where $\mathcal{M}^t(X, \theta)$ is the outcome $\mathcal{M}$ on task t based on model parameters θ. Note that $\mathcal{F}$ uses an operator (max) over several $\mathcal{L}$ conditioned on different demographics (g, $\overline{g}$), that enables $\mathcal{M}$ to emphasise the demographic group on which it makes the maximum likelihood error. Further, we denote $\nabla_{\theta_\alpha}^{\mathcal{L}_t}$, and

$\nabla_{\theta_\alpha}^{\mathcal{F}_t}$ as the gradient for parameter θ_α w.r.t., loss $\mathcal{L}_t$, and $\mathcal{F}_t$ resp. on task t.

The unfairness of $\mathcal{M}$ on task t can be measured based on the generic framework by [20] as the absolute difference in predictions between g and $\overline{g}$ under a given set of conditions $\mathbb{C}$ which govern the type of fairness definition used:

$$F_{viol}^{(t)}(\mathcal{M}) = \sum_{c \in \mathbb{C}} |P(\mathcal{M}^t(X)|S = g, c) - P(\mathcal{M}^t(X)|S = \overline{g}, c)| \tag{2}$$

In this work, we adopt two popular fairness measures [21]: *Equal Opportunity* ($EP_{viol}^{(t)}(\mathcal{M})$) where $\mathbb{C} : \{[\mathcal{M}^t(X) = 1|Y_t = 1]\}$, and *Equalized Odds* ($EO_{viol}^{(t)}(\mathcal{M})$) where $\mathbb{C} : \{[\mathcal{M}^t(X) = 1, Y_t = 1], [\mathcal{M}^t(X) = 0, Y_t = 1]\}$. EP ephasizes fairness in the positive class, while EO considers fairness across all classes.

B. Negative Transfer and Gradient Conflict

The term *negative transfer* is akin to the concept of negative knowledge gain. Knowledge gain (KG) on a task t by any MTL model $\mathcal{M}$ is assessed as the difference in accuracy between $\mathcal{M}$ and a single-task learner (STL) $\mathcal{H}$ trained on t:

$$KG(t) : P(\mathcal{M}^t(X) = Y_t) - P(\mathcal{H}(X) = Y_t) \tag{3}$$

The ideal scenario is to achieve a positive (or at least non-negative) transfer, i.e., $KG(t) \geq 0$ for all tasks. Any failure to meet this condition is termed as *negative transfer*, where $KG(t) < 0$. Research into conflicting gradients has identified accuracy conflict as the root cause of the *negative transfer* problem [2], [11], [12]. Accuracy conflict between any two task gradients $\nabla_{\theta_\alpha}^{\mathcal{L}t_1}$ and $\nabla\theta_\alpha{}^{\mathcal{L}t_2}$ is defined as:

$$conflict(\nabla\theta_\alpha{}^{\mathcal{L}t_1}, \nabla\theta_\alpha{}^{\mathcal{L}t_2}) : \nabla\theta_\alpha{}^{\mathcal{L}t_1} \cdot \nabla\theta_\alpha{}^{\mathcal{L}t_2} < 0 \tag{4}$$

It follows from Eq. 4 that accuracy conflict happens when $\frac{\pi}{2} < \measuredangle(\nabla_{\theta_\alpha}^{\mathcal{L}_{t_1}}, \nabla_{\theta_\alpha}^{\mathcal{L}_{t_2}}) < -\frac{\pi}{2}$.

C. Bias Transfer and Fairness Conflict

Following the idea of knowledge gain (Eq. 3), we define the concept of discrimination gain (DG) for a given task t as the difference of fairness violation for any `MTL` $\mathcal{M}$ against an `STL` $\mathcal{H}$ on task t:

$$DG(t) : F_{viol}^{(t)}(\mathcal{M}) - F_{viol}^{(t)}(\mathcal{H}) \tag{5}$$

We say a negative gain of fairness *aka bias transfer* is observed when $DG(t) > 0$ for any given t. Notice that contrary to *negative transfer*, the condition for *bias transfer* is attained when the left part of Eq 5 is positive. This is because ideally we want the bias of the `MTL` to be lower than that of `STL`.

We hypothesize that similar to *negative transfer*, *bias transfer* is induced by a gradient conflict, which we term as **fairness conflict**, $conflict(\nabla_{\theta_\alpha}^{\mathcal{F}_{t_1}}, \nabla_{\theta_\alpha}^{\mathcal{F}_{t_2}}) : \nabla_{\theta_\alpha}^{\mathcal{F}_{t_1}} \cdot \nabla_{\theta_\alpha}^{\mathcal{F}_{t_2}} < 0$.

Our aim is to ensure *negative transfer* free and *bias transfer* free learning of an `MTL` $\mathcal{M}$ by ensuring conflict free learning for both accuracy and fairness. Now, unrolling the gradient update to a parameter θ_α for losses $\mathcal{L}$ and $\mathcal{F}$ of any two tasks t_1 and t_2, learned by a vanilla fair-`MTL` with a learning rate η, we have:

$$\begin{aligned} \theta_\alpha \leftarrow \theta_\alpha - \eta \sum_{t \in \{t_1, t_2\}} \nabla_{\theta_\alpha}(\mathcal{L}_t + \lambda_t \mathcal{F}_t) = \theta_\alpha - \eta(\nabla_{\theta_\alpha}^{\mathcal{L}_{t_1}} \\ + \nabla_{\theta_\alpha}^{\mathcal{L}_{t_2}}) - \eta(\lambda_{t_1} \nabla_{\theta_\alpha}^{\mathcal{F}_{t_1}} + \lambda_{t_2} \nabla_{\theta_\alpha}^{\mathcal{F}_{t_2}}) \end{aligned} \tag{6}$$

Now, from Eq. 6 we infer that for any two tasks we can tackle the accuracy conflict and fairness conflict separately.

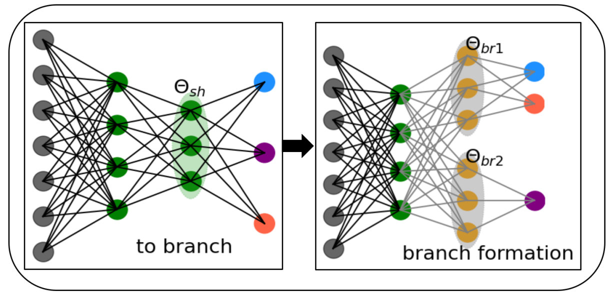


Fig. 2: A High Level Depiction of Branch Formation

IV. FairBranch

Our method, *FairBranch*, addresses both negative transfer (*negative transfer*) and unfair transfer (*bias transfer*). In Algorithm 1, we initialize each task t as a task-group $\{t\}$, with task-specific parameters θ_t and shared parameters $\theta_{sh}^{1,\cdots,d}$ of d layers. TG denotes the collection of all such task-groups. At each training loop, we compute task gradients $\nabla_{\theta_\alpha}^{\mathcal{L}t}$ and $\lambda_t \nabla\theta_\alpha{}^{\mathcal{F}t}$ for each task $t = 1, \cdots, T$ (line 2). Here, λ_t is an intra-task weight addressing accuracy-fairness conflicts, set to 0 when $\nabla\theta_t{}^{\mathcal{L}t} \cdot \nabla\theta_t{}^{\mathcal{F}_t} < 0$.

We then correct fairness conflicts (FBGrad) in each task branch (line 3). After updating $\mathcal{M}$'s parameters (line 4), we check the d_c layer for conditions (line 5) and cluster similar task-groups within TG based on branch parameter similarities at $d_c - 1$ (line 6). We merge task-groups in each cluster by forming branch parameters at d_c exclusive to the cluster, minimizing accuracy conflicts and *negative transfer*. Finally, we update d_c (line 7) and continue training. We detail the Branching mechanism at current depth d_c (Sec. IV-A), and Fairness-conflict correction mechanism on each branch parameter in $\mathcal{M}$ (Sec.IV-B).

Algorithm 1 The FairBranch algorithm

Input: $D = \{(x_i, s_i, y_i^1, \cdots y_i^T)\}_{i=1}^n$, $\mathcal{M}$ parameters: $\theta = \theta_{sh}^{1,\cdots,d} \times \theta_1^{d+1} \times \cdots \times \theta_T^{d+1}$
Initialisation : current layer depth: $d_c \leftarrow d$, $e \leftarrow 0$, $TG \leftarrow \{\{1\}, \cdots, \{T\}\}$
1: **Until** $\{\mathcal{L}_t\}$ and $\{\mathcal{F}_t\}$ convergence do $e \leftarrow e + 1$
2: compute task gradients $\nabla_{\theta_\alpha}^{\mathcal{L}_t}$, $\lambda_t \nabla_{\theta_\alpha}^{\mathcal{F}_t}$; $\forall \theta_\alpha \in \theta$, $t = 1, \cdots T$
3: FBGrad: apply fairness-conflict correction on branches.
4: Update all $\theta_\alpha \in \theta$ (c.f., Eq 6)
5: **if** branch condition is True
6: apply branching mechanism on $\mathcal{M}$
7: $d_c \leftarrow d_c - 1$
8: **End if**
9: **End Until**
Output: fair-Branch MTL $\mathcal{M}$

A. Branching Mechanism

In *FairBranch*, branching occurs only if two conditions are met: a) $|TG| \geq 2$, indicating multiple groups within TG, and b) $d_c > 1$, meaning the current depth is not at the input layer. Once both conditions hold true, branching proceeds with three ordered steps as follows:

1) Measuring task-group affinity:: The pairwise affinity between the task-groups within TG is measured using a parameter similarity function $sim()$ and a threshold hyper-parameter $\tau \in (0, 1]$.

$$\mathcal{A} \leftarrow \{v | v = sim(\theta_{\alpha}^{d_c+1}, \theta_{\beta}^{d_c+1}) \geq \tau; \alpha, \beta \in TG; \alpha \neq \beta\} \quad (7)$$

The idea is to cluster together only task-groups pairs that have similarity higher than or equal to the given threshold. The similarity function ($sim()$) that we use in *FairBranch* is based on central kernel alignment [22], which has gained recent popularity in parameter similarity measures [23]–[25] due to its desired invariant properties [23]. Formally, it is defined as:

$$sim(\theta_{\alpha}, \theta_{\beta}) = cka(K(\theta_{\alpha}), K(\theta_{\beta})) \quad (8)$$

where $\theta_{\alpha}, \theta_{\beta}$ are branch parameters exclusive to task-groups α and β respectively at the layer depth just one above the current depth d_c, $K(\theta_{\alpha}) = \theta_{\alpha}\theta_{\alpha}^{\intercal}$ is a linear kernel function with $\theta_{\alpha}^{\intercal}$ as the transpose of θ_{α}, and cka is kernel alignment measure defined as:

$$cka(K_{\alpha}, K_{\beta}) = \frac{tr(\mathcal{I}(K_{\alpha})\mathcal{I}(K_{\beta}))}{\sqrt{tr(\mathcal{I}(K_{\alpha})\mathcal{I}(K_{\alpha}))tr(\mathcal{I}(K_{\beta})\mathcal{I}(K_{\beta}))}} \quad (9)$$

where $\mathcal{I}$ is a centering function [26], K_{α} is $K(\theta_{\alpha})$, and $tr()$ denotes trace of the resultant centred matrix.

2) Clustering on affinity: We use the affinities $\mathcal{A}$ (Eq. 7) to cluster the task-groups in TG. Although our algorithm offers flexibility in the selection of the clustering method, in our implementation we opted for the Single Linkage Hierarchical Clustering (SLHC) [27]. We start with an empty cluster $\mathcal{C} = \emptyset$, and then recursively include task (or task group) pairs in $\mathcal{C}$, greedily on the basis of $\mathcal{A}$ until $\mathcal{A}$ is $\emptyset$:

$$\begin{aligned} &\text{Until } \mathcal{A} \neq \emptyset : \mathcal{C} \leftarrow \mathcal{C} \cup \{\{\tilde{\alpha}, \tilde{\beta}\}\} | \tilde{\alpha}, \tilde{\beta} = \underset{\alpha,\beta \in TG}{\operatorname{argmax}} \mathcal{A}; \\ &\mathcal{A} \leftarrow \mathcal{A} / \{\{sim(\alpha, \beta) | \alpha = \tilde{\alpha} \vee \beta = \tilde{\beta}; \alpha, \beta \in TG\} \end{aligned} \quad (10)$$

The tasks (or tasks-groups) that are not included in $\mathcal{C}$ are added in TG as a singleton. The main motivation behind finding such binary task groups is to limit the scope of the number of possible conflicts (both accuracy and fairness) between task groups in any given branch, which enables the model to efficiently scale to a large number of tasks.

$$TG \leftarrow Cluster(TG, \mathcal{A}) \cup \{\{\gamma\} | sim(\theta_{\gamma}, \theta_{\alpha}) \notin \mathcal{C}, \gamma, \alpha \in TG\} \quad (11)$$

3) Branch formation: Next, we use the updated task-groups TG (Eq. 11) to form the branches br^{d_c}. Branches are a collection of parameters $br^{d_c} = \{\theta_{br_t} | \theta_{br_t} = copy(\theta_{sh}^{d_c}); t = 1, \cdots, |TG|\}$, where every parameter θ_{br_t} initiates with a replica of the shared parameter $\theta_{sh}^{d_c}$ which is currently being branched. In $\mathcal{M}$, we replace the parameter $\theta_{sh}^{d_c}$ with the parameters collection br^{d_c}, and connect each $\theta_{br_t} \in br^{d_c}$ with θ_{sh}^{d-1} below. Based on the above, each θ_{br_t} is connected with a unique parameter pair $\theta_{\alpha}^{d_c+1}$, $\theta_{\alpha}^{d_c+1}$ s.t., $(\alpha, \beta) \in \tilde{p}$, $\exists! \tilde{p} \in TG$. Fig. 1 depicts a toy example to highlight the architectural change $\mathcal{M}$ undergoes by forming new branches.

B. Fairness Conflict Correction

Denoted by FBGrad, in our FairBranch algorithm this step is responsible to mitigate *bias transfer*. This step is highly motivated from PCGrad update [18] for correcting gradient conflicts. The key difference here is that for the fairness gradient correction instead of the adjusting the gradients at every layer, we look only into the layers that have been branched. The intuition is to apply fairness correction only on parameters without any *negative transfer*, to limit the scope of cross-task fairness-accuracy conflicts (this problem is discussed in detail in Sec. V). To execute this step we look into each of the branch parameters $\theta_{br_t}^{b} \in br^{b}$ and b runs from d to , d_c, and identify the tasks $(t_1, t_2, \cdots)$ connected to $\theta_{br_t}^{b}$.

For each pair (if any) of tasks t_1 and t_2 connected with any branch θ_{br}, we check for fairness conflicts between $\nabla_{\theta_{br}}^{\mathcal{F}_{t_1}}$ and $\nabla_{\theta_{br}}^{\mathcal{F}_{t_2}}$ (c.f. Sec III-B). *Iff* conflict is found we correct both the task gradients by FBGrad function w.r.t. one another, where update of $\nabla_{\theta_{br}}^{\mathcal{F}_{t_1}}$ w.r.t $\nabla_{\theta_{br}}^{\mathcal{F}_{t_2}}$ is defined as:

$$FBGrad : \nabla_{\theta_{br}}^{\mathcal{F}_{t_1}} = \nabla_{\theta_{br}}^{\mathcal{F}_{t_1}} - \frac{\nabla_{\theta_{br}}^{\mathcal{F}_{t_1}} \cdot \nabla_{\theta_{br}}^{\mathcal{F}_{t_2}}}{\|\nabla_{\theta_{br}}^{\mathcal{F}_{t_2}}\|^2} \nabla_{\theta_{br}}^{\mathcal{F}_{t_2}} \quad (12)$$

V. Theoretical Analysis

A. Why Parameter Similarity?

Let us assume any two tasks t_1 and t_2, are identified to form a group by FairBranch. Now, using the Hilbert-Schmidt Independence criterion [28] and Eq. 7 and 8, we get

$$sim(\theta_{t_1}, \theta_{t_2}) \geq \tau \implies \frac{\|\theta_{t_2}^{\intercal}\theta_{t_1}\|_{\mathbb{F}}^2}{\|\theta_{t_1}^{\intercal}\theta_{t_1}\|_{\mathbb{F}}\|\theta_{t_2}^{\intercal}\theta_{t_2}\|_{\mathbb{F}}} \geq \tau \quad (13)$$

where $\|\cdot\|_{\mathbb{F}}$ is the Hilbert-Schmidt norm. Since, we choose $\tau > 0$, we have $tr(\theta_{t_2}^{\intercal}\theta_{t_1}) > 0$ i.e. $\theta_{t_1} \cdot \theta_{t_2} > 0$. Thus, θ_{t_1} and θ_{t_2} are moving in similar directions.

Let $\mathcal{L}_t$ be a Lipschitz continuous and convex [29] task loss, and $\theta_t^{(0)} = C$ be the initial parameters for task t. Also let $\mathcal{L}_t(j)$ be the loss, and $\theta_t(j)$ be the parameter for t at j-th epoch; then after e epochs we have:

$$\theta_t(e) \leftarrow \theta_t(e-1) - \eta\nabla_{\theta_t(e-1)}^{\mathcal{L}_t(e)} = \theta_t(0) - \eta\sum_{j=1}^{e}\nabla_{\theta_t(j-1)}^{\mathcal{L}_t(j)} \quad (14)$$

Since $\theta_{t_1}(0) = \theta_{t_2}(0) = \mathcal{C}$, geometrically we assume $\theta_{t_1}(0)$ and $\theta_{t_2}(0)$ to be the common starting point of t_1 and t_2 (say (0,0) in 2D co-ordinates). Without loss of generality, we can say that $\sum_{j=1}^{e}\nabla_{\theta_t(j-1)}\mathcal{L}_t(j)$ is the resulting gradient vector of all the gradients observed till epoch e for task t. Thus, we can infer that $\sum_{j=1}^{e}\nabla_{\theta_{t_1}(j-1)}\mathcal{L}_{t_1}(j) \cdot \sum_{j=1}^{e}\nabla_{\theta_{t_2}(j-1)}\mathcal{L}_{t_2}(j) > 0$ when $sim(\theta_{t_1}, \theta_{t_2}) \geq \tau$, i.e. the resulting gradient movement for both tasks is in a similar direction. Since, such resulting gradients are accumulated over multiple batches of the data, it is expected to be stable and give a strong estimation of the direction of minima. Henceforth, our intuition is that given a strong similarity ($\tau \rightarrow 1$), we can ensure that the direction of minima of two tasks t_1 and t_2 is similar when $sim(\theta_{t_1}, \theta_{t_2}) \geq \tau$, and thus is expected to move together without any conflict.

B. Why Only Branch Specific Fairness Correction?

In fair-MTL frameworks, at least two different losses($\mathcal{L}_t$ and $\mathcal{F}t$) per task t are accommodated, which can lead to conflicts when gradients from these losses disagree with each other in the direction of update. With T tasks, a fair-`MTL` has

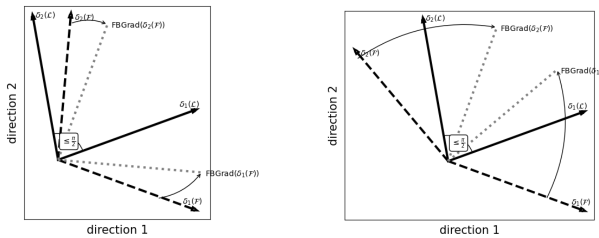


Fig. 3: Example Showing Effect of Fairness Gradient Correction on Task-grouped Branches.

the potential for $T(2T-1)$ conflicts at every θ^d_{sh} layer. For instance, with only two tasks t_1 and t_2, there are $T(2T-1) = 6$ conflicts, including four inter-task conflicts ($(\nabla_{\mathcal{L}_1}, \nabla_{\mathcal{L}_2})$, $(\nabla_{\mathcal{F}_1}, \nabla_{\mathcal{F}_2})$, $(\nabla_{\mathcal{L}_1}, \nabla_{\mathcal{F}_2})$, $(\nabla_{\mathcal{L}_2}, \nabla_{\mathcal{F}_1})$), and two intra-task conflicts ($(\nabla_{\mathcal{L}_1}, \nabla_{\mathcal{L}_2})$). In our algorithm (Sec. IV), we address intra-task conflicts by imposing a strong condition on λ_t and handle inter-task accuracy conflicts ($(\nabla_{\mathcal{L}_1}, \nabla_{\mathcal{L}_2})$) in the created branches by grouping related tasks. Thus, in the branched layers, applying FBGrad not only resolves fairness conflicts ($(\nabla_{\mathcal{F}_1}, \nabla_{\mathcal{F}_2})$) but also reduces the likelihood of inter-task fairness-accuracy conflicts ($(\nabla_{\mathcal{L}_1}, \nabla_{\mathcal{F}_2})$, $(\nabla_{\mathcal{L}_2}, \nabla_{\mathcal{F}_1})$) in most scenarios. Examining potential post-conflict correction scenarios within branches unveils five possibilities, with four leading to FBGrad projecting fairness gradients towards zones free of inter-task fairness-accuracy conflicts. Two illustrative hypothetical examples, showcased in Fig 3, demonstrate these scenarios. However, in shared layers, lacking a branch mechanism precludes such assurances. Correcting fairness conflicts in shared layers might inadvertently worsen inter-task fairness-accuracy conflicts. Our experimental findings will showcase how fairness conflict correction in task-group branches effectively mitigates negative and biased transfer.

VI. Experiments

Datasets: We conduct experiments on two datasets across four setups. The first two setups use tabular data from the *ACS-PUMS* dataset [30], following a protocol of training on one year and testing on the next [6]. The setups are: i) *ACS-PUMS 18-19*, trained on 2018 and tested on 2019 census data, and ii) *ACS-PUMS 19-21*, trained on 2019 and tested on the latest available 2021 census data. We use gender as the protected attribute in both setups. The next two setups are based on the CelebA dataset [31], consisting of celebrity face images. We follow the provided training-test partition. Adopting an existing fair-MTL protocol [6], we create two experiment setups: i) *CelebA gen* with 17 tasks and gender as the protected attribute, and ii) *CelebA age* with 31 tasks and age as the protected attribute. **Competitors:** We compare *FairBranch* with six state-of-the-art MTL methods. The MTL competitors are selected from every direction that our work covers:

- Task-grouping: i) **FAFS** [14], and ii) **TAG** [16].
- Conflict-aware: iii) **PCGrad** [2], and iv) **Recon** [12].
- Fairness-aware: v) **L2TFMT** [6] and vi) **WB-fair** [5].

TABLE II: Comparative Results: $\bar{KG}$ (Higher is better) for Accuracy (Negative Values indicates Negative Transfer), and $\bar{DG}$ (Lower is better) for Fairness (Positive Values indicates Bias Transfer). Best Values in Gray Cell, Second Best underlined.

	Model	Metric		ACS-PUMS 18-19	ACS-PUMS 19-21	CelebA gen	CelebA age
Task-grouping	FAFS	$\bar{KG}$		0.028	0.012	-0.011	-0.024
		$\bar{DG}$	EP	0.009	0.019	0.015	0.017
			EO	0.013	0.020	0.019	0.026
	TAG	$\bar{KG}$		0.022	**0.064**	-0.012	-0.010
		$\bar{DG}$	EP	0.008	0.015	0.015	0.013
			EO	0.014	0.022	0.010	0.017
Conflict aware	PCGrad	$\bar{KG}$		0.015	0.025	0.035	0.025
		$\bar{DG}$	EP	0.004	0.006	0.007	0.009
			EO	0.006	0.006	0.008	0.004
	Recon	$\bar{KG}$		0.025	0.017	0.026	**0.028**
		$\bar{DG}$	EP	0.015	0.014	-0.001	0.005
			EO	0.040	0.036	0.001	0.009
Fairness aware	L2TFMT	$\bar{KG}$		0.024	-0.005	-0.022	-0.020
		$\bar{DG}$	EP	0.001	0.001	-0.002	0.0
			EO	0.002	0.003	0.001	0.003
	WB-fair	$\bar{KG}$		-0.016	0.002	-0.051	-0.080
		$\bar{DG}$	EP	0.001	0.004	0.001	0.002
			EO	0.002	0.006	0.003	0.007
Our	FairBranch	$\bar{KG}$		**0.036**	0.032	**0.036**	0.006
		$\bar{DG}$	EP	**-0.001**	**0.0**	**-0.004**	**-0.001**
			EO	**0.0**	**0.0**	**-0.003**	**0.0**

We implement the methods in their vanilla form[1].

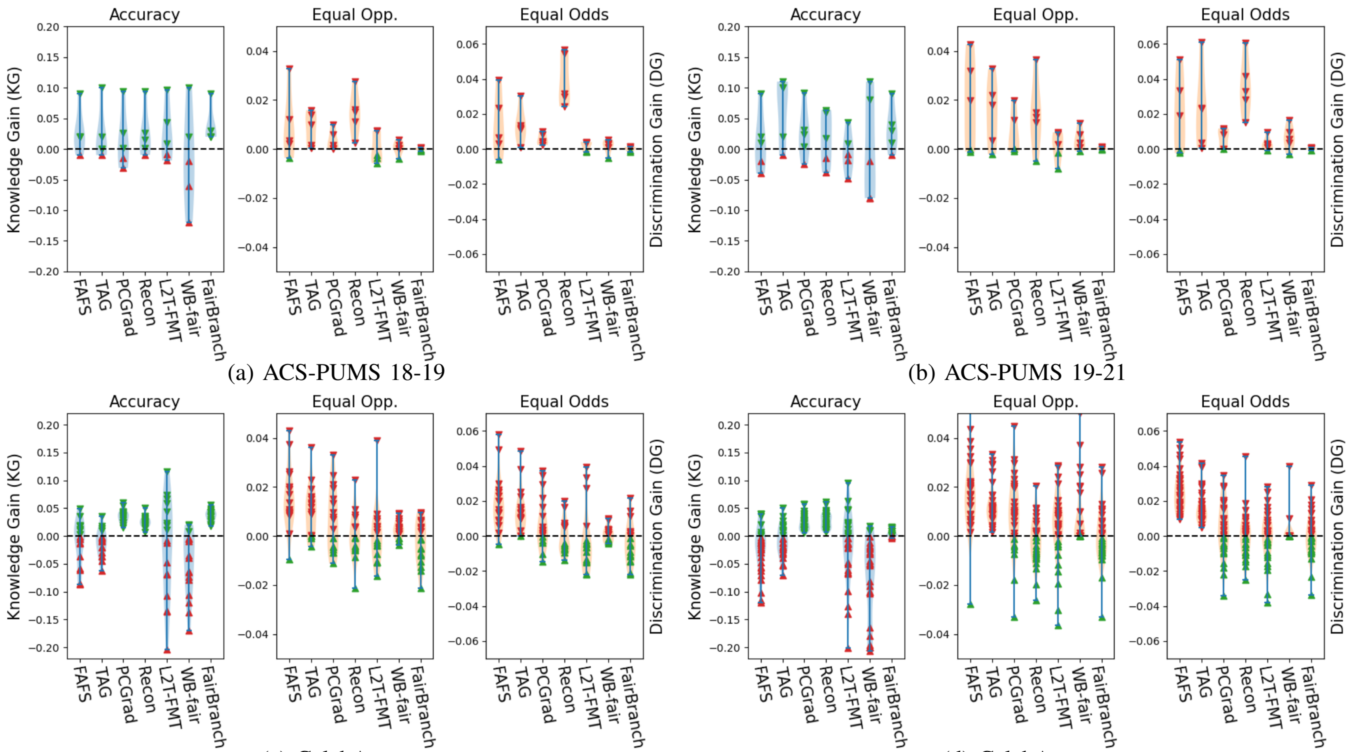
Evaluation Measures: For comparative overall evaluation in Sec. VI-A, we report the average knowledge gain $\bar{KG} = \frac{1}{T}\sum_t KG(t)$ (Eq. 3) for *negative transfer*, and average discrimination gain $\bar{DG} = \frac{1}{T}\sum_t DG(t)$ (Eq. 5) for *bias transfer*. Then, to obtain in-depth per-task performance comparison of the models, we plot the negative and bias transfer distribution over the tasks. For qualitative analysis of *FairBranch* in tackling negative and bias transfer, in Sec. VI-B we present the distribution of fairness conflicts and accuracy conflicts of the learned gradients observed between the tasks while training. To understand which tasks have the most conflicts over the training, we plot the cross-task conflict heat-maps.

Hyperparameters: For tabular setups, we use $\tau = 0.7$, we split the training data into 70:30 training:validation, stratified across all census states. For computer vision setups, we use $\tau = 0.8$, and the predefined training:validation:test split [31].

A. Comparative Results

FairBranch outperforms the competitors on average knowledge and discrimination gain. Table II presents the $\bar{KG}$ and $\bar{DG}$ values of various `MTLs` across different data setups. Notably, *FairBranch* achieves the best outcome in 10 instances and the second best in one out of 12 occasions.

[1]No fairness correction for FAFS, TAG, PCGrad, and Recon.



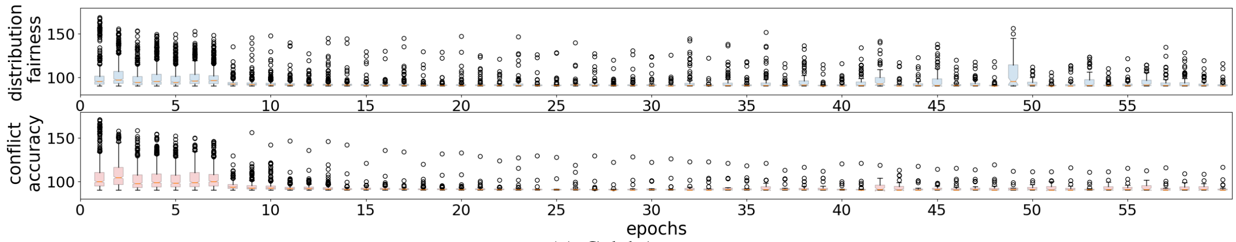
(a) ACS-PUMS 18-19
(b) ACS-PUMS 19-21
(c) CelebA gen
(d) CelebA age

Fig. 4: Comparison on Knowledge Gain (KG) and Discrimination Gain (DG) Distribution: Each box provides comparison on a given Metric Labelled on Top. In boxes every triangle depicts Difference between an MTL with Task Specific STLs. Red Triangles indicates Negative/Bias Transfer and Green indicates Positive/Unbiased Gain. Positive Difference for Accuracy, Negative for Fairness are better.

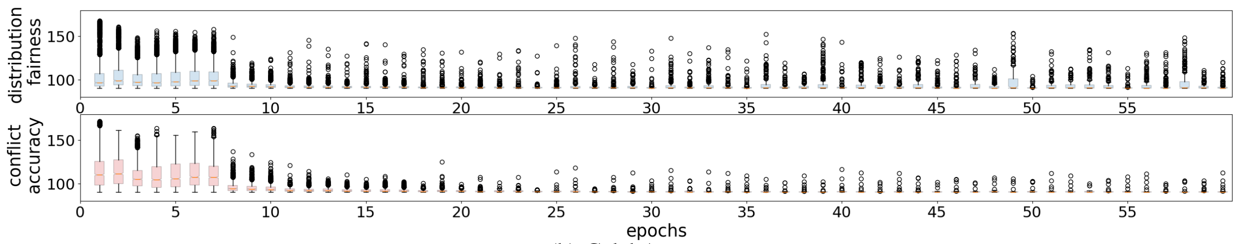
It is only outperformed by *TAG for $\bar{KG}$ on ACS-PUMS 19-21 and by* Recon on CelebA age. Importantly, *FairBranch* consistently achieves positive ($\bar{KG} > 0$) average knowledge gain, addressing *negative transfer*, and non-positive ($\bar{DG} \leq 0$) average discrimination gain, tackling *bias transfer*. Among the competitors, fairness-aware MTLs (L2TFMT and WB-fair) handle discrimination gain better than accuracy-based conflict-aware and task-grouping methods, with L2TFMT having a slight edge over WB-fair. However, none of the competitors achieve negative $\bar{DG}$ values, indicating evidence of *bias transfer* even in fair-MTL. Accuracy-based conflict-aware MTL methods like PCGrad and Recon excel in achieving positive average knowledge gain across all experiment setups. Task-grouping methods FAFS and TAG perform well in addressing *negative transfer* on tabular data but exhibit negative knowledge gain on visual data, indicating signs of *negative transfer*. FairBranch with parameter-based grouping combines the benefits of conflict-awareness and task-grouping, effectively mitigating both negative and bias transfer.

FairBranch tackles *negative transfer* and *bias transfer* better than the competitors. To highlight how *FairBranch* performs against the competitors on *negative transfer* and *bias transfer*, we illustrate in Fig. 4 the distribution of knowledge gain (see Eq. 3) w.r.t., accuracy, and discrimination gain w.r.t., EP, and EO of each MTL over the tasks in each dataset. In each of the boxes, green triangles indicate ('> 0' for accuracy and '< 0' for fairness) a positive/unbiased transfer, while red triangles indicate a negative/bias transfer of knowledge. We first note that overall *FairBranch* predominantly exhibits green triangles in accuracy on all data setups, which verifies the achievement of our goal of avoiding *negative transfer*. On tabular data (Fig 4a and 4b) for both the measures EO and EP, our performance is very close (DG(t)$\approx$ 0) to that of STL in all tasks, thus remaining unaffected from *bias transfer*. On visual data (Fig 4c and 4d), we mostly have unbiased transfer, achieving dense concentration of low green triangles, for both EO and EP. But we still suffer from *bias transfer* in some of the tasks on both data setups. Interestingly, even the fair-MTL methods (L2TFMT and WB-fair) also fail to overcome this challenge, showcasing the difficulty of *bias transfer* under a large number of tasks.

Tackling *negative transfer* on parameter space is advantageous over on output (loss) space. The gradient correction competitors (PCGrad and Recon), although better than *FairBranch* on accuracy by achieving higher positive difference, both fail to tackle *bias transfer* by consistently producing many red triangles across all data setups. Task-grouping methods (FAFS and TAG) tackle with the *negative transfer* in tabular data, but collapse when dealing with a large number of tasks in visual data setups. The finding highlights the advantage of focusing on parameter space (like PCGrad and Recon), rather than on actual output space (like FAFS and



(a) CelebA gen



(b) CelebA age

Fig. 5: Accuracy and Fairness Loss Gradient Conflicts of FairBranch over Training Epochs. Each Box shows Distribution of Angle of Conflict Observed at an Epoch. Less Densely Crowded Lower Boxes are Better.

TAG), and justifies our reason of using parameter similarity to identify task-groups.

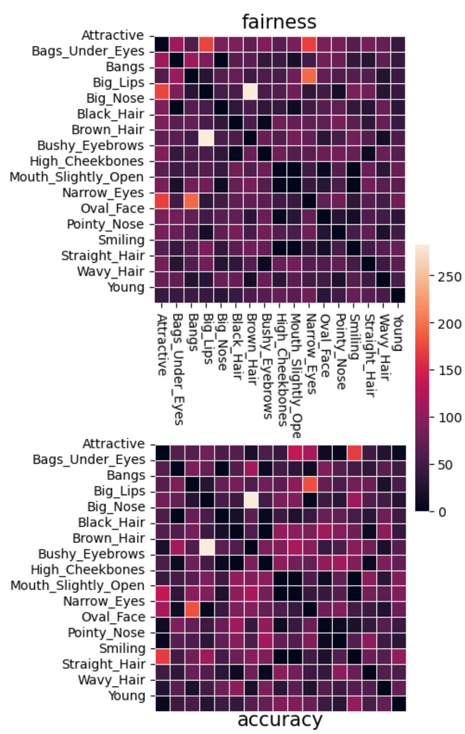


Fig. 6: Heatmap of Accuracy and Fairness Conflicts on CelebA gen. Brighter colour indicates Higher number of Conflicts.

B. Accuracy and Fairness Conflicts

In this section, we aim to analyze the reasons behind the errors observed in *FairBranch* in Section VI-A. Despite overcoming the challenge of *negative transfer* in visual data setups, *FairBranch* still suffers from *bias transfer* in certain tasks. Our hypothesis suggests that while *FairBranch* effectively resolves accuracy conflicts during training, it struggles to completely eliminate fairness conflicts in certain tasks. To verify this, we plot the distribution of accuracy and fairness conflicts in Fig. 5.

In both CelebA gen (Fig.5a) and CelebA age (Fig.5b), *FairBranch* reduces both the frequency and severity of conflicts as training progresses. However, towards the end of training, the accuracy conflict boxes are much smaller than the fairness conflict boxes, consistent with our observations in Section VI-A. We investigate whether conflict occurrence is dominated by a few tasks, given that *bias transfer* is observed in only a few tasks (cf. Fig.4c, 4d). Heatmaps of conflicts between tasks accumulated over training epochs are plotted in Fig6 and 7. While no task is free of either accuracy or fairness conflicts, some task pairs exhibit fewer conflicts over multiple epochs and at multiple layer depths during training.

An intriguing observation is that attribute prediction tasks like 'Attractive' in Fig. 6 and '5_o_Clock_shadow' in Fig. 7 have fewer accuracy conflicts but more fairness conflicts. These pattern suggests that while such tasks contribute positively to accuracy knowledge transfer, they hinder fairness knowledge transfer for most tasks, highlighting the complex decision-making challenges faced by fair-MTL.

VII. Conclusion

We introduced the study of *bias transfer* and showed that learning a fair-MTL model requires to solve the combined problem of *bias transfer* to tackle discrimination and *negative transfer* to tackle accuracy issues. We showed that similar to accuracy conflicts for *negative transfer*, *bias transfer* originates from fairness conflicts between task gradients. We proposed *FairBranch*, an in-processing algorithm that tackles the problem at the level of model parameters using parameter similarity-based branching to alleviate *negative transfer*, and with fairness loss gradients correction for reducing *bias transfer*. Empirically we show that *FairBranch* outperforms many state-of-the-art MTLs for both fairness and accuracy. Our qualitative analysis points out the scalability issues of conflict

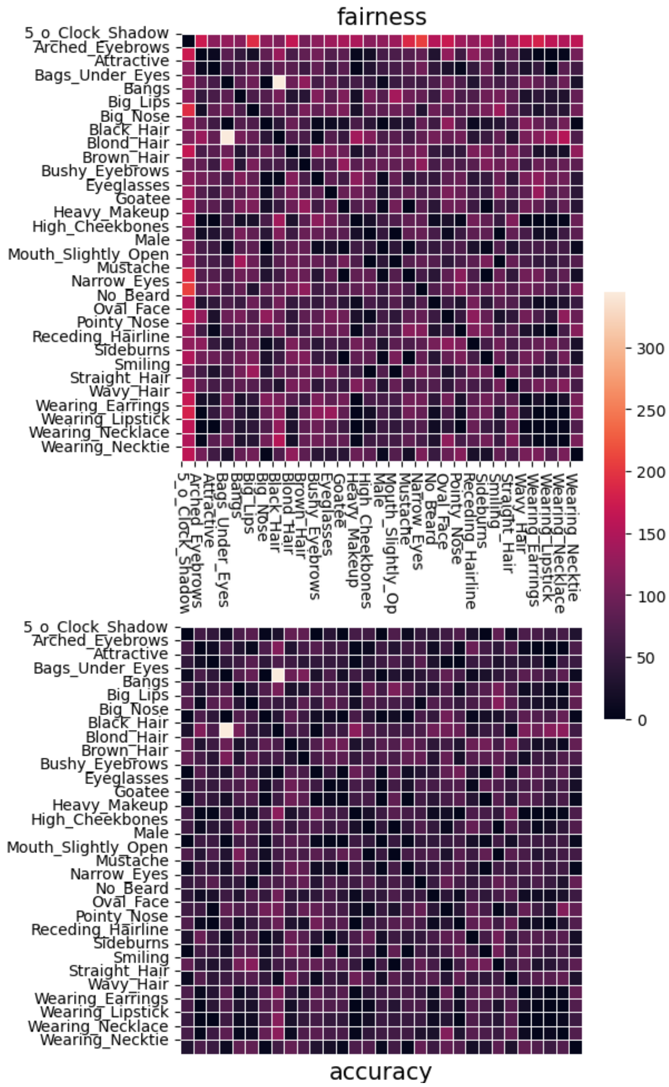


Fig. 7: Heatmap of Accuracy and Fairness Conflicts on CelebA age. Brighter colour indicates Higher number of Conflicts.

occurrence in fair-MTL, and highlights some open challenges for future work.

Acknowledgment

This research work received fund from the European Union under the Horizon Europe MAMMOth project, Grant Agreement ID: 101070285, and also supported by the EU Horizon Europe project STELAR, Grant Agreement ID: 101070122.

References

[1] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE TKDE*, vol. 34, no. 12, pp. 5586–5609, 2022.

[2] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *NeurIPS*, vol. 33, pp. 5824–5836, 2020.

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

[4] T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" in *35th ICML*. PMLR, 2020, pp. 9120–9132.

[5] F. Hu, P. Ratz, and A. Charpentier, "Fairness in multi-task learning via wasserstein barycenters," in *ECMLPKDD*. Cham: Springer Nature Switzerland, 2023, pp. 295–312.

[6] A. Roy and E. Ntoutsi, "Learning to teach fairness-aware deep multi-task learning," in *ECMLPKDD*. Springer, 2022, pp. 710–726.

[7] Y. Wang, X. Wang, A. Beutel, F. Prost, J. Chen, and E. H. Chi, "Understanding and improving fairness-accuracy trade-offs in multi-task learning," in *27th ACM SIGKDD*, 2021, pp. 1748–1757.

[8] D. Pessach and E. Shmueli, "A review on fairness in machine learning," *ACM CSUR*, vol. 55, no. 3, pp. 1–44, 2022.

[9] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *35th ICML*, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 794–803.

[10] Y. Yao, J. Cao, and H. Chen, "Robust task grouping with representative tasks for clustered multi-task learning," in *KDD*, 2019, pp. 1408–1417.

[11] Y. Du, W. M. Czarnecki, S. M. Jayakumar, M. Farajtabar, R. Pascanu, and B. Lakshminarayanan, "Adapting auxiliary losses using gradient similarity," *arXiv preprint arXiv:1812.02224*, 2018.

[12] S. Guangyuan, Q. Li, W. Zhang, J. Chen, and X.-M. Wu, "Recon: Reducing conflicting gradients from the root for multi-task learning," in *11th ICLR*, 2022.

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

[14] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris, "Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification," in *CVPR*, 2017, pp. 5334–5343.

[15] D. Bruggemann, M. Kanakis, S. Georgoulis, and L. Van Gool, "Automated search for resource-efficient branched multi-task networks," in *BMVC*, 2020.

[16] C. Fifty, E. Amid, Z. Zhao, T. Yu, R. Anil, and C. Finn, "Efficiently identifying task groupings for multi-task learning," *NeurIPS*, vol. 34, pp. 27 503–27 516, 2021.

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

[18] V. Kurin, A. De Palma, I. Kostrikov, S. Whiteson, and P. K. Mudigonda, "In defense of the unitary scalarization for deep multi-task learning," *NeurIPS*, vol. 35, pp. 12 169–12 183, 2022.

[19] A. Rezaei, R. Fathony, O. Memarrast, and B. Ziebart, "Fairness for robust log loss classification," in *AAAI*, vol. 34, no. 04, 2020, pp. 5511–5518.

[20] A. Roy, J. Horstmann, and E. Ntoutsi, "Multi-dimensional discrimination in law and machine learning-a comparative overview," in *ACM FAccT*, 2023, pp. 89–100.

[21] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," *NeurIPS*, vol. 29, pp. 3315–3323, 2016.

[22] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *JMLR*, vol. 13, no. 1, pp. 795–828, 2012.

[23] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *ICML*. PMLR, 2019, pp. 3519–3529.

[24] S. Tang, W. J. Maddox, C. Dickens, T. Diethe, and A. Damianou, "Similarity of neural networks with gradients," *arXiv preprint arXiv:2003.11498*, 2020.

[25] A. Csiszárik, P. Kőrösi-Szabó, Á. Matszangosz, G. Papp, and D. Varga, "Similarity and matching of neural network representations," *NeurIPS*, vol. 34, pp. 5656–5668, 2021.

[26] B. O'Neill, "The double-constant matrix, centering matrix and equicorrelation matrix: Theory and applications," *arXiv preprint arXiv:2109.05814*, 2021.

[27] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE NEUR NET*, vol. 16, no. 3, pp. 645–678, 2005.

[28] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with hilbert-schmidt norms," in *ALT*. Springer, 2005, pp. 63–77.

[29] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[30] F. Ding, M. Hardt, J. Miller, and L. Schmidt, "Retiring adult: New datasets for fair machine learning," *NeurIPS*, vol. 34, 2021.

[31] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, December 2015.