Controllable Pareto Trade-off between Fairness and Accuracy

Yongkang Du[1*], **Jieyu Zhao**[1], **Yijun Yang**[2], **Tianyi Zhou**[3]
[1]University of Southern California, [2]University of Technology Sydney, [3]MBZUAI
duyongka@gmail.com, jieyuz@usc.edu
yijun.steven.yang@gmail.com, tianyi.zhou@mbzuai.ac.ae

Abstract

The fairness-accuracy trade-off is a key challenge in NLP tasks. Current work focuses on finding a single "optimal" solution to balance the two objectives, which is limited considering the diverse solutions on the Pareto front. This work intends to provide controllable trade-offs according to the user's preference of the two objectives, which is defined as a reference vector. To achieve this goal, we apply multi-objective optimization (MOO), which can find solutions from various regions of the Pareto front. However, it is challenging to precisely control the trade-off due to the stochasticity of the training process and the high dimensional gradient vectors. Thus, we propose **C**ontrollable **P**areto **T**rade-off (CPT) that can effectively train models to perform different trade-offs according to users' preferences. CPT 1) stabilizes the fairness update with a moving average of stochastic gradients to determine the update direction, and 2) prunes the gradients by only keeping the gradients of the critical parameters. We evaluate CPT on hate speech detection and occupation classification tasks. Experiments show that CPT can achieve a higher-quality set of solutions on the Pareto front than the baseline methods. It also exhibits better controllability and can precisely follow the human-defined reference vectors.

1 Introduction

As language models (LMs) have shown human-level performance on various kinds of tasks, the fairness of LMs over different groups becomes a critical concern in practical applications. Unfairness in LMs can manifest in various ways and across different domains, e.g., LMs trained on biased text corpora can exhibit gender bias in text generation tasks (Wan et al., 2023; Wambsganss et al., 2023; Du et al., 2025), encode societal stereotypes and prejudices present in the training data (Huang et al., 2025; Omrani et al., 2023). Achieving fairness at the group level aim to emphasize that algorithmic decisions neither favor nor harm certain subgroups defined by the sensitive attribute, such as gender, race, religion, age, sexuality, nationality, and health conditions (Chu et al., 2024).

Current methods for group fairness can be divided into three categories (Schumacher et al., 2025; Gallegos et al., 2023). 1) Pre-processing, which aims to balance the training data and prevent unfairness from affecting LMs (Qian et al., 2022; Garimella et al., 2022); 2) In-processing, which optimizes the fairness loss function during the training process (Zhao et al., 2023). The most intuitive strategy might be minimizing a linear combination of fairness and task loss (Roy and Ntoutsi, 2022). Another strategy is constrained optimization, which minimizes the task loss (Cheng et al., 2022) under a fairness constraint; 3) Post-processing, which aims at modifying the LMs's output to achieve group fairness (Liu et al., 2024a; Dhingra et al., 2023). Although the above methods are developed to balance fairness and accuracy, it is still an open challenge for them to precisely control and customize the trade-off.

Pre-processing and post-processing methods may roughly generate models with different preferences, but the effects are limited since they neglect the training process which is usually sensitive and complicated. In this paper, we focus on in-processing methods and intend to train a set of "optimal" models on the Pareto front, which is a set of equilibrium on which one cannot improve an objective without degrading another. While the Pareto front of fairness and accuracy can be highly complicated and contains rich solutions performing different trade-offs, simply combining objectives with different weights cannot guarantee visiting all of them and in the worst case, it may only end up with models optimized for a single objective

[*]Work done during master's study at University of Southern California.

Proceedings of the 6th Workshop on Trustworthy NLP (TrustNLP 2026), pages 108–120

(§4.7.4 of Boyd and Vandenberghe (2004)). Moreover, the conflicts between objectives or constraints can limit the exploration of diverse solutions on the Pareto front. As a result, models struggle to balance multiple objectives.

Multi-objective optimization (MOO) methods such as Multi-Gradient Descent Algorithm (MGDA) are able to converge to a Pareto equilibrium (Désidéri, 2012) by finding a common optimization direction in each step on which all objectives are improving or at least staying the same. Moreover, given a pre-defined reference vector that indicates the preference for different objectives, MGDA has the potential to visit different regions of the Pareto front in the objective space.

However, it is still challenging to directly apply MGDA to fairness-accuracy trade-off when training language models because: 1) MGDA relies on the full gradients of objectives to determine the common optimization direction, while stochastic gradient is more commonly used in training neural networks. Although stochasticity is important to generalization ability, it may lead to a drift of the fairness loss since samples in a mini-batch might not cover all subgroups. 2) The inner products between gradients play an important role in determining the common optimization direction. However, when applied to train language models with millions of parameters, the curse of dimensionality might lead to less informative inner products reflecting the objective correlation. Moreover, many parameters can be pruned without affecting the model performance but they together may contaminate the inner product and thus are detrimental to the search for the common descent direction. 3) It is challenging to control MGDA's optimization trajectory precisely following a pre-defined reference vector.

To overcome these challenges, we propose **C**ontrollable **P**areto Fairness-Accuracy **T**rade-off method (CPT). Our contribution can be summarized as follows:

- We utilize the moving average of stochastic gradients for each objective to approximate the full gradients used in MGDA for finding the common descent direction without missing subgroups.
- We prune the gradient per objective and use a joint mask to reduce all gradients' dimensionality so MGDA can estimate a more precise common descent direction out of the pruned gradients.
- Our experiments on hate speech detection and occupation classification tasks show that CPT, compared to a rich class of baselines, can better follow the reference vectors and find diverse Pareto solutions with different trade-offs, resulting in a better hypervolume on the test set.

2 Related Work

Fairness-Aware Training Recently, fairness-aware training has gained significant traction across diverse domains, including natural language generation (Li et al., 2025), general NLP tasks (Nadeem et al., 2025; Sheng et al., 2021), and multi-task learning (Roy and Ntoutsi, 2022; Oneto et al., 2019). Common approaches are generally categorized into three types: 1) **Regularization**, which introduces penalty terms to decouple model outputs from sensitive attributes, often targeting embeddings (Yang et al., 2023), attention mechanisms (Liu et al., 2024b), or token distributions (Liu et al., 2024a). 2) **Constrained Optimization**, which imposes strict upper bounds on unfairness metrics during the training process (Kim et al., 2018; Cheng et al., 2022). 3) **Adversarial Training**, which employs minimax games between a primary classifier and a sensitive attribute discriminator (Lahoti et al., 2020; Han et al., 2022).

A significant fact for fairness-aware training is the trade-off between fairness and model performance. Dutta et al. (2020) investigates the essential trade-off between fairness and accuracy metrics. Tang et al. (2023) gives fine-grained categories to study properties of fairness-accuracy Pareto front. Kozdoba et al. (2025) proposes a hypernetwork-based approach to achieve scalable trade-offs between fairness and accuracy in large language models with focus on architectural scaling. Here, we focus on the controllability of fairness-accuracy trade-off.

Multi-Objective Optimization Multi-objective optimization (MOO) aims to optimize multiple, often conflicting objectives by finding a representative set of Pareto-optimal solutions. Traditional **gradient-free methods**, such as NSGA-II (Deb et al., 2002) and various evolutionary strategies (Deb, 2011), are robust but often computationally prohibitive for high-dimensional neural network parameters.

In contrast, **gradient-based methods** offer better scalability. The Multiple Gradient Descent Algorithm (MGDA) (Désidéri, 2012) ensures convergence to Pareto-stationary points but suffers from

high computational overhead. To mitigate this, the Stochastic Multi-Gradient Descent (SMSGDA) was introduced (Poirion et al., 2017) and later adapted for fairness tasks (Liu and Vicente, 2022). More recent theoretical refinements have further improved the convergence rates of these stochastic variants (Zhou et al., 2022). To handle diverse user preferences, preference-guided methods like Pareto Multi-Task Learning (PMTL) (Lin et al., 2019) and Exact Pareto Optimal search (EPO) (Mahapatra and Rajan, 2020) have been developed. Furthermore, the emergence of comprehensive libraries like LibMOON (Zhang et al., 2024) has standardized the benchmarking of these techniques.

Despite these advances, the potential of sophisticated gradient-based MOO remains under-explored in the specific context of controllable fairness-accuracy navigation. In this paper, we bridge this gap with CPT, a novel framework designed for precise and controllable trade-off management.

3 Method

In this section, we define fairness-accuracy trade-off as a MOO problem in Section 3.1, introduce the key components of CPT from Section 3.2 to Section 3.4, and give a detailed version of CPT in Section 3.5.

3.1 Fairness-Accuracy Trade-off as MOO

MOO aims at optimizing multiple objectives simultaneously, which can be defined as below.

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \triangleq (\mathcal{L}_1(\boldsymbol{\theta}), \mathcal{L}_2(\boldsymbol{\theta}), \cdots, \mathcal{L}_m(\boldsymbol{\theta}))^\intercal \quad (1)$$

where m is the number of objectives, $\boldsymbol{\theta}$ denotes the parameters to be optimized, $\mathcal{L}_i$ denotes the i-th objective. Instead of finding one single solution in general single-objective optimization, we strive to achieve Pareto stationarity in MOO.

Definition 1 Pareto stationarity for MOO A solution $\boldsymbol{\theta}^*$ is Pareto stationary if there exist $\boldsymbol{\alpha} \in \mathbb{R}^m$ such that $\sum_{i=1}^m \alpha_i = 1$, $\alpha_i \geq 0$, and $\sum_{i=1}^m \alpha_i \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(\boldsymbol{\theta}^*) = \mathbf{0}$, which implies that we cannot find a common updating direction for improving all objectives.

The Pareto front $\mathcal{P}$ represents a set of Pareto stationary solutions, in which each solution achieves a certain trade-off between the objectives. $\mathcal{P}$ forms a boundary in the objective space, and any point inside this boundary represents a suboptimal solution because it can be improved in at least one objective without degrading others.

Definition 2 Fairness-Accuracy Trade-off Given a dataset D with n samples, consisting of input features X, labels Y (c number of classes), sensitive attributes A (such as the demographic group information), and a classifier f parameterized by $\boldsymbol{\theta}$, we utilize CrossEntropy for classification loss $\mathcal{L}_{acc}$, which is defined by Eq. 2.

$$\mathcal{L}_{acc} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} y_{ij} \log(f(x_{ij})) \quad (2)$$

where y_{ij} and $f(x_{ij})$ indicate the label and the prediction of j-th class of i-th sample respectively. We utilize DiffEodd (Chuang and Mroueh, 2021) for fairness loss $\mathcal{L}_{fair}$ (defined by Eq. 4), which is the gap regularization method for equalized odd (EODD) (Hardt et al., 2016). For each $y \in Y$, DiffEodd intends to minimize the gap between the conditional prediction probability given a certain attribute a and the overall prediction probability (see Eq. 3).

$$\begin{aligned} P^y_{all} &= P(f(X)|Y = y), \\ P^y_a &= P(f(X)|A = a, Y = y) \end{aligned} \quad (3)$$

$$\mathcal{L}_{fair} = \sum_{a \in A} \sum_{y \in Y} \left| \mathbb{E}_{x \sim P^y_a} f(x) - \mathbb{E}_{x \sim P^y_{all}} f(x) \right| \quad (4)$$

Our goal is to train f, so that it can perform well on classification tasks and make fair predictions for each subgroup. Then, the fairness-accuracy trade-off problem could be defined as:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \triangleq (\mathcal{L}_{fair}(\boldsymbol{\theta}), \mathcal{L}_{acc}(\boldsymbol{\theta}))^\intercal \quad (5)$$

Definition 3 Common Descent Vector When using the gradient-based optimization algorithm to solve the MOO problem, the common decent vector g provides the direction for optimization and the distance to update along the direction. MGDA defines the common descent vector as the vector with minimum L2 norm in the convex hull of the gradient of each objective (see Eq. 6) so that the objectives will not conflict with each other.

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})\|_2, \text{ s.t. } \|\boldsymbol{\alpha}\|_1 = 1, \boldsymbol{\alpha} \geq \mathbf{0} \quad (6)$$

where $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$ indicates the gradient of the objective function and $\boldsymbol{\alpha}$ is the combination weights. Eq. 6 can be solved with the Frank-Wolfe algorithm (Jaggi, 2013). Then the common descent

vector in multi-objective optimization can be defined as:

$$g \triangleq \sum_{i=1}^{m} \alpha_i \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(\boldsymbol{\theta}) \tag{7}$$

In this paper, we extend MGDA to fairness-accuracy trade-off and propose a novel method named CPT to generate controllable Pareto stationary solutions.

3.2 Moving Average of Stochastic Gradient to Address Fairness Loss Drift

When optimizing one single objective, we usually employ a stochastic approach, where a subset of data is used to compute the mini-batch stochastic gradient. However, directly using stochastic gradients for MOO may not be a wise choice. First, the stochastic nature of the optimization process introduces noise into the gradient, which could be misleading for calculating the common descent vector. Second, as one single mini-batch may not cover all the subgroups, the mini-batch fairness loss as well as its gradient could be inaccurate.

Inspired by SGD with momentum, which intends to stabilize the gradient during optimization, CPT keeps moving average gradients (see Eq. 8) to approximate the whole gradients of objective functions. This method smooths the gradient of each objective before calculating the common descent vector, which leads to a more precise weight for each objective. Also, by accumulating the previous fairness gradients, CPT takes into account those subgroups that might be missing in the current mini-batch, which leads to a better fairness goal.

The moving average gradient of step k is calculated with:

$$\bar{G}^k = \beta * \bar{G}^{k-1} + (1-\beta) * \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \tag{8}$$

where $\bar{G}$ and β are the moving average gradient and the moving average weight.

3.3 Gradient Pruning in MGDA

In addition to refining MGDA with moving average gradient in Section 3.2, we also intend to get a better common descent vector by denoising the gradient vector and lowering its dimension.

When searching the direction for the common descent vector, MGDA uses inner products of gradient vectors (more details in Appendix A.1). However, high-dimension gradients could be dominated by noise, making the common descent vector calculated by MGDA imprecise. Since the parameters with higher values are more influential for the optimization process, we generate a mask based on the parameters' magnitude and filter out the gradients of parameters with low magnitude. The pruning mask $\mathbf{M}$ is initialized as a matrix of ones who has the same shape as $\boldsymbol{\theta}$. Then we apply Alg. 1 to generate the pruning mask $\mathbf{M}$. Given the parameter $\boldsymbol{\theta}$ of the neural network and the pruning ratio γ, we first compute the average magnitude of $\boldsymbol{\theta}$ and get the pruning threshold $\gamma\|\boldsymbol{\theta}\|_1$. Then we iterate the parameter and generate the corresponding pruning mask. The pruned gradient is calculated by:

Algorithm 1: Generation of Pruning Mask

```
Input parameter θ, pruning mask M,
  pruning ratio γ
for θ ∈ θ, M ∈ M do
    if |θ| ≤ γ‖θ‖_1 then
        M ← 0
Output Pruning mask M
```

$$\widetilde{G}_i = \mathbf{M} \odot \bar{G}_i \tag{9}$$

where $\bar{G}_i$ is the moving average gradient of objective i. With gradient pruning, we are able to accelerate the computation as well as get a better common descent vector. In order to keep the theoretical guarantee of MGDA (the objectives will not confilict with each other), we apply the pruned gradient to calculate the combination weights $\boldsymbol{\alpha}$ and the common descent vector g (Eq. 7 is updated by Eq. 10) and only update the parameters that have non-zero gradients.

$$g \triangleq \sum_{i=1}^{m} \alpha_i \widetilde{G}_i \tag{10}$$

3.4 Reference Vector Following

To better control the optimization, CPT utilizes reference vector $\vec{v} = (v_{fair}, v_{acc})$, where $v_{fair}, v_{acc} \in \mathbb{R}$, to guide the optimization process. The reference vector indicates the expected ratio of two loss values. When $\frac{v_{fair}}{v_{acc}} > 1$, we except $\mathcal{L}_{acc}$ to be lower than $\mathcal{L}_{fair}$, which means a preference for accuracy. We define the constraint loss by the Kullback–Leibler (KL) divergence between the loss value vector $\vec{l}$ and the reference vector $\vec{v}$ (see Eq. 11). Different reference vectors set different

constraints for the optimization process and lead to diverse trade-offs on the Pareto set.

$$\Psi(\vec{l},\vec{v}) \triangleq D_{KL}\left(\frac{\vec{l}}{\|\vec{l}\|_1}||\frac{\vec{v}}{\|\vec{v}\|_1}\right), \quad (11)$$

where $\vec{v}=(v_{fair},v_{acc})$ is the reference vector and $\vec{l}=(\mathcal{L}_{fair},\mathcal{L}_{acc})$ is the vector for two objective loss values.

CPT applies two stages for the optimization: correction stage and MOO stage. In the correction stage, CPT applies single objective optimization to satisfy the constraint: $\Psi(\vec{l},\vec{v})<\psi$, where ψ is a predefined threshold. The correction stage provides a suitable starting point for the MOO stage that follows the reference vector $\vec{v}$. In the MOO stage, CPT simultaneously optimizing three objectives including fairness loss $\mathcal{L}_{fair}$, classification loss $\mathcal{L}_{acc}$, and the constraint loss $\Psi(\vec{l},\vec{v})$. Thus, the objective function for MOO stage can be written as:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \triangleq \min_{\boldsymbol{\theta}} \left(\mathcal{L}_{fair}(\boldsymbol{\theta}),\mathcal{L}_{acc}(\boldsymbol{\theta}),\Psi(\vec{l},\vec{v})\right)^{\intercal} \quad (12)$$

3.5 Controllable Pareto Fairness-Accuracy Trade-off

In this section, we provide a detailed version of CPT in Alg. 2, which generates a controllable Pareto fairness-accuracy trade-off. CPT first finds a starting point for multi-objective optimization that satisfies the constraint set by the reference vector $\vec{v}$ in the correction stage. Then it jointly optimizes fairness loss $\mathcal{L}_{fair}$, classification loss $\mathcal{L}_{acc}$, and the constraint loss $\Psi(\vec{l},\vec{v})$ to find the Pareto stationary solution in a certain region in the MOO stage.

The moving average gradients for accuracy $\bar{G}_{acc}$ and fairness $\bar{G}_{fair}$ are updated through the whole optimization process, while $\bar{G}_{kl}$ is updated only in the MOO stage. Meanwhile, CPT prunes the moving average gradient of each objective with the mask $\mathbf{M}$. Finally, CPT computes the common descent vector g and updates the parameters.

Algorithm 2: Training Procedure of CPT

Input dataset $D=\{(X,Y,A)\}$, reference vector $\vec{v}=(v_{\text{fair}},v_{\text{acc}})$, threshold ψ, moving average weight β, pruning ratio γ, learning rate η
Initialize model $f(\boldsymbol{\theta})$, FrankWolfeSolver $\mathtt{F}$ (Jaggi, 2013), $\bar{G}_{\text{fair}},\bar{G}_{\text{acc}},\bar{G}_{\text{KL}} \leftarrow \mathbf{0},\mathbf{0},\mathbf{0}$
for $k=0,\ldots,K$ **do**
  Get pruning mask $\mathbf{M}_k$ by Alg. 1 with γ
  Get $\mathcal{L}^k_{\text{acc}}$ and $\mathcal{L}^k_{\text{fair}}$ by Eq. (2) and Eq. (4)
  /* Gradient Moving Average
  Update $\bar{G}_{\text{fair}}$ and $\bar{G}_{\text{acc}}$ by Eq. (8) with β
  if $\Psi(\boldsymbol{\mathcal{L}},\vec{v})>\psi$ **then**
    /* Correction Stage
    if $\mathcal{L}_{\text{fair}}/\mathcal{L}_{\text{acc}} > v_{\text{fair}}/v_{\text{acc}}$ **then**
      Get descent direction $g=\bar{G}_{\text{fair}}$
    else
      Get descent direction $g=\bar{G}_{\text{acc}}$
  else
    /* MOO Stage
    Update $\bar{G}_{\text{KL}}$ by Eq. (8) with β
    /* Gradient Pruning
    Get $\tilde{G}_{\text{fair}},\tilde{G}_{\text{acc}},\tilde{G}_{\text{KL}}$ by Eq. (9) with $\mathbf{M}_k$
    /* Compute Combination Weights
    $\boldsymbol{\alpha}_k=\mathtt{F}(\tilde{G}_{\text{fair}},\tilde{G}_{\text{acc}},\tilde{G}_{\text{KL}})$
    Get descent vector g by Eq. (10) with $\boldsymbol{\alpha}_k$
  /* Parameter Update
  $\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t-\eta g$
Output Pareto-optimal solution $\boldsymbol{\theta}^*$ following $\vec{v}$

4 Experiments

In this section, we evaluate CPT from the following aspects. 1) Can CPT control the fairness-accuracy trade-off by precise reference vector following? 2) Can CPT generate more diverse trade-off solutions between the two objectives? 3) Can the trade-off solution obtained by CPT generalize to unseen data? Specifically, Section 4.1 describes the experimental setting. Section 4.2 shows the superiority of CPT by comparing it with several state-of-the-art (SoTA) MOO methods. Section 4.3 presents a thorough ablation study to demonstrate the effectiveness of gradient moving average and gradient pruning. We show the result of the case study in Apendix A.3.

4.1 Experimental Setting

Benchmarks We use Jigsaw dataset [1] to evaluate CPT on the toxicity classification task and focus on race bias as it has been proved to show the most significant bias over other attributes (Cheng et al., 2022). In addition, we use BiasBios dataset (De-

[1] https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification/data

Data -set	Method	Accuracy (↑)						EODD (↓)					
		(2,1)	(3,2)	(1,1)	(2,3)	(1,2)	(1,3)	(2,1)	(3,2)	(1,1)	(2,3)	(1,2)	(1,3)
Jigsaw	Scalarization	**+0.48**	+0.07	74.35	-0.15	-0.30	-23.55	+1.54	+0.84	5.86	-0.73	-0.72	**-5.75**
	MGDA	-1.13	-1.13	**71.96**	-0.26	-0.33	-0.51	+1.28	+1.28	**8.20**	+2.68	+4.30	+1.78
	PMTL	**+0.67**	-4.09	72.71	-0.93	-0.54	-0.21	+0.37	-0.34	4.90	+4.37	**-1.02**	-0.32
	EPO	**+0.31**	+0.80	73.63	+0.03	-0.25	-0.54	-1.77	-1.39	6.69	-1.61	**-1.85**	-1.15
	CPT(w/o Prune)	+0.33	**+0.46**	73.48	-0.52	-1.52	-1.26	**-1.44**	+0.16	5.64	+0.12	+1.81	+1.24
	CPT(w/o GA)	**+1.41**	+1.26	71.55	-0.80	-1.11	-2.01	+6.87	+5.26	**1.74**	+0.38	+2.65	+1.83
	CPT	**+1.11**	+0.44	72.09	-0.70	-1.31	-2.29	+4.39	+1.81	3.47	-0.66	-0.89	**-0.92**
BiasBios	Scalarization	**+0.14**	+0.11	91.09	-0.15	-0.27	-0.46	+0.64	+0.22	8.68	-0.79	-1.11	**-1.30**
	MGDA	**+0.02**	-0.03	90.49	+0.01	-0.06	-0.04	+0.00	+0.05	**7.19**	+0.08	+0.11	+0.00
	PMTL	**+0.88**	-0.2	90.24	+0.06	+0.23	+0.26	+0.61	+0.03	+6.94	**-0.079**	+0.00	+0.01
	EPO	**+0.49**	+0.24	90.28	-1.00	-2.36	-6.89	+0.94	+0.5	**7.01**	+0.35	+0.19	+1.17
	CPT(w/o Prune)	**+4.04**	+2.47	84.36	-1.85	-3.38	-6.45	-0.01	-0.02	7.31	-0.23	-0.78	**-2.34**
	CPT(w/o GA)	**+3.4**	+2.16	84.54	-3.19	-6.94	-12.01	-0.60	-0.48	7.50	-0.10	-2.56	**-3.60**
	CPT	**+2.71**	+1.66	85.31	-1.87	-4.27	-9.53	+0.01	-0.09	7.84	-0.37	-0.89	**-3.59**

Table 1: Accuracy and EODD (fairness) trade-off on the test set. The results for reference vector $v = (1, 1)$ are reported in their original values, while the results for the other five reference vectors are differences from metrics achieved at $v = (1, 1)$. For each method, the best accuracy and fairness among the six reference vectors are highlighted by **bold**. The ideal case is that model achieves best accuracy with vector (2,1) and best fairness with vector (1,3). **CPT's fairness and accuracy on the test set better match the reference vectors** than other methods.

Arteaga et al., 2019) to evaluate CPT on the occupation classification task and focus on gender bias. Following Brandl et al. (2023), we use a subset of the original dataset which contains five medical occupations with clear gender imbalance. The statistics of the datasets are shown in Appendix A.5.

We utilize accuracy as the classification metric and EODD as the fairness metric to evaluate CPT. The fairness metric is the difference between true positive rate (TPR) and false positive rate (FPR) under different sensitive attributes and the overall TPR and FPR (see Eq. 13).

$$\sum_{a \in \mathbf{A}} (|\mathrm{TPR}_a - \mathrm{TPR}_{\text{overall}}| + |\mathrm{FPR}_a - \mathrm{FPR}_{\text{overall}}|) \tag{13}$$

A higher accuracy indicates better classification performance and a lower EODD value indicates there is less bias among predictions of different subgroups.

Baselines We compare CPT with several baselines and SoTA MOO methods below:
(1) **Scalarization** that directly optimizes a weighted sum of multiple objectives.
(2) **MGDA (Sener and Koltun, 2018) with diverse initialization**: We first provide MGDA with diverse initial solutions and then apply MGDA to solve the multi-objective optimization problem with respect to each of them.
(3) **Pareto Multi-Task Learning (PMTL)** (Lin et al., 2019) generates solutions falling to different regions of the Pareto front by decomposing a multi-objective optimization problem into multiple sub-problems, each characterized by a distinct preference among those objectives.
(4) **Exact Pareto Optimization (EPO)** (Mahapatra and Rajan, 2020) combines multiple gradient descent with an elaborate projection operator to achieve convergence to the required Pareto solution.
(5) **CPT(w/o Prune)**: CPT without gradient pruning.
(6) **CPT(w/o GA)**: CPT without gradient moving average.

Training details We apply sentence transformer (Reimers and Gurevych, 2019) as the encoder and stack two fully connected layers as classification heads. We use an SGD optimizer with an initial learning rate of 0.01, which is decayed by a small constant factor of 0.8 until the number of epochs reaches a pre-defined value. All of the experiments are conducted on a 4090Ti GPU with four random seeds for fair comparison. More details on the hyperparameters used in training can be found in the Appendix A.4. In order to represent different trade-offs between fairness and accuracy, we set a diverse set of reference vectors: $\mathbb{V} = \{(2, 1), (3, 2), (1, 1), (2, 3), (1, 2), (1, 3)\}$. By optimizing the loss function with the chosen reference vector (see Eq. 11), CPT can precisely control the trade-off between fairness and accuracy.

4.2 Main Results

Controllable Pareto trade-off by following reference vector. In order to demonstrate the ad-

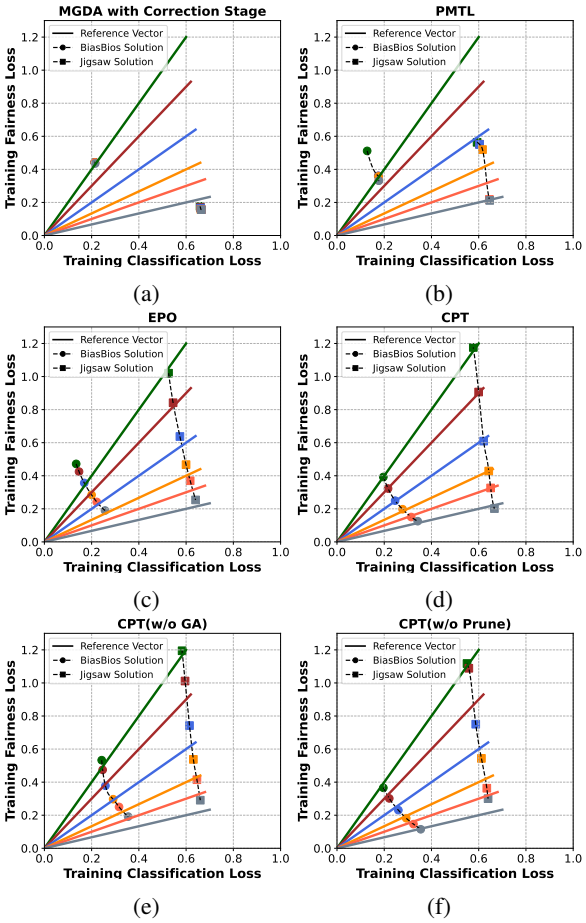


Figure 1: Fairness-accuracy trade-off solutions generated by different methods using six reference vectors. **Among all methods, CPT (Figure 1d) is the best one whose solutions precisely follow the reference vectors**. Reference vectors from top to bottom are $(2,1),(3,2),(1,1),(2,3),(1,2),(1,3)$. The x-axis denotes the classification loss while the y-axis denotes the fairness loss on the training set.

vantage of CPT, we compare it with MGDA with diverse initialization (Figure 1a) and two SoTA reference vector-based MOO methods: PMTL (Figure 1b) and EPO (Figure 1c). As shown in Figure 1, MGDA can only generate one single solution even with different initialization. PMTL fails to generate diverse solutions with given reference vectors, and the solutions are mainly located in two regions. One possible explanation is that PMTL only uses reference vectors to determine initial solutions but lacks a principled method to follow them during the rest of the optimization process.

While EPO achieves lower accuracy and fairness loss values than CPT for vectors with a preference for the accuracy objective (see $\vec{v}=(2,1)$ and $\vec{v}=(3,2)$), this advantage disappears on unseen data. The results in Figure 4d and Figure 4j (in Appendix) indicate that EPO achieves worse fairness performance on the testing set for reference vectors $\vec{v}=(2,1)$ and $\vec{v}=(3,2)$, reflecting that EPO suffers from overfitting to training data. Furthermore, EPO fails to follow the reference vectors with higher fairness preference. This is because EPO uses a noisy stochastic gradient to determine the update direction for each step, which could be inaccurate as we discussed in Section 3.2, and thus the fairness performance is harmed. This challenge is successfully solved by CPT. Benefits from the pruning and moving average of gradients, CPT is able to precisely follow each reference vector. We show that the KL divergence between the reference vector and the loss value vector first decreases and then stays stable for the rest of the training process (see Figure 3 in Appendix), which indicates that the training process is well-guided by the reference vector.

Evaluate solutions' quality with fairness weighted hypervolume. We evaluate CPT on the testing set and show the result in Table 2 and Figure 4 in Appendix. For a fair comparison, we apply the same reference point $(2,1)$ for all methods. Hypervolume (Zitzler and Thiele, 1999) is a widely used metric in MOO. It calculates the area/volume of the resulting set of nondominated solutions with respect to a reference point to measure the diversity of these solutions (more details can be found in Appendix A.2). In the experiment, the reference point is the worst-case result for each objective, i.e., the largest classification and fairness losses (the yellow point on the top right corner in Figure 4).

However, the original hypervolume metric neglects the difficulty of optimization for different objectives and treats them equally. For example, in our case, the fairness loss is harder optimize than the classification loss. In order to address this issue, we utilize a reference point that is more favorable to fairness. As shown in Table 2, CPT and CPT(w/o Prune) achieves the best performance compared with other methods.

Generalizable Pareto trade-off to unseen data. When addressing the fairness-accuracy trade-off in real-world prediction problems, the resulting models are expected to work on training data meanwhile generalizing to unseen data. Hence, a reliable method should achieve a consistent fairness-accuracy trade-off on training and testing sets under the same reference vector. An ideal result in our experiment should satisfy: 1) Achieve the highest

Method	Hypervolume	
	Jigsaw	BiasBios
Scalarization	0.53	0.29
MGDA	0.63	0.33
PMTL	0.69	0.45
EPO	0.72	0.53
CPT(w/o GA)	0.70	0.51
CPT(w/o Prune)	0.71	**0.55**
CPT	**0.73**	0.54

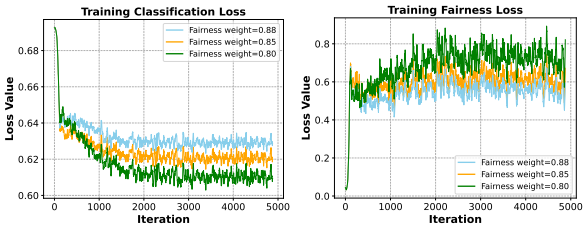
Table 2: Hypervolume (test set) of the solutions achieved by different methods in the fairness-accuracy space. CPT and CPT(w/o Prune) achieve the best hypervolume on the test set.

accuracy when $v = (2, 1)$ and the lowest EODD when $v = (1, 3)$; 2) For reference vectors with preference on accuracy (fairness), the results are expected to show higher (lower) accuracy and higher (lower) EODD than $v = (1, 1)$. As shown in Table 1, only scalarization and CPT exhibit these characteristics. However, when $v = (1, 3)$, the model trained with scalarization performs like a random model on the Jigsaw dataset.

4.3 Ablation study

Here we study how the moving average and pruning of the objectives' gradients affect the performance. Comparing CPT(w/o GA) with CPT in Figure 1, we find that there is a consistent increase of fairness loss for nearly all solutions, demonstrating that the gradient moving average technique can lead to a better fairness performance. On the other hand, when CPT removes the gradient pruning, the optimization process becomes more unstable, highlighting the importance of gradient pruning in stabilizing the optimization and determining a more accurate descent direction.

We then explore how different moving average weights affect the optimization. We set reference vector to $\vec{v} = (1, 1)$, fix the weight for accuracy ($\beta_{acc} = 0.80$), and apply different weights ($\beta_{fair} = \{0.88, 0.85, 0.80\}$) for fairness. The results in Figure 2 indicate that increasing the moving average weight of one objective could make it more dominant in the optimization process. For example, when increasing β_{fair} from 0.80 to 0.88, the fairness loss decreases and the classification loss increases accordingly. Although the model might be sensitive to the weights, it makes the training process more controllable.



(a) Classification Loss (b) Fairness Loss

Figure 2: Moving average weights $\beta_{fair} \in \{0.80, 0.85, 0.88\}$ applied to the fairness gradients when using reference vector $v = (1, 1)$. While the solution associated with $\beta_{fair} = 0.85$ is the closest to v, increasing (decreasing) β_{fair} introduces a bias further minimizing (maximizing) the fairness loss.

5 Conclusions

In this paper, we present CPT, a method for controllable Pareto fairness-accuracy trade-off. CPT provides two techniques to refine the application of the gradient-based multi-objective optimization method in fairness-accuracy trade-off. First, CPT applies moving average gradients instead of stochastic gradients for each objective, which stabilizes the training process and results in better fairness performance. Second, CPT generates a mask based on parameter magnitude to prune the gradient, the denoised low dimensional gradient benefits MOO by providing a more precise common descent vector. We evaluate CPT on real-world datasets and show its advantage in both optimization process and test results. In the future work, we would like to explore how to get a set of Pareto stationary solutions near the reference vector instead of a single solution for each vector.

Limitations

Sensitivity of moving average weights: Although applying moving average gradients can benefit the training process, it could be tedious to tune the moving average weights. And the weights may not always generalize well when training the model on various datasets due to the difference in data distribution. **Trade-off within each class:** We have shown that CPT is able to generate controllable solutions based on the preference of fairness and accuracy over the whole training and test datasets. However, the performance in each class may not follow the preference as discussed in Appendix A.3.

Broader Impact Statement

This work introduces CPT, a controllable multi-objective optimization framework to navigate the

trade-off between fairness and accuracy in NLP models. By enabling precise control via reference vectors, our method provides a principled way for practitioners to align algorithmic decisions with specific societal values and legal requirements. The use of gradient moving averages specifically enhances the protection of minority subgroups, mitigating the risk of bias drift during training. While CPT offers a more diverse and high-quality set of Pareto solutions, we encourage users to remain cautious about class-level generalization and suggest incorporating fine-grained fairness audits in high-stakes deployments.

References

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Stephanie Brandl, Emanuele Bugliarello, and Ilias Chalkidis. 2023. On the interplay between fairness and explainability. *arXiv preprint arXiv:2310.16607*.

Lu Cheng, Suyu Ge, and Huan Liu. 2022. Toward understanding bias correlations for mitigation in nlp. *arXiv preprint arXiv:2205.12391*.

Zhibo Chu, Zichong Wang, and Wenbin Zhang. 2024. Fairness in large language models: A taxonomic survey. *arXiv preprint arXiv:2404.01349*.

Ching-Yao Chuang and Youssef Mroueh. 2021. Fair mixup: Fairness via interpolation. *arXiv preprint arXiv:2103.06503*.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128.

Kalyanmoy Deb. 2011. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*, pages 3–34. Springer.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

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

Harnoor Dhingra, Preetiha Jayashanker, Sayali Moghe, and Emma Strubell. 2023. Queer people are people first: Deconstructing sexual identity stereotypes in large language models. *arXiv preprint arXiv:2307.00101*.

Yongkang Du, Jen-tse Huang, Jieyu Zhao, and Lu Lin. 2025. Faircoder: Evaluating social bias of llms in code generation. *arXiv preprint arXiv:2501.05396*.

Sanghamitra Dutta, Dennis Wei, Hazar Yueksel, Pin-Yu Chen, Sijia Liu, and Kush Varshney. 2020. Is there a trade-off between fairness and accuracy? a perspective using mismatched hypothesis testing. In *International conference on machine learning*, pages 2803–2813. PMLR.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2023. Bias and fairness in large language models: A survey. *arXiv preprint arXiv:2309.00770*.

Aparna Garimella, Rada Mihalcea, and Akhash Amarnath. 2022. Demographic-aware language model fine-tuning as a bias mitigation technique. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 311–319.

Xudong Han, Timothy Baldwin, and Trevor Cohn. 2022. Towards equal opportunity fairness through adversarial learning. *arXiv preprint arXiv:2203.06317*.

Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29.

Jingyuan Huang, Jen-tse Huang, Ziyi Liu, Xiaoyuan Liu, Wenxuan Wang, and Jieyu Zhao. 2025. AI sees your Location—But with a bias toward the wealthy world. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18019–18039, Suzhou, China. Association for Computational Linguistics.

Martin Jaggi. 2013. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML*.

Michael Kim, Omer Reingold, and Guy Rothblum. 2018. Fairness through computationally-bounded awareness. *Advances in neural information processing systems*, 31.

Mark Kozdoba, Binyamin Perets, and Shie Mannor. 2025. Efficient fairness-performance pareto front computation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. 2020. Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems*, 33:728–740.

Haoyuan Li, Rui Zhang, and Snigdha Chaturvedi. 2025. Improving fairness of large language models in multi-document summarization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1143–1154.

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

Suyun Liu and Luis Nunes Vicente. 2022. Accuracy and fairness trade-offs in machine learning: A stochastic multi-objective approach. *Computational Management Science*, 19(3):513–537.

Tianci Liu, Haoyu Wang, Shiyang Wang, Yu Cheng, and Jing Gao. 2024a. Lidao: towards limited interventions for debiasing (large) language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Yan Liu, Yu Liu, Xiaokang Chen, Pin-Yu Chen, Daoguang Zan, Min-Yen Kan, and Tsung-Yi Ho. 2024b. The devil is in the neurons: Interpreting and mitigating social biases in language models. In *The twelfth international conference on learning representations*.

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

Afrozah Nadeem, Mark Dras, and Usman Naseem. 2025. Context-aware fairness evaluation and mitigation in llms. *arXiv preprint arXiv:2510.18914*.

Ali Omrani, Alireza Salkhordeh Ziabari, Charles Yu, Preni Golazizian, Brendan Kennedy, Mohammad Atari, Heng Ji, and Morteza Dehghani. 2023. Social-group-agnostic bias mitigation via the stereotype content model. In *Proc. The 61st Annual Meeting of the Association for Computational Linguistics (ACL2023)*.

Luca Oneto, Michele Doninini, Amon Elders, and Massimiliano Pontil. 2019. Taking advantage of multi-task learning for fair classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 227–237.

Fabrice Poirion, Quentin Mercier, and Jean-Antoine Désidéri. 2017. Descent algorithm for nonsmooth stochastic multiobjective optimization. *Computational Optimization and Applications*, 68:317–331.

Rebecca Qian, Candace Ross, Jude Fernandes, Eric Smith, Douwe Kiela, and Adina Williams. 2022. Perturbation augmentation for fairer nlp. *arXiv preprint arXiv:2205.12586*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Arjun Roy and Eirini Ntoutsi. 2022. Learning to teach fairness-aware deep multi-task learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 710–726. Springer.

Tobias Schumacher, Marlene Lutz, Sandipan Sikdar, and Markus Strohmaier. 2025. Properties of group fairness measures for rankings. *ACM Transactions on Social Computing*, 8(1-2):1–45.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2021. Societal biases in language generation: Progress and challenges. *arXiv preprint arXiv:2105.04054*.

Hua Tang, Lu Cheng, Ninghao Liu, and Mengnan Du. 2023. A theoretical approach to characterize the accuracy-fairness trade-off pareto frontier. *arXiv preprint arXiv:2310.12785*.

Thiemo Wambsganss, Xiaotian Su, Vinitra Swamy, Seyed Parsa Neshaei, Roman Rietsche, and Tanja Käser. 2023. Unraveling downstream gender bias from large language models: A study on ai educational writing assistance. *arXiv preprint arXiv:2311.03311*.

Yixin Wan, George Pu, Jiao Sun, Aparna Garimella, Kai-Wei Chang, and Nanyun Peng. 2023. " kelly is a warm person, joseph is a role model": Gender biases in llm-generated reference letters. *arXiv preprint arXiv:2310.09219*.

Ke Yang, Charles Yu, Yi R. Fung, Manling Li, and Heng Ji. 2023. Adept: a debiasing prompt framework. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Xiaoyuan Zhang, Liang Zhao, Yingying Yu, Xi Lin, Yifan Chen, Han Zhao, and Qingfu Zhang. 2024. Libmoon: A gradient-based multiobjective optimization library in pytorch. *Advances in Neural Information Processing Systems*, 37:2026–2044.

Dora Zhao, Jerone Andrews, and Alice Xiang. 2023. Men also do laundry: Multi-attribute bias amplification. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 42000–42017. PMLR.

Shiji Zhou, Wenpeng Zhang, Jiyan Jiang, Wenliang Zhong, Jinjie Gu, and Wenwu Zhu. 2022. On the convergence of stochastic multi-objective gradient manipulation and beyond. *Advances in Neural Information Processing Systems*, 35:38103–38115.

Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271.

A Appendix

A.1 Computing Common Descent Vector

Considering the case of two objectives, the optimization problem could be defined as

$$\min_{\alpha\in[0,1]} \left\|\alpha\nabla_{\boldsymbol{\theta}}\mathcal{L}_1\left(\boldsymbol{\theta}\right)+(1-\alpha)\nabla_{\boldsymbol{\theta}}\hat{\mathcal{L}}_2\left(\boldsymbol{\theta}\right)\right\|_2^2 \quad (14)$$

Then, the analytical solution for α is:

$$\alpha=\frac{\left(\nabla_{\boldsymbol{\theta}}\mathcal{L}_2(\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\mathcal{L}_1(\boldsymbol{\theta})\right)^T*\nabla_{\boldsymbol{\theta}}\mathcal{L}_2(\boldsymbol{\theta})}{\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_1(\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\mathcal{L}_2(\boldsymbol{\theta})\right\|^2} \quad (15)$$

When it comes to multiple objectives, the calculation of the common descent vector still relies on the inner product.

A.2 Hypervolume

Hypervolume is a valuable metric in multi-objective optimization that measures the quality of a set of solutions by quantifying the objective space they cover. The hypervolume metric can be defined as follows: given a set of points $P \subset \mathbb{R}^n$ and a reference point $\boldsymbol{r} \in \mathbb{R}_+^n$, the hypervolume of $\mathbb{R}$ is measured by the region of non-dominated points bounded above by $\boldsymbol{r}$:

$$HV(P)=\mathrm{VOL}\left(\left\{s\in\mathbb{R}_+^n \mid \exists p\in P:(p\preceq s)\wedge(s\preceq r)\right\}\right) \quad (16)$$

In the bi-optimization problem, it can be represented by the area of the polygon bounded by the solution set and reference point. We show the hypervolume on the test set in Figure 4.

A.3 Case Study

In order to showcase how different reference vectors can affect the model's performance, we conduct a case study on the BiasBios dataset. We first randomly sample 20 cases (10 for male and 10 for female) for each class. Then we feed the samples into models trained with reference vector (2,1) and (1,2) and analyze the model's outputs. We repeat the whole process for three times and present the true positive rate (TPR) and accuracy in Table 3.

Model trained with (2,1) achieves better accuracy in most classes but there is a larger TPR gap between the two groups. Model trained with (1,2) tends to achieve better fairness by minimizing the TPR gap. But it does not apply to all classes, for "Psychologist" and "Dentist" classes, solutions may already satisfy Pareto stationary and there is no update on TPR and accuracy. For "Nurse" and "Surgeon" classes, model sacrifices accuracy to achieve better fairness. For "Physician" class, the TPR for the male group is improved, which leads to a smaller TPR gap as well as higher accuracy.

Class	Metric	Reference Vector	
		(2,1)	(1,2)
Psychologist	TPR(male)	0.90	0.90
	TPR(female)	0.90	0.90
	Accuracy	90.00	90.00
Nurse	TPR(male)	0.50	0.50
	TPR(female)	0.80	0.70
	Accuracy	65.00	60.00
Surgeon	TPR(male)	0.50	0.10
	TPR(female)	0.40	0.10
	Accuracy	45.00	10.00
Dentist	TPR(male)	1.00	1.00
	TPR(female)	1.00	1.00
	Accuracy	100.00	100.00
Physician	TPR(male)	0.80	0.90
	TPR(female)	1.00	1.00
	Accuracy	90.00	95.00

Table 3: True positive rate (TPR) of two groups and accuracy with difference reference vectors. Comparing with model trained with (2,1), the gap between TPR of two groups is reduced when applying model trained with (1,2).

It can be concluded that even though we achieve the controllable trade-offs on the whole dataset via different reference vectors, the preference may not always generalize to each class. The community may need to consider a fine-grained fairness loss function. Overall, the Pareto front of fairness and accuracy is still complicated and worth studying.

A.4 Implementation Details

The version of Sentence Transformer we use is paraphrase-MiniLM-L3-v2 [2]. The classifier consists of two fully connected layers with size (384, 384) and (384,1). We utilize SGD with 0.9 momentum. The learning rate is set to 0.01 initially and decreases every epoch with a 0.8 decay rate. The number of epochs is 40 and the batch size is set to 128. As for hyperparameters related to our method, we set the threshold ψ to be 0.002.

A.5 Dataset Statistics

Table 4 shows the statistics of Jigsaw training set. For positive and negative classes, the data points for each race group are unbalanced. As for BiasBios training set (shown in in Table 5), the distribution

[2] https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L3-v2

Subgroup	Label	
	Positive	Negative
White	5636	5410
Black	3747	3050
Latino	313	497
Asian	183	224

Table 4: Statistics of Jigsaw training-set.

of the female group and male group in each occupation is also unbalanced. The imbalance in the training data brings more unfairness in the model's decision but sometimes could benefit the prediction accuracy, making them suitable datasets to study the trade-off between fairness and accuracy.

Occupation	Gender	
	Female	Male
Psychologist	7491	4400
Surgeon	1203	7424
Nurse	11178	1153
Dentist	3283	6128
Physician	10782	14285

Table 5: Statistics of BiasBios training-set.

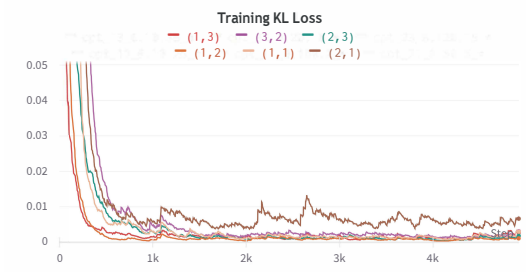
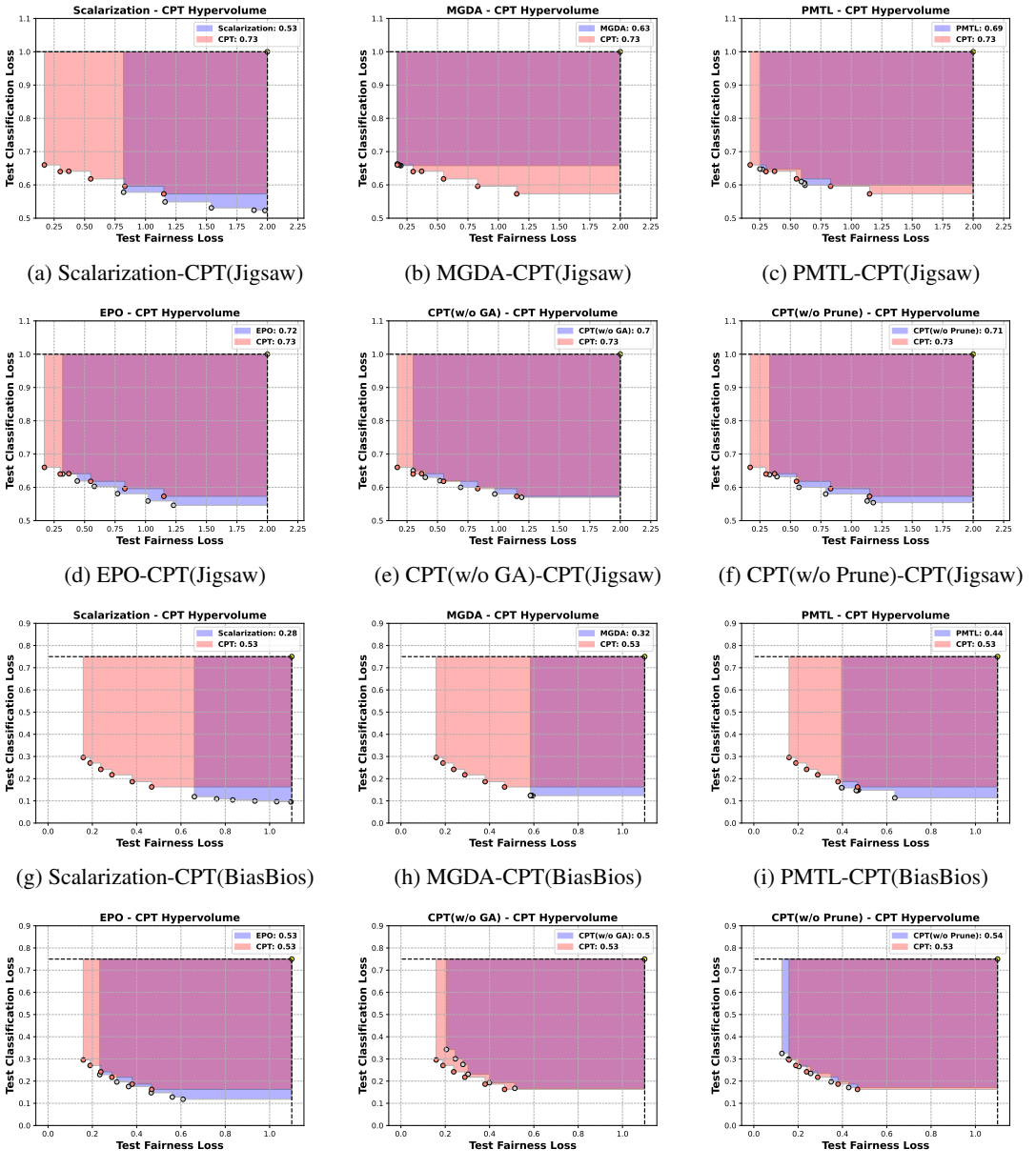


Figure 3: Training KL divergence loss: The KL loss decreases from correction stage to MOO stage and converges at the end of the training, which indicates the optimization process follows the reference vector very well.



(a) Scalarization-CPT(Jigsaw) (b) MGDA-CPT(Jigsaw) (c) PMTL-CPT(Jigsaw)

(d) EPO-CPT(Jigsaw) (e) CPT(w/o GA)-CPT(Jigsaw) (f) CPT(w/o Prune)-CPT(Jigsaw)

(g) Scalarization-CPT(BiasBios) (h) MGDA-CPT(BiasBios) (i) PMTL-CPT(BiasBios)

(j) EPO-CPT(BiasBios) (k) CPT(w/o GA)-CPT(BiasBios) (l) CPT(w/o Prune)-CPT(BiasBios)

Figure 4: Hypervolume (test set) of the solutions achieved by different methods in the fairness-accuracy space. Numerical results are reported in Table 2. **CPT achieves higher hypervolume, indicating the diversity of solutions** that provide different trade-offs.