The Dissertation Committee for Soumyajit Gupta
certifies that this is the approved version of the following dissertation:

Toxic Language and Target Detection under Sparsity by Modeling Group-Specific Representations

Committee:

Matthew Lease, Supervisor

Maria De-Arteaga, Co-supervisor

Qiang Liu

Joydeep Ghosh

Toxic Language and Target Detection under Sparsity by Modeling Group-Specific Representations

by

Soumyajit Gupta

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

April 2025

Abstract

Toxic Language and Target Detection under Sparsity by Modeling Group-Specific Representations

Soumyajit Gupta, PhD
The University of Texas at Austin, 2025

SUPERVISORS: Matthew Lease, Maria De-Arteaga

In the natural language processing (NLP) domain of modeling and mitigating toxic language, it is common to encounter scenarios where multiple tasks and/or objectives are of interest. **Multi-Task Learning** (**MTL**) and **Multi objective Optimization** (**MOO**) are well-established approaches that have seen increasing use with toxic language modeling in recent years [69, 86, 110, 129, 152]. My dissertation consists of two lines of related work on toxic language modeling in NLP based on MTL (*Problem 1* below) and MOO (*Problem 2* below). While my application and experiments focus exclusively on toxic language modeling in NLP, my modeling approaches are more general and could potentially have broader applicability.

PROBLEM 1: MULTI-TASK LEARNING. In developing NLP models for toxicity detection [4, 131, 143], it is often assumed that toxic language manifests similarly across different demographic targeted groups. This typically leads to pooling of data from all groups in model training to learn general patterns of toxicity. However, toxic language directed at different groups can vary quite markedly. Furthermore, an imbalanced group distribution in datasets risks over-fitting to majority groups, potentially at the expense of systematically weaker model performance on minority

group(s). Thus, a "one-size-fits-all" modeling approach maybe sub-optimal, raising concerns of algorithmic fairness [4, 110, 129]. At the same time, radically siloing off datasets for each target group would prevent models from learning broader linguistic patterns of toxicity across different groups being targeted. To characterize this phenomenon where toxic language exhibits both important commonalities and important differences, we borrow the popular phrase of "*same same, but different*" [153].

To address the above issues, we develop a **Conditional MTL** (**CondMTL**) framework (**Chapter 3**), which combines shared and task-specific layers, allowing the model to specialize in detecting toxic language for different target groups while leveraging shared patterns across groups. In this setting, each MTL *task* corresponds to detecting toxic language targeting a specific group. Shared layers benefit from cross-group training, while task-specific layers are trained only on group-specific posts. Given the challenge of sparse labeling for each task *i.e., only a small fraction of the data is labeled for each task*, the CondMTL framework is specifically designed to operate under such setting. We also extend CondMTL to the **SAJ-MTL (Stakeholder-Aware Joint MTL)** framework (**Chapter 4**), that accounts for the *interaction of multiple stakeholder groups*, *i.e.,* annotators and target, to better model the perceived toxicity directed at varied groups. Additionally, the model accounts for *inter-group* and *intra-group* disagreements between annotators. Furthermore, the framework is optimized for *scalable computational efficiency w.r.t.* increasing group cardinality. The results show *improved predictive performance, fairness* across target groups, and *scalability* compared to the latest SoA baselines.

PROBLEM 2: BALANCING COMPETING OBJECTIVES. In parallel to *Problem 1*, accurately detecting the demographic group targeted by toxic language is crucial, since the expression of toxic language varies by target group. A fair detection system should consider the risks of disparate impact of groups. Unlike traditional fairness tasks (*e.g.,* college admissions) that assume asymmetric error costs, errors in target detection are symmetric. For example, we assume that misclassifying a post targeting

group-*Black* as group-*Latinx* is just as problematic as the reverse. This calls for a fairness objective based on equal accuracy across all groups, *i.e., Accuracy Parity* (AP) [160]. However, platforms with skewed user demographics face a trade-off: enforcing equal accuracy for every demographic group *may* may reduce accuracy for dominant group(s). Just as democratic governance must balance majority rule *vs.* minority rights, platforms may need to strike a balance between maximizing absolute fairness for all user groups *vs.* ensuring sufficient service for the majority user group(s) so that the platform can remain in business to serve everyone.

Given this setting, we develop a fairness loss function and a MOO framework. While AP is often mentioned as a fairness metric of potential interest [9, 61, 100], there has been scant research to actually operationalize it. To address this, we propose a new fairness measure, **Group Accuracy Parity (GAP)** (**Chapter 6**), which is differentiable and equivalent to AP, under a binary group setting. To balance this fairness measure *vs.* overall accuracy (driven by the majority group(s)), we develop a multi-objective optimization (MOO) framework **HNPF** [135] with *numerical correctness checks* [53]. We further extend HNPF to a scalable **SUHNPF** [54] frameowrk (**Chapter 5**) that can act as *hypernetworks* for training large scale neural models. This enables us to learn the full trade-off space between GAP *vs.* Overall Accuracy during training, which the user could then flexibly browse at run-time. We conduct experiments and report results for the task of toxic language detection across two demographic groups. We also make extensions to the GAP measure (**Chapter 7**) to account for multiple demographic groups, and conduct different experiments for the task of target-group detection, where we argue for the importance of symmetric error costs. Additionally, we show an incompatibility result between the Accuracy Parity and Equalized Odds measures, addressing a common misconception that balancing equalized odds across groups automatically leads to balanced accuracy across groups.

Table of Contents

Chapter 1: Introduction

The nature and form of toxic language pertaining to different demographic groups can vary quite markedly across groups. The *target group* adds a layer of context because what might be considered toxic when directed at one demographic group might not carry the same meaning and significance when directed at another. A "one-size-fits-all" modeling approach may yield sub-optimal performance by risking over-fitting forms of toxic language most relevant to the majority group(s), potentially at the expense of systematically weaker model performance on minority group(s), thereby raising concerns of algorithmic fairness [4, 110, 129]. Therefore, we require careful consideration of potential biases and disparities in the detection of toxic content across different demographic groups.

> **TLDR:** Given empirical evidence from prior works [38, 50], this thesis posits that the perception of toxicity is a joint interaction of stakeholder identities *i.e.,* annotators and target demographics in our case. We propose a deployable toxicity detection pipeline with two parallel framework paths: a) *Identification of target demographics* from posts; and b) *Group conditioned toxicity detection* from posts. We propose and develop novel architectural designs and evaluation measures (with numerical correctness checks) for model training, and show that these methods lead to improved detection of group-targeted toxicity, while operating on sparse labels, with improved memory and runtime efficiency.

When using terms such as "majority" or "minority", it is important to distinguish between the *statistical minority vs.* a *societal* or *social minority*. By *statistical majority*, we refer to the group(s) that constitutes the largest sized constituent group in a given dataset. In contrast, by *societal majority*, we refer to the group(s) that may hold the most significant social, cultural and political influence within a society, reflecting historical power dynamics and social structures.

For example, according to the 2022 US census [18], the US Caucasian (White) population represents the statistical majority of the US population. We would also

assert that the same population is generally understood to represent the societal majority group as well, *w.r.t.* power and influence. A dataset constructed by random sampling the US population would likely yield the same statistical majority group. Alternatively, one could choose to oversample a societal minority group of interest in order to focus on them in particular, in which case that group might then represent the statistical majority in the dataset. How we sample, however, does not alter which group represents the societal majority. If fairness considerations seek to remedy historical injustices, for example, then we are likely more concerned about the societal majority rather than the statistical majority. This, in turn, may lead us to curate a dataset by such oversampling as described above, such that a societal minority may be elevated to become the statistical majority. In this manner, that group may receive greater benefit from algorithms optimizing overall performance across all instances in a dataset. **In our work, we explicitly focus on societal majority *vs.* minority to account for fairness** in seeking to aid vulnerable groups and historically marginalized communities.

Multi-Task Learning (MTL) treats toxicity detection as a collection of related tasks, each focused on detecting toxicity within a specific demographic group. By jointly training a model to perform multiple group-specific tasks with shared and task specific parameters, we can leverage shared information across groups while also capturing the unique characteristics and nuances of toxicity within each demographic group. This approach promotes the development of models that can generalize well across groups while still accommodating the specificities of each group's language and cultural context, thereby having the potential to achieve better predictive performance on each task than training separate models for each task [19].

Accurately detecting which demographic group is being targeted by toxic language is another important task. A fair and balanced target detection model involves equalizing false positive and false negative rates, *i.e., balanced error across groups* [25, 61], across different minority groups or other targeted vulnerable population, thereby proving fairness and protection to under-represented groups. However, if

the interests and needs of the majority users in a platform are not effectively addressed, there is a risk of losing a significant portion of the user base. Striking the right balance between catering to the majority's preferences and maintaining a fair and inclusive environment becomes crucial, and should be decided by stakeholders (platform administrators, content moderators *etc.*).

Multi-Objective Optimization (MOO) treats target detection as a single classification task to simultaneously optimize for overall accuracy while minimizing the disparities across different demographic group, by training a single model with shared parameters. By tuning the shared parameters, the model learns Pareto trade-off between tasks, where one task cannot improve without detriment to the other [94]. One can trade-off between overall accuracy and group-fairness measures, thereby providing the stakeholders with varying control on the desired deployment policy.

1.1 Motivation for Toxic Language Detection via MTL

Differential subgroup validity (DsGV) [67], also known as subgroup fairness or subgroup equity, refers to the property of a machine learning model or algorithm that ensures consistent and fair performance across different subgroups, by evaluating and mitigating any disparities or biases in the model's predictions for different subgroups. In the context of demographic-targeted toxicity detection, DsGV aims to ensure that the toxicity detection models do not disproportionately favor or penalize specific groups based on their demographic attributes, thereby having consistent performance across various demographic groups. Studies by Sap et al. [129], have highlighted the importance of DsGV in hate speech detection. They emphasize the need to assess and mitigate biases in hate speech detection models to prevent unfair treatment of certain demographic groups.

For standard classification tasks, one seeks to simply maximize overall predictive accuracy. However, when different demographic groups are involved within the dataset, an overall measure does not suffice due to the "same same but differ-

ent" nature of language. The model often misidentifies a harmless minority linguistic pattern as offensive, thereby marking it toxic. There has been a growing interest in using MTL for hate speech and toxicity detection, which involves identifying and flagging online content that is abusive, derogatory, or discriminatory towards a particular group of people [69, 86, 152]. Thus, in this group-targeted setting, we ideally need to optimize the following: a) *Overall Measure*: High predictive performance of the model, independent of the group; or b) *Group Specific Measure*: High predictive performance of the model on specific groups.

In this group-targeted setting (be it Hate Speech Classification [86], Toxicity Detection [143], Fake News Identification [81], Media Bias Estimation [137], Misinformation Detection [89] *etc.*), the **labels for each task are often sparse**, meaning that *only a small fraction of the data is labeled for each task*, making it difficult for traditional MTL frameworks to operate. Although there exist works to tackle this issue of sparse labels *w.r.t.* the loss functions [161] or architectural choices [99], they are mostly guided by heuristics, with limited numerical correctness checks. As such, there are inconsistencies in model behavior when the tasks or datasets vary, leading to the practitioner trying out finite possible model variations to figure out the strategy to settle on for their application.

In this research (**Problem 1**), we aim to provide a more principled way of applying MTL to problem settings involving sparse labels, specifically *w.r.t. learning group-specific representations*, such that the model **improves group-specific performance over target groups**. Therefore, a MTL model is required to maximize representations of individual groups in each task branch, by handling such sparse labeling scenario. The design would lead to selecting only a set of examples in training batch, relevant to the group, to contribute towards calculating the loss for that group and effectively adjust neural weights during backpropagation. This would also require updates to the conventional labeling schema to account for group information for the network to operate on. **Fig. 1.1** shows the ideal MTL setup for group-targeted toxicity detection setting. The shared layers are responsible for learning the general and

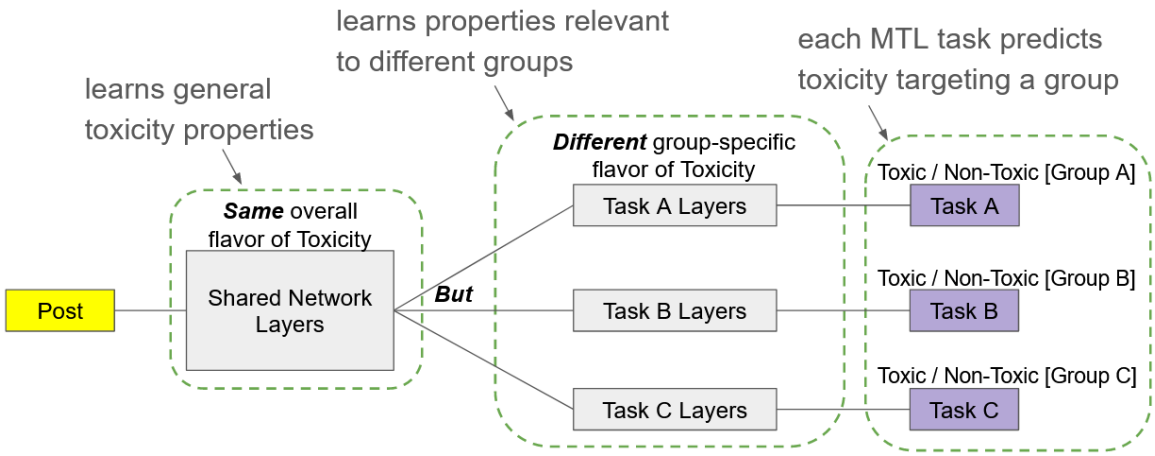


Figure 1.1: Expected Behavior of a Multi-Task Learning (MTL) framework in a group-targeted Toxicity detection setting. The shared layer learns general toxicity properties, while the task-specific layers learn the group-relevant properties of linguistic toxicity. Each task can now maximize it's performance corresponding to it's group, while avoiding overfitting and improved generalization.

overall linguistic toxicity properties, while the task-specific layers picks up the group-specific attributes, thereby catering to each group's predictive performance apart from having high overall predictive accuracy. We aim to maximize group specific toxicity detection performance via the MTL model for all the demographic-target groups.

While group-targeted toxicity detection is important, prior work [50] has also empirically demonstrated that the perceived toxicity is a joint interaction between annotators and targets, as annotators belonging to the same demographic group as the target have a better perception of toxicity due to their lived experience. We therefore take this annotators-target interaction into our modeling approach, which enhances the model's ability to handle multiple stakeholder interactions jointly. We also explicitly account for both inter-group and intra-group disagreements among annotators, following the perception that not only people think differently across demographics, but also individuals within a demographic group might not always think alike. This joint modeling approach enables the system to capture more nuanced perceptions of toxicity as directed at varied target groups, providing a more principled way of learning group-specific representations. We also optimize our framework for

scalability as group cardinality increases, addressing challenges related to expanding demographic categories without compromising computational efficiency.

1.2 Motivation for Balanced Target Detection via MOO

The expression of toxicity varies by target, hence the need for a model to detect group-targeted toxicity. This requires target detection as a pre-processing step, wherein such a target detection system should be fair, providing comparable performance across different groups to ensure non-disparate treament. For well-known fairness tasks associating with providing services (*e.g.,* college admission [72], recidivism [31], hiring [2] *etc.*), we typically assume that errors have asymmetric costs (*e.g.,* errors in being mistakenly granted admission (moderately erroneous) *vs.* being mistakenly denied (severely erroneous) are not equal). However, for our target detection task, errors instead appear to be symmetric: if a toxic post truly targets group-A but is mistakenly detected as targeting group-B, this would be equally bad as a toxic post targeting group-B being mistakenly detected as targeting group-A. This calls for a different fairness objective having symmetric error costs across labels and provide equal accuracy across all demographic groups in target detection.

A standard target detection system would maximize predictive accuracy, which in most cases tends to be biased towards the majority demographic group(s). However, for a fair target detection system, only the overall measure does not suffice, since we want the model to be equally predictive for all groups involved, ensuring fairness *i.e.,* non-disparate impact amongst groups. There has been growing interest in using MOO for toxicity detection, which involves achieving fairness through some balancing measure across all groups [5, 95, 144]. Thus in this group-targeted setting, we ideally need to optimize the following: a) *Overall Measure*: High predictive performance of the model, independent of the group; or b) *Balancing Measure*: Similar predictive measures across all groups involved.

Accuracy Parity [160] or Accuracy Difference [25] are nomenclatures of the

same fairness measure, which demands equal predictive accuracy across all groups involved, thereby ensuring that a classifier is not biased towards the majority group by giving equal exposure to all groups. While achieving a balanced error rate across all groups is an admirable goal, it is important to acknowledge the practical challenges and trade-offs involved. The trade-off between favoring the majority rule *vs.* minority rights decision should be a collaborative effort involving various stakeholders (platform administrators, content moderators, and community), who are better positioned to understand the unique dynamics, priorities, and values of the platform's user base. MOO comes into play here, by allowing us to *navigate the trade-off space* between such competing objectives.

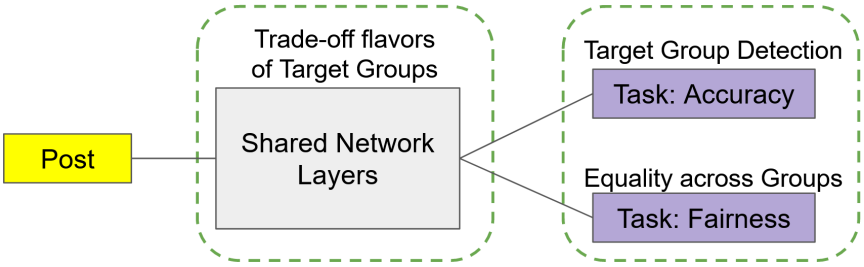


Figure 1.2: Expected Behavior of a Multi-Objective Optimizing (MOO) framework in a group-targeted Toxicity detection setting. The shared layer learns to trade-off between flavors toxic properties, depending on the trade-off. Maximize overall detection accuracy can fit to majority group and poor performance on minority group. Maximizing fairness leads to equal performance across groups, at the cost of drop in overall accuracy. Alternatively, one might choose some trade-off balance in between the two extremes.

In this research (**Problem 2**), we aim to address this fairness concern in *balanced target detection across demographic groups*, via our GAP [57] measure. Additionally, the measure is *differentiable* in nature thereby allowing it to be used to *optimize gradient based models*. We use GAP in conjunction with Overall Accuracy to optimize an MOO model to map out the feasible trade-off space. **Fig. 1.2** shows the ideal MOO setup for group-targeted toxicity detection setting. The shared layers are responsible for learning ideal trade-off between the tasks of overall detection accuracy *vs.* a group-fairness measure. In other words, the model is trading off be-

tween overall performance *vs.* equal group specific performance along some intended practical criteria. We extend our GAP measure to a multi-group setting to **account for more than two groups**, and ground the importance of this measure compared to existing ones from a fairness point of view of target-group detection, requiring symmetric errors. We also show an impposibility results showing that optimizing for Equalized Odds does not necessarily guarantee balanced accuracy across groups.

1.3 Research Questions and Scope of Work

In this section, we summarize the two key problems to be addressed in this dissertation work: 1) better toxicity prediction across different demographic groups; and 2) fair and balanced target detection across different demographics. For each problem, we describe below the underlying broad research questions and further break it down into modeling and application cases.

Regarding my publications to date related to my dissertation work, since 2021 we have published three peer-reviewed articles [53, 54, 56] and posted two pre-prints [57, 135]. Of these five papers, my most recent [56] is related to Problem 1, while the other four are related to Problem 2. This also reflects the larger amount of work we have left to complete on Problem 1 *vs.* Problem 2. Regarding my pre-prints, proposed work will strengthen one [57], with an aim toward publication, while we do not plan to pursue publishing the other [135] at this time.

1.3.1 Problem 1: Differential sub-Group Validity (DsGV)

With DsGV [67], the predictive relationship of a data point *w.r.t.* to its label varies as a function of its group. One naive way of addressing the issue is to train individual models from each group, however those representations will be skewed towards the samples pertaining to that specific group and wouldn't have have any global sense of language. Furthermore, this would lead to huge computational cost to train and deploy individual models in the platform. Additionally, works exist to tackle this

issue of sparse labels *w.r.t.* either the loss functions [161] or architectural choices [99], however, they are mostly guided by heuristics, with limited numerical benchmarking. As such, there are inconsistencies in model behavior when the tasks or datasets vary, leading to the practitioner trying out finite possible model variations to figure out the strategy to settle on for their application. Furthermore, current models do not account for the probabilistic nature of post targets, or consider differences in labeling opinions of annotators both at intra- and inter-group level. Our proposed work aims to address these issues mentioned above by developing architectural and labeling pipelines that are numerically verified for correctness, independent of dataset. This work would result in improved group-specific performance measures under different stakeholder interaction settings.

RQ1: Limitations of applying Traditional MTL to group-specific setting (Chapter 3 and [56]): **a)** How does the nature of task labels in traditional MTL provide barriers to operate under the sparsely labeled group-specific setting, and thereby lead to Label Contamination? **b)** Can we propose an updated labeling schema to provide correct group-specific labels to relevant examples?

RQ2: Addressing group-targeted harm using conditional MTL framework (Chapter 3 and [56])): a) Given the new schema that avoids Label Contamination, can we design an updated MTL framework that accounts for conditional backpropagation on group-relevant examples? **b)** Can this proposed framework account for lower group-specific harm compared to other single-task and multi-task baselines, *w.r.t.* evaluation measures for a stakeholder? **c)** Can the proposed pipeline design address memory efficiency and model runtime concerns *w.r.t.* SoA baselines?

RQ3: Joint multiple stakeholders model for improved toxicity prediction in MTL setting (Chapter 4): a) How can we update the architectural pipeline of CondMTL to learn a joint model tailored for specific stakeholder (annotator - target) interactions? **b)** Can we improve model performance and fairness by taking into account annotator disagreements both at the inter-group and intra-group level

to better reflect the perception of toxicity? **c)** How can existing multi-task learning architectures be extended to remain scalable and computationally efficient as the number of task branches (demographic group-pair) increases? **d)** Can a model better capture this joint annotator-target interaction via group-conditioned losses compared to text-augmentation-based approaches?

1.3.2 Problem 2: Balanced Accuracy Across Groups

With Accuracy Parity [160], the model seeks to achieve equal accuracy measure for all groups involved. The importance of usage of any fairness measures has often not been discussed in literature in the context of the downstream task. Furthermore, most existing fairness measures are probabilistic in nature, hence cannot be used to optimize any gradient-based model. We aim to bridge this gap by proposing a differentiable fairness measure, practically grounding the need for it in specific downstream applications where symmetric errors are required.

RQ4: Fairness measure to optimize Accuracy Parity (Chapter 6) and [57]: a) Can we design a differentiable fairness measure corresponding to *Accuracy Parity*, which accounts for balanced accuracy across groups? **b)** How we use existing MOO frameworks to approximately and efficiently trace out the trade-off space of competing measures?

RQ5: GAP measure for balanced detection accuracy across target-groups (Chapter 7) and [57]: a) From a fairness use-case, how to we ground the importance for such measure in target-group detection task? **b)** What are feasible extensions on the proposed measure to account for multiple demographic groups (beyond binary)? **c)** Are Equalized Odds and Accuracy Parity mutually incompatible?

1.3.3 Proposed Toxicity Detection Pipeline

A robust toxicity detection pipeline needs to address both the identification of target groups and the specific toxicity levels directed toward those groups. To

achieve this, we propose a two-module system that operates in parallel as shown in **Fig. 1.3**. The upper module is a target-group detection system, a multi-label classifier designed to identify which demographic group(s) are being targeted in a given post. Since a single post may be aimed at multiple groups, this classifier allows for flexible, multi-label outputs, ensuring that the model captures all potential target groups involved. The lower module is a toxicity detection system, which assigns group-specific toxicity labels, indicating whether a post is toxic when viewed from the perspective of each targeted group. By splitting these tasks, the pipeline allows for more nuanced toxicity detection: the output of the target-group detection module informs the toxicity detection module about which group-specific branches to activate. This design ensures that the toxicity assessment is contextually aware, meaning it can differentiate between general toxic language and toxicity that is particularly harmful when directed at a specific group.

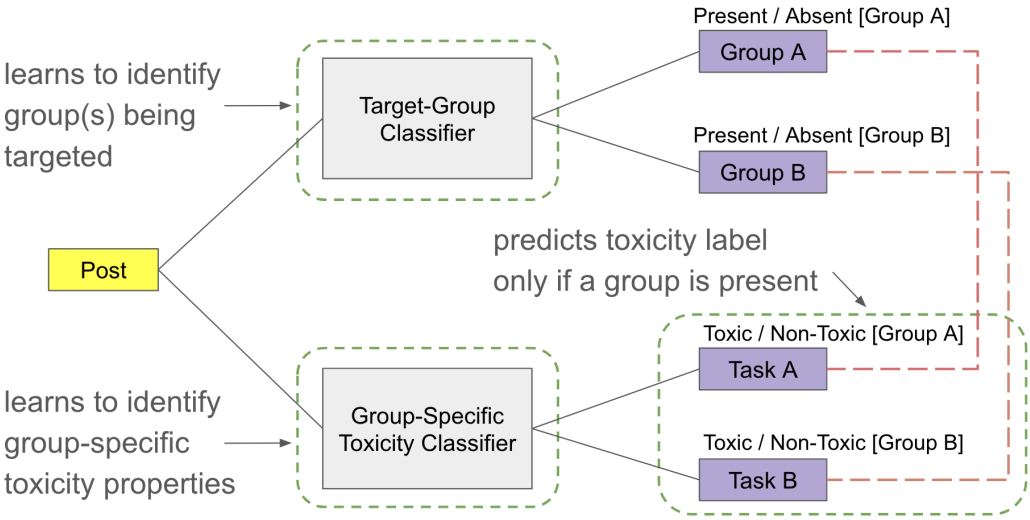


Figure 1.3: Proposed Toxicity Pipeline in our Work. The target-group detection module and the group-conditioned toxicity detection module work in parallel during model deployment, and allows for a better perception of toxicity.

Chapter 2: Problem Setup and Notations

Data: We are given a dataset $\mathcal{D} \in \mathbb{R}^{N \times F}$, with N samples (posts) and F features. These F dimensional features can be extracted using any off-the-shelf NLP model. We also have G demographic groups for each post pertaining to the stakeholder scenario we are considering, *i.e.,* annotators ($\mathcal{A}$) and targets ($\mathcal{T}$) of post. Thus each post can be mapped to an overall (group-agnostic) toxicity label $d \rightarrow y$, as well as multiple group-targeted toxicity labels $d \rightarrow y_g, \forall g \in G$, where the overall label $y = \bigcup_G y_g$ considers the data to be toxic/non-toxic, irrespective of the group. Due to the nature of the G independent groups, we have the combined dataset $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \ldots \cup \mathcal{D}_G$ as the union of the demographic specific data points $\mathcal{D}_g \in \mathbb{R}^{N_g \times F}$.

MTL Problem Setup for Toxicity Detection of Posts: Our objective is to optimize Binary Cross Entropy (BCE), for maximal binary detection accuracy of posts as toxic (1) or non-toxic (0). We can do it over: a) the entire dataset $\mathcal{D}$; or b) target demographic group specific data subsets $\mathcal{D}_g, g \subseteq G$. The Single Task (STL) model $\mathcal{M}_{\text{STL}}$ has independent classifiers for each split $\mathcal{D}_g$, while the Multi Task Learning (MTL) model $\mathcal{M}_{\text{MTL}}$ has one joint classifier with $G+1$ branches with G task-specific branches for each split $\mathcal{D}_g$ and one for overall dataset $\mathcal{D}$. Differential sub-Group Validity (DsGV) [67] arises when the predictive relationship of a data point *w.r.t.* to its label varies as a function of its group, *i.e.,* $\mathcal{F}_g : D \xrightarrow{g} Y$. Therefore we want the model to perform equally across all groups *w.r.t.* a performance measure. Ideally this would mean learning one STL model $\mathcal{M}_{\text{STL}}$ for each group $\mathcal{D}_g$. However, such individual and independent classifiers would lack generalization in term of learning general toxic language properties. Therefore, to address this problem via the model $\mathcal{M}_{\text{MTL}}$, we focus on learning the heterogeniety across groups in each of the task-specific branches, while leveraging the common properties in the shared layers. Thus $\mathcal{M}_{\text{MTL}}$ would learn to achieve high predictive performance across groups addressing DsGV while

also learning general toxicity characteristics as well over the entire population.

MOO Problem Setup for Target Group Detection: Our objective is to optimize Binary Cross Entropy (BCE), for maximal binary detection accuracy of posts target group as present (1) or absent (0). This leads to a mapping of a post d to the groups g that are targeted in it, *i.e.,* $\mathcal{F}_g : D \rightarrow G$. To ensure fairness in target-group detection, given the entire dataset $\mathcal{D}$, we want equal performance across all groups $g \in G$ involved. This notion refers to a Pareto trade-off [109], where we are trading overall detection accuracy over $\mathcal{D}$ *vs.* gaining comparable detection performance across all groups $\mathcal{D}_g$, *i.e.,* trading off between majority rights *vs.* minority protections.

Distinction between Problem 1 *vs.* Problem 2: While for the MTL model (*Problem 1*) we are targeting improved performance across all groups, in the MOO model (*Problem 2*) we are balancing performance of target detection across all groups at the expense of majority/overall performance.

Chapter 3: Conditional Multi Task Learning

In developing natural language processing (NLP) models to detect toxic language [4, 131, 143], we typically assume that toxic language manifests in similar forms across different targeted groups. For example, HateCheck [123] enumerates templatic patterns such as "I hate [GROUP]" that we expect detection models to handle robustly across groups. Moreover, we typically pool data across different demographic targets in model training in order to learn general patterns of linguistic toxicity across diverse demographic targets. However, the nature and form of toxic language used to target different demographic groups can vary quite markedly. Furthermore, an imbalanced distribution of different demographic groups in toxic language datasets risks over-fitting forms of toxic language most relevant to the majority group(s), potentially at the expense of systematically weaker model performance on minority group(s).

This chapter is based on the work: "Same same, but different: Conditional multi-task learning for demographic-specific toxicity detection", Gupta, Lee, De-Arteaga and Lease - published at Web Conf 23.
Online edition: https://dl.acm.org/doi/pdf/10.1145/3543507.3583290

TLDR: Work contributions in this chapter are summarized as follows:

1. We argue the need for group-targeted toxicity detection system compared to a one-size-fits all model, as toxicity manifests differently across groups.
2. We identify an existing flaw in labeling schema [86] that leads to contamination and misinterpretation of group-targeted toxicity labels.
3. We develop a Conditional MTL (CondMTL) framework to model group-targeted toxicity, where each task is identifying toxicity for a specific group.
4. Empirical results show improved prediction performance across target groups *vs.* SoA MTL baselines, with better runtime and memory footprint.

For this reason, a "one-size-fits-all" modeling approach may yield sub-optimal

performance and more specifically raise concerns of algorithmic fairness [4, 110, 129]. At the same time, radically siloing off datasets for each different demographic target group would prevent models from learning broader linguistic patterns of toxicity across different demographic groups targeted. To characterize this phenomenon in which toxic language exhibits both important commonalities and important differences, we borrow the popular phrase of "*same same, but different*" [153].

More formally, such heterogeneity of toxic language targeting different groups can be conceptually framed in terms of *differential subgroup validity* [67]: a relationship $f : X \rightarrow Y$ mapping the input data X to labels Y may have different predictive power across groups. The wide diversity of demographics targeted by toxic language, and the ways in which minority groups may be disproportionately targeted, underscores the importance of understanding and recognizing this phenomenon.

From an algorithmic fairness perspective, it has been shown that excluding sensitive attributes from the features used in prediction, also known as "fairness through unawareness" is ineffective [33]. Some methods, *e.g.,* adversarial fairness approaches [159], address this problem by penalizing models from learning relationships that are predictive of sensitive attributes. Others have noted that making use of such attributes may significantly improve performance for minority groups and reduce algorithmic bias [71, 85], a reason that is tightly linked to the presence of differential subgroup validity. Prior work [23] has shown that differential subgroup validity can be addressed by training models that learn group-specific idiosyncratic patterns, such as decoupled classifiers [34]. In the context of toxic language detection, inclusion of demographics has the potential to boost performance in detecting toxic language targeting the minority group(s) who are less represented in a given dataset.

To address the challenge of differential subgroup validity in toxicity detection, we propose to model demographic-targeted toxic language via multi-task learning (MTL). MTL combines shared and task-specific layers, allowing a model to specialize on relationships relevant to different groups while leveraging shared properties across

groups. In this setting, each MTL *task* corresponds to detecting toxic language targeting a different group. Shared layers can benefit from training across posts targeting multiple groups, while task-specific layers are trained only on posts that target each respective group. For example, if a post targets group-A, it should influence the shared layers and its own task-specific layers, but not task-specific layers for group-B.

3.1 Label Contamination

RQ (1a). How does the nature of task labels in traditional MTL provide barriers to operate under the sparsely labeled group-specific setting, thereby leading to Label Contamination?

In the group-targeted classification setting (be it Hate Speech Classification [86], Toxicity Detection [143], Fake News Identification [81], Media Bias Estimation [137], Misinformation Detection [89] *etc.*), the **labels for each task are often sparse**, meaning that *only a small fraction of the data is labeled for each task*. Traditional MTL (TradMTL) approaches assume that each training point has labels for all tasks, making it difficult to operate under this sparse label setting. To counter this issue, Liu et al. [86] employs a fuzzy rule based schema to identify potential groups of hate targets over examples and update the rule thresholds *w.r.t.* training error. For hate speech, typically the hate class is the smaller class with fewer examples, so they mark all unlabelled examples in their dataset as the larger class, *i.e.,* non-hate, which leads to the issue of *label contamination*.

3.1.1 Who is being targeted?

The set of all possible posts contain: a) Either Toxic (T) or Non-Toxic (NT) posts; b) Targeting neither group; c) Targeting one group ONLY; and d) Targeting BOTH groups. The same scenario can be extended to other stakeholder options *w.r.t.* their group associations. Given this setting, we would observe the effect assigning group-specific labels to different posts in an MTL setup, and analyze the

interpretation of such labeling schema.

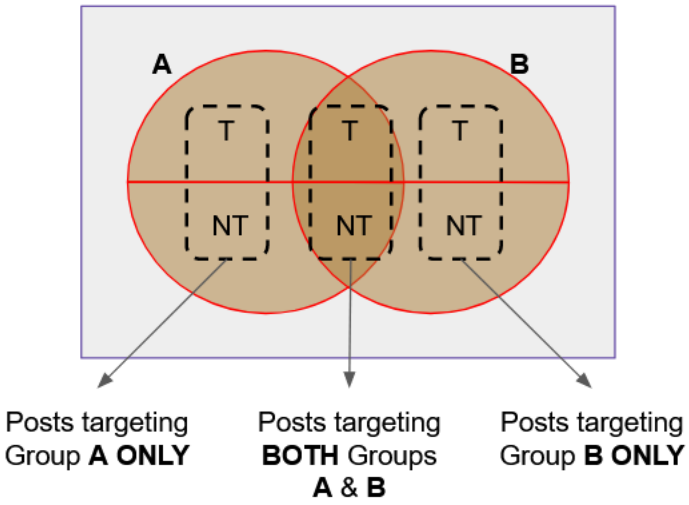


Figure 3.1: Distribution of posts targeting groups. Posts can belong to a) Either Toxic (T) or Non-Toxic (NT) posts; b) Targeting neither group; c) Targeting one group ONLY; and d) Targeting BOTH groups.

3.1.2 Contamination Illustration

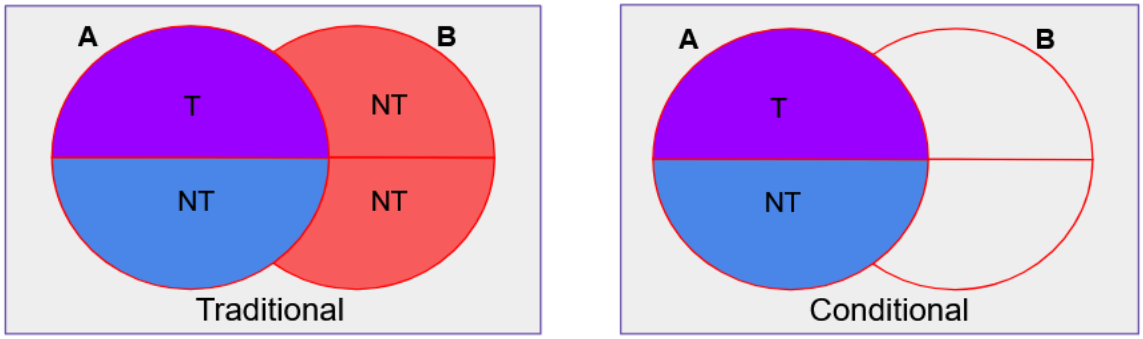
Following the strategy in Liu et al. [86] for **Fig. 3.1**, a post targeting **Group A**, irrespective of toxicity label, is assumed to be non-toxic towards **Group B** as well. This formulation of the task leads to many posts containing toxic language being labeled as non-toxic, by the labels marked red as shown in **Table 3.1**. We argue that this labeling schema, which blends together the questions *is the post toxic?* and *who is the target of the post?*, leads to *label contamination*.

Post	Traditional MTL Labels		Correct Labels	
	Group A	**Group B**	**Group A**	**Group B**
"I hate **Group A**"	Toxic	Non-Toxic	Toxic	•
"I love **Group A**"	Non-Toxic	Non-Toxic	Non-Toxic	•

Table 3.1: Label contamination occurs in a Traditional MTL label assignment when posts that target a given (**Group A**) are assumed to be non-toxic towards any other group (*e.g.*, **Group B**). Red denotes unsupported label assignments, while (•) correctly denotes that these posts do not contain a label *w.r.t.* the target **Group B**.

In order to let the model differentiate between demographic-specific examples,

we consider group-conditional labels from the set {T, NT, •}, where • is an indicator denoting that the label of the current example is irrelevant/unknown *w.r.t.* the group. The Venn diagram in Fig. 3.1 gets updated under the two schema as shown in **Fig. 3.2**. Observe that under the Traditional schema, posts exclusively relevant to **Group B** are forcibly marked as Non-Toxic (NT) for **Group A**. In the proposed Conditional schema we update the labels by marking posts as irrelevant when they are not targeting a particular group (toxic or otherwise).



(a) Labels for TradMTL *w.r.t.* **Group A** (b) Labels for CondMTL *w.r.t.* **Group A**

Figure 3.2: Assigned labels for posts *w.r.t.* **Group A**. In Traditional setting, posts belonging to **Group B ONLY** are marked as Non-Toxic for **Group A**. In Conditional setting, we ignore posts that are *not relevant* to **Group A**. Thereby, posts belonging to **Group B ONLY** are marked as irrelevant for **Group A**.

3.1.3 Proposed Labeling Schema

RQ (1b). Can we propose an updated labeling schema to provide correct group-specific labels to relevant examples?

To illustrate the reasoning for the schema, we show a series of example post templates and their corresponding labels in **Table 3.2**. Note that in the traditional labeling schema, as proposed in Liu et al. [86] and widely followed in the MTL literature, a) any post that is toxic towards a specific group is considered non-toxic towards every other group (see rows 2 and 3); and b) any post that is non-toxic to a group is considered non-toxic towards every other group as well (see rows 5 and 6). Our conditional schema enables each demographic branch of the CondMTL model to

conditionally filter out irrelevant examples (both toxic and non-toxic) for each group.

Hypothetical Post	TradMTL Label		CondMTL Label	
	Group A	**Group B**	**Group A**	**Group B**
"I hate **Group A** & **Group B**"	T	T	T	T
"I hate **Group A**"	T	**NT**	T	•
"I hate **Group B**"	**NT**	T	•	T
"I love **Group A** & **Group B**"	NT	NT	NT	NT
"I love **Group A**"	NT	**NT**	NT	•
"I love **Group B**"	**NT**	NT	•	NT

Table 3.2: CondMTL group-specific labels *vs.* TradMTL labels for some posts. T and NT denote toxic and non-toxic labels, The label (y) denotes the toxicity of a post towards a target group k. The • indicates unknown toxicity wrt. the given group, whereas TradMTL methods erroneously assume such training examples are non-toxic.

When considering the template posts from **Table 3.2**, the TradMTL model with its labeling schema [86] would correctly backpropagate its losses for the all, men, and women branches for the example *I hate* ***Group A*** *&* ***Group B***. Given that this post does target both **Group A** and **Group B** and is toxic, the traditional label (T, T) is equivalent to the conditional label (T, T). However, the template post *I hate* ***Group A*** reveals an issue with the traditional labeling schema and the subsequent information that a TradMTL model would learn; the traditional MTL model would backpropagate a misleading loss for the **Group B** branch due to the **Group B** label in the traditional label (T, NT) being marked as non-toxic (NT). The traditional MTL model would erroneously learn that a post which is toxic towards **Group A** is nontoxic if it were targeted at **Group B**, ultimately confusing the model. In contrast, the CondMTL model avoids backpropagating the loss which may confuse the model by examining the demographic flag corresponding to the label (T, •) and using it to compute the loss only for the **Group A** branch.

3.2 Conditional MTL Framework

RQ (2a). Given the new schema that avoids Label Contamination, can we design an updated MTL framework that accounts for conditional backpropagation on group-relevant examples?

We describe our Conditional MTL [56] function that can operate on our updated labeling schema. The architecture is similar to Traditional MTL, with the change of the loss function that allows the model to incorporate DsGV *w.r.t.* the demographic groups. Our proposed function allows the model to learn common flavors of toxic language in the shared layers, while learning the relevant group-toxicity properties in the task-specific layers of the MTL architecture.

3.2.1 CondMTL Algorithm

The Conditional loss function is a selective variant of the standard weighted Binary Cross Entropy (wBCE). wBCE is a variation of BCE that re-weights the error for the different classes proportional to their inverse label frequency in the data [83]. This strategy is available in popular packages like SkLearn [112] and is useful to address class imbalance (*e.g.,* between toxic *vs.* non-toxic examples).

Algorithm 1 Conditional MTL Loss $(y_{\text{true}}, y_{\text{pred}})$

Input: True Label $y_{\text{true}} = y$ ▷ true label *w.r.t.* current branch
Input: Predicted Label $y_{\text{pred}} = \hat{y}$ ▷ Predicted probability of classifier
Input: Class Weights $w_{\text{toxic}}, w_{\text{non-toxic}}$ ▷ Assigned weights of classes

Select demographic relevant examples in current mini batch
$y^k_{\text{true}}, y^k_{\text{pred}} = \{\}, \{\}$ ▷ Empty lists to hold selected examples
for $i \in n$ **do** ▷ Loop over examples in current mini batch
  if $y \in k$ **then** ▷ current example is relevant to branch k
    $y^k_{\text{true}} = y^k_{\text{true}} \cup y$ ▷ Append current label for consideration
    $y^k_{\text{pred}} = y^k_{\text{pred}} \cup \hat{y}$

Compute weighted BCE loss over relevant selected subset of examples
$err = wBCE(y^k_{\text{true}}, y^k_{\text{pred}}, w_{\text{toxic}}, w_{\text{non-toxic}})$
Output: Error for backpropagation err

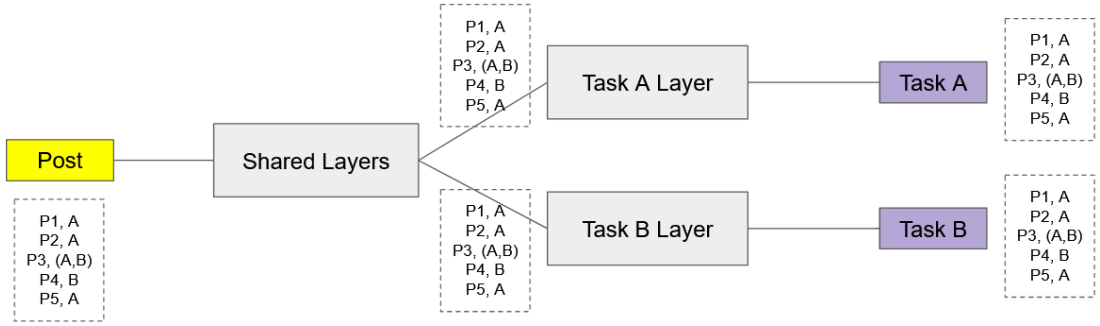
For a given MTL architecture, we can consider $K+1$ tasks: a generic one and

K group-specific ones. We would therefore have three tasks: T_{overall}: given a post, is it toxic?; $T_{\text{Group-A}}$: given a post, is it toxic towards **Group A**? and $T_{\text{Group-A}}$: given a post, is it toxic towards **Group B**? All the examples in the dataset $\mathcal{D}$ are passed through the network, where the T_{overall} branch learns a demographic-independent toxic *vs.* non-toxic representation over N examples. While all N examples and their labels get passed to the demographic-specific branches ($T_{\text{Group-A}}$ and $T_{\text{Group-B}}$) as well, CondMTL only allows backpropagation for relevant instances. For example, only the N_1 examples of $\mathcal{D}_1$ that are targeted towards **Group-A** demographics would be considered by the $T_{\text{Group-A}}$ branch. Similarly, the $T_{\text{Group-B}}$ branch has access to all N examples of D, but only calculates error over N_2 examples of $\mathcal{D}_2$ that are relevant.

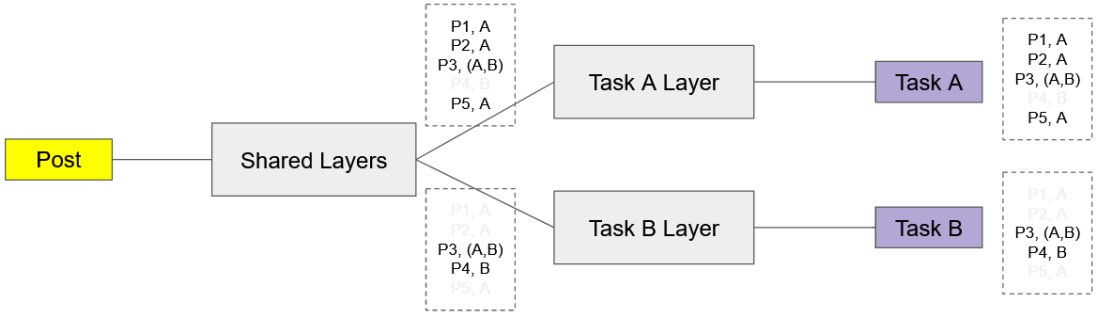
The conditional loss is shown in **Alg. 1**, which operates over each mini batch of examples to compute errors for backpropagation (*steps 5-8*). It accepts two arguments, the true labels ($y_{\text{true}} = y$) and the predicted labels ($y_{\text{pred}} = \hat{y}$). Note that in our CondMTL loss, we are using the conditional label format as shown in Table 3.2, thereby y_{true} is the label conditioned on the demographic flag. Iterating over each example (*step 5*) in the mini batch, we only select relevant instances to that demographic branch based on the demographic flag (k) (*step 6*) and append the true and predicted labels to $y^k_{\text{true}}, y^k_{\text{pred}}$, respectively (*step 7-8*). We also have the weights for each class ($w_{\text{toxic}}, w_{\text{non-toxic}}$), which are pre-computed during label generation. These weights can also be computed over each mini batch on the basis of the number of toxic *vs.* non-toxic examples in the selected subset y^k_{true}. We leave the choice of selecting weights up to the practitioner to account for class imbalance. Finally, we compute the weighted BCE loss on the selected relevant examples for backpropagation (*step 9*). A simple illustration of the working of the loss in shown in **Fig. 3.3**.

3.3 Results

In this section, we present the dataset used, baseline and evaluation measures for comparing the models, along with memory and runtime performances.



(a) Forward pass of CondMTL. All examples are sent to both branches.



(b) Backward pass of CondMTL. Relevant examples are backpropagated in each branch.

Figure 3.3: Forward and Backward passes through the CondMTL framework. All five posts are sent forwarded to both branches. During backpropagation for CondMTL loss, examples relevant to **Group A** are used to loss calculation and alter weights of Task A layers. Similarly, examples relevant to **Group B** are used to loss calculation and alter weights of Task B layers. All five examples influence the shared layer weights, as the linear summation of the two branch losses.

3.3.0.1 Dataset Used

To assess differential subgroup validity in toxic language detection, we focus on toxicity and gender [124, 147]. We use the *Civil Comments* [13] portion of *Wilds* [74]. The dataset has 48,588 training posts labeled as Toxic or non-Toxic. Each post has an explicit annotation for the demographics *i.e.,* gender groups of the target entity, with probability scores about the annotator consensus. We select posts where more than 50% of annotators agreed on the gender of the target. We include only women (W) and men (M) genders, to construct a simplified binary sensitive attribute for our experiments. However, we emphasize that this is a simplification and acknowledge the non-binary nature of gender. Moreover, we note that the reliance on annotators

to identify the gender of the target may contain errors.

	Train Split			Test Split		
Branch	Toxic	Non-Toxic	Total	Toxic	Non-Toxic	Total
All	7,099 (14%)	41,489 (86%)	48,588	3,350 (15%)	19,236 (85%)	22,586
Men (M)	3,940 (15%)	22,499 (85%)	26,439	1,920 (15%)	10,694 (85%)	12,614
Women (W)	4,560 (14%)	28,723 (86%)	33,283	2,068 (14%)	12,964 (86%)	15,032

Table 3.3: Statistics of the Wilds [74] dataset. We consider the binary sensitive target gender as men *vs.* women. The all branch contains all data points, while men and women branches contain the data points in which posts target men or women groups, respectively.

We consider posts where either group (women: 22,149 and men: 15,305) or both groups (both: 11,134) are targeted. **Table 3.3** shows the distribution of targets and labels in the dataset. We use the same procedure for both the train and test splits from the dataset. We observe roughly a 15%-85% split between toxic *vs.* non-toxic labels across all three branches for both the train and test splits.

3.3.0.2 Compared Baselines

For a single task (**STL**) baseline, we use a DistilBERT [128] representation layer to extract numerical features from posts. This is followed by layers of dense neuron connections with *relu* activation and added biases, ending in a classification node with *sigmoid* activation with 0.5 classification threshold (**Fig. 3.4**). For our experiments, we freeze the weights of the DistilBERT [128] representation layer. The only trainable parameters in the models are the dense neuron units that follow the DistilBERT layer until the output branch. One can replace the DistilBERT layer with any other advanced feature representation without altering the rest of the model.

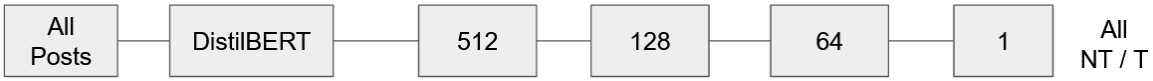


Figure 3.4: Architecture for Single Task, where all posts are passed though a neural network and get classified as toxic *vs.* non-toxic.

Stacked STL model (**Fig. 3.5**) contains independent classifiers for each demographics, distinguishing toxic *vs.* non-toxic. For the Wilds dataset, we construct All, Men,

and Women classifiers resulting in 3× the trainable parameters of one Single task classifier.

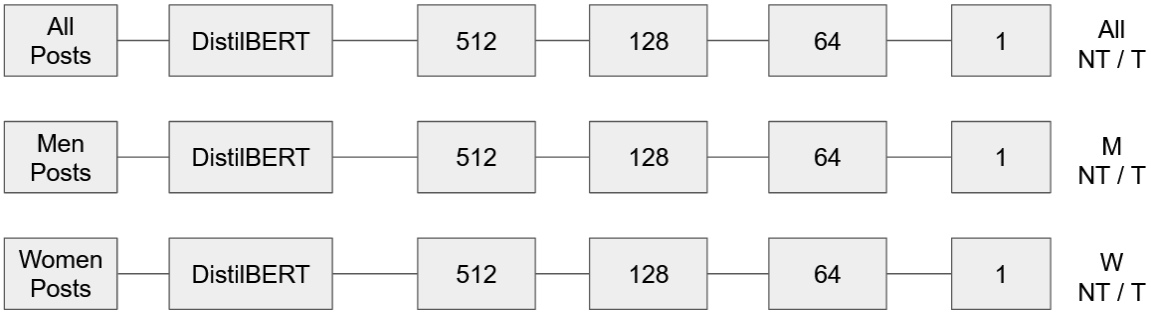


Figure 3.5: Architecture for stacked STL contains three independent single task models, one for each portion of the data.

Traditional Multi Task (TradMTL) model (**Fig. 3.6**) contains a shared layer of 512 dense neurons across all the tasks, while the individual task-specific layers (enclosed in dashed boxes) have dense connections of 128, 64 and 1 each, following the architecture of the STL model. The shared layer is responsible for learning a representation that is common across all tasks, while the task specific layers learn representations specific to their own tasks for differentiating between toxic *vs.* non-toxic posts.

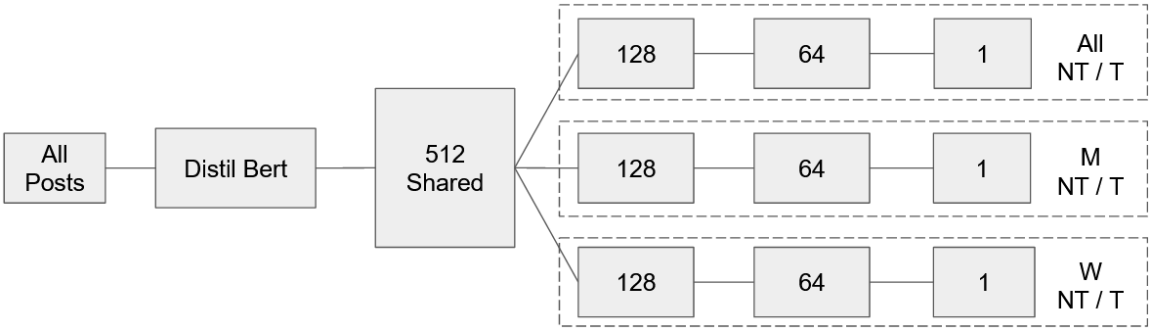


Figure 3.6: Architecture for TradMTL and CondMTL, where the 512 dense neurons are shared across all three tasks while maintaining independent task specific layers (mark by dashed boxes).

Cross Stitch Multi Task (CSMTL) model (**Fig. 3.7**) is similar to the stacked STL model (Fig. 3.5) with Cross Stitch (CS) units [99] placed between each dense

layer. The CS layer is a $K \times K$ weight matrix, initialized as Identity I_K. The intuition is that if the K tasks are independent, then the identity holds even after training with backpropagation. If the tasks are correlated, then the CS matrix at each layer would deviate from identity and learn some common correlation structure across similar tasks. However, both theoretically and empirically, the CS structure does not always improve performance, while taking up more than $K\times$ trainable parameters. We choose this framework for comparison, as it is one of the most widely used ones in the MTL literature.

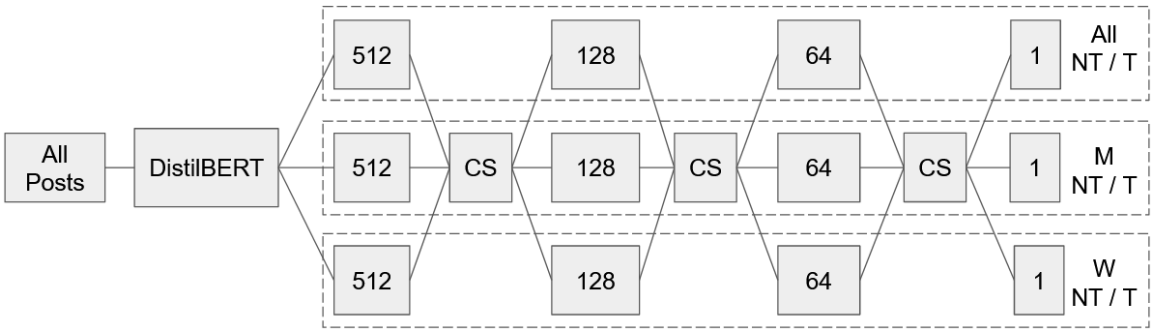


Figure 3.7: Architecture for CSMTL, which is replica of the Stacked STL model, with cross stitch (CS) units between each dense layers, allowing them to share weights across tasks for task similarity.

3.3.0.3 Performance Measures

RQ (2b). Can this proposed framework account for lower group-specific harm compared to other single-task and multi-task baselines, *w.r.t.* evaluation measures for a stakeholder?

We show the performance comparison of the models on the Wilds-Civil Comments [74] test dataset. The Accuracy numbers in **Table 3.4** indicate that all of the models roughly perform the same in terms of overall accuracy *w.r.t.* the Stacked STL ($\sim 86\%$). Since the dataset is imbalanced, with the non-toxic class encompassing 85% of the labels, a model which trivially predicts all testing posts as non-toxic would also achieve a roughly 85% accuracy score. Given that all models perform approximately the same as this trivial baseline, we need to consider other metrics to more holistically evaluate model performance.

Loss type	All	Men	Women
Stacked Single Task	86.3 ± 0.1	85.6 ± 0.2	87.0 ± 0.1
Cross Stitch Multi Task [99]	86.3 ± 0.0	85.8 ± 0.1	86.6 ± 0.0
Traditional Multi Task	86.2 ± 0.2	85.5 ± 0.2	86.7 ± 0.1
Conditional Multi Task (Ours)	86.2 ± 0.2	85.7 ± 0.1	86.7 ± 0.1

Table 3.4: Mean and Standard Deviation measures of models across five runs. All the models perform the same *w.r.t.* Overall Accuracy.

In order to identify the discrepancies between the models, we compare the Recall, F1 and Precision scores in **Table 3.5**. Since the dataset is imbalanced with roughly a 85%-15% split between the non-toxic *vs.* toxic labels, we observe the bias of the classifier towards detection of non-toxic examples in spite of class re-weighting during model training. For the non-toxic (NT) class, all of the post hoc measures are roughly equivalent for the all branch (96% for Recall, 92% for F1, and 88% for Precision) across the compared models. Similar behavior is observed over the men and women branches.

We observe in Table 3.5 that CondMTL achieves better recall values *w.r.t.* baselines over the smaller *i.e.,* toxic class. CondMTL produces recall values of $(29\%, 31\%)$ for the men and women branches respectively, showing marked improvement over CSMTL and TradMTL, which produce recall values of $(13\%, 5\%)$ and $(4\%, 3\%)$, and also outperforming Stacked STL $(24\%, 24\%)$. The superior performance of CondMTL in terms of recall, likely driven by its more accurate understanding of toxicity at a group-specific level, is crucial in the context of automated toxicity detection, where we would like to ensure that toxic posts are not mislabeled as non-toxic (misses), as such errors could disproportionately affect marginalized demographic groups. For instance, women are disproportionately affected by stalking and by sexualized forms of abuse [147].

In terms of precision, CSMTL performs the best $(67\%, 69\%)$ compared to TradMTL $(48\%, 47\%)$ and CondMTL $(56\%, 54\%)$. CSMTL's higher precision number

		All		Men		Women	
		NT	T	NT	T	NT	T
Recall	Stacked STL	96.9	25.2	96.8	23.8	97.1	23.6
	CSMTL	97.2	23.6	98.8	13.3	99.7	4.6
	TradMTL	94.4	20.1	95.8	4.2	95.6	2.8
	CondMTL (Ours)	96.1	29.0	95.9	**28.7**	95.1	**31.2**
F1	Stacked STL	92.3	35.3	92.0	33.5	92.8	33.3
	CSMTL	92.3	33.8	92.2	22.2	92.8	8.7
	TradMTL	92.1	29.8	91.9	7.9	92.6	5.4
	CondMTL (Ours)	92.2	38.3	92.9	**37.9**	93.6	**39.5**
Precision	Stacked STL	88.2	58.6	87.6	56.9	88.9	56.4
	CSMTL	88.0	59.3	86.4	**67.0**	86.8	**69.1**
	TradMTL	87.5	54.4	85.3	47.7	86.6	46.7
	CondMTL (Ours)	88.6	56.1	88.2	55.9	89.7	53.7

Table 3.5: Statistic Comparison between different methods based on internal stats: Recall, F1 and Precision. Numbers are bolded only when they are significantly better than the other models. For a toxic language detection task, *Recall* is of prime importance over the smaller toxic class, since we want to detect as many of the toxic posts as possible in deployment. We observe that CondMTL achieves significantly better recall values for both groups on the toxic labels.

suggests that it is more reserved when predicting a test example to be toxic, which results in less false alarms (*i.e.,* a non-toxic post that is erroneously flagged).

F1 provides a joint view of both precision and recall. In terms of F1, we observe that CondMTL $(38\%, 40\%)$ provides the best results, outperforming CSMTL $(22\%, 9\%)$, TradMTL $(8\%, 5\%)$ and Stacked STL $(34\%, 33\%)$. This is because CondMTL's recall values are a scale apart compared to the other models.

Although our CondMTL model is not optimized over any strict differentiable measure of fairness, we post hoc observe that it has a low false negative error rate balance *i.e.,* improved equal opportunity [59]. Mathematically, a classifier with equal false negative rate (FNR) will also have equal true positive rate (TPR) or recall.

We have shown that CondMTL achieves much better recall values compared to other MTL variants. **Table 3.6** shows the post-hoc measured Equal Opportunity (EO) gap across both groups for the models. All models except for CSMTL (9.0) produce low EO gaps. Although having a lower EO gap value is ideal, it is necessary to evaluate the EO gap values of the different models with the context of their recall values. Thus, while TradMTL has the lowest EO gap value (1.4) among the MTL variants, given its poor recall values this model is unlikely to be desirable in practice, whereas CondMTL produces a low EO gap of 2.5 while maintaining higher recall values.

Model	Recall (Men)	Recall (Women)	EO Gap
Stacked STL	23.8	23.6	0.2
CSMTL	13.6	4.6	9.0
TradMTL	4.2	2.8	1.4
CondMTL (Ours)	28.7	31.2	2.5

Table 3.6: Recall per group and Equal Opportunity (EO) gap measured as the absolute difference between recall values over the groups. For recall, higher values are better. For EO, lower values are better.

Comparing the confusion matrices of the three branches of the MTL models in **Fig. 3.8** reveals that CondMTL performs better in the demographic group branches (men and women) for the smaller toxic class. All models perform fairly well when classifying the nontoxic examples *i.e.,* the cyan sections. Non-toxic posts that are erroneously flagged as toxic (*i.e.,* false alarms) are shown in the blue sections. On the other hand, TradMTL and CSMTL both struggle to correctly identify toxic examples and instead classify a greater portion of the toxic test examples as nontoxic (*i.e.,* misses). These misses correspond to the orange sections of the confusion matrices. Comparing the red sections of the model confusion matrices reveals that CondMTL correctly classifies a greater proportion of the smaller toxic class. Given that non-toxic language is more common, CondMTL's ability to capture a greater proportion of the toxic posts would be valuable in a deployed toxicity detection model.

3.3.1 Architecture and Runtime

RQ (2c). Can the proposed pipeline design address memory efficiency and model runtime concerns *w.r.t.* SoA baselines?

Table 3.7 shows trainable parameters for the baseline models and for CondMTL.

Model type	# Params	Δ	Time(s)	Δ
Stacked STL (3 models)	1,403,139	-	7,200	-
CSMTL [99]	1,403,166	+0%	2,600	-64%
TradMTL	615,683	-56%	2,200	-69%
CondMTL (Ours)	615,683	-56%	2,050	-72%

Table 3.7: Space (parameter size) and training time (seconds for 10 epochs) required by STL *vs.* MTL models on the Wilds dataset. The DistilBERT representation is frozen and the dense layers are trainable, with each STL model having 467,713 trainable parameters. For the 3 tasks considered, we assume 3 different STL models and report space and time summed over all 3. We then report % space and time reduction achieved by MTL models *vs.* this baseline of 3 STL models.

The single task model (Fig. 3.4) has 467,713 trainable parameters, hence the stacked STL (Fig. 3.5) operating on the All, Men, and Women portions of the data has $3\times$ or $1,403,139$ trainable parameters. We report space reduction achieved by MTL models *vs.* this reference of 3 STL models. The TradMTL and CondMTL models (Fig. 3.6) have the same architecture but different labeling schema and loss functions. They have a shared 512 unit layer representation and three task specific branches which collectively have 56% fewer trainable parameters when compared to the stacked STL model. The CSMTL model (Fig. 3.7) is a replica of the Stacked Single Task model with cross stitch (CS) units between each of the dense layers. It has 27 ($\sim$ +0%) more trainable parameters when compared to the Stacked STL model due to the extra connections from the CS units. In terms of training and further deployment, the traditional and conditional MTL models are preferable due to significantly reduced model size even when dealing with multiple tasks (three in this case). One can observe that the trainable parameters in Cross Stitch networks

scale linearly *w.r.t.* number of tasks, causing memory stagnation. This issue has been raised and studied in [138].

We report the training runtime in Table 3.7 *w.r.t.* 10 epochs. Both TradMTL and CondMTL models have the same number of trainable parameters. However the CondMTL model only trains over a subset of the data in its men and women branches, which reduces runtime. Empirically, we observe a reduction of 72% in CondMTL *vs.* 69% in TradMTL. The stacked STL and CSMTL models take longer to train, since they roughly have the same number of trainable parameters. However, the CSMTL model operates on the three branches in a single model rather than three independent models, resulting in a lower GPU pipeline load and a 64% reduction in time.

3.3.2 Analysis of Conditional MTL

We make two remarks based on the theoretical working and empirical analysis of the CondMTL and CSMTL networks. Furthermore, we verify our stated propositions for CondMTL and CSMTL through simple and verifiable benchmarking cases.

Remark. Our proposed Conditional MTL does not allow contamination of weights across shared task layers and learns only over the group specific distribution for each demographic branch.

The CondMTL architecture (Fig. 3.6) is an exact copy of TradMTL with the distinction of the updated loss function and labeling schema. Since the task specific layers (indicated by dashed boxes) do not interact with each other, each loss function is strictly guided by the examples that are relevant to its own branch. Assuming that the data distribution *w.r.t.* two groups $\mathcal{D}_1$ and $\mathcal{D}_2$ are independent of each other, each branch learns a representation of their own dataset and does not take into account group irrelevant examples. The CondMTL loss (Alg. 1) computes the loss over each

group-specific distribution (Eq. 3.1), thereby avoiding label contamination.

$$
\begin{aligned}
err_{all} &= wBCE([y_{true}]_{\mathcal{D}}, [y_{pred}]_{\mathcal{D}}) \\
err_{men} &= wBCE([y_{true}]_{\mathcal{D}_1}, [y_{pred}]_{\mathcal{D}_1}) \\
err_{women} &= wBCE([y_{true}]_{\mathcal{D}_2}, [y_{pred}]_{\mathcal{D}_2})
\end{aligned} \tag{3.1}
$$

Remark. Cross Stitch MTL [99] allows contamination of weights across shared task layers.

The CS unit (Fig. 3.7) is initialized with an identity structure, where the number of tasks dictates the size of the matrix. For illustration, let us consider two tasks for $CS \in I_2$. We find the following two flaws *w.r.t.* the logic of CS units: a) if the two tasks are truly independent, then the CS unit should not deviate from identity; and b) even when two tasks are correlated, allowing deviation from identity, the CS unit should still be a symmetric matrix since two tasks talking to each other are symmetrically equivalent. However, such constraints are not present in the implementation of the CS units, which causes them to learn arbitrary weights during model training. The weights become cross-contaminated across tasks.

To verify illustration 2, we also show the final weights of the CS units in our Wilds dataset training. One can observe that the symmetric property of CS units is violated. Note that due to the *same same but different* nature of group-targeted toxicity, they share some commonality *i.e.,* they are not fully independent of each other, which would cause the CS matrix to deviate from identity. However, since tasks talking amongst each other should be symmetrical in nature, we would expect the updated CS matrix to hold the symmetric property. The values reported in Eq. 3.2 are *w.r.t.* Fig. 3.7 where we have three CS matrices. These three CS matrices show clear deviation from symmetry in the off-diagonal elements.

$$
\begin{aligned}
CS_1 &= \begin{bmatrix} 1.00 & -3.69e-3 & -1.46e-4 \\ 1.53e-3 & 1.00 & 2.53e-3 \\ -4.28e-3 & -1.04e-3 & 1.01 \end{bmatrix} \\
CS_2 &= \begin{bmatrix} 1.00 & 2.49e-2 & -2.29e-2 \\ 1.05e-2 & 1.01 & 2.99e-3 \\ -1.03e-2 & 4.74e-3 & 1.01 \end{bmatrix} \\
CS_3 &= \begin{bmatrix} 1.00 & 1.29e-2 & 5.13e-2 \\ 3.53e-2 & 1.01 & 2.19e-2 \\ 7.74e-5 & 9.51e-3 & 1.01 \end{bmatrix}
\end{aligned} \tag{3.2}
$$

3.4 Discussion and Future Work

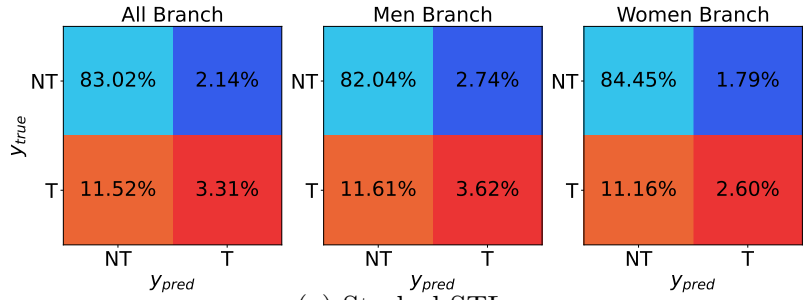
Effect of Label Contamination. As a result of the misleading labeling schema discussed in Section 3.1, the TradMTL and CSMTL models learn to mostly label examples as non-toxic (NT). When considering an example post *I hate men*, we know that this post is both toxic and directed at men only; however, had this example post been in the training set, it would have erroneously taught the women branch of the baseline MTL models that the post was non-toxic. Similarly, an example post *I hate women*, which is toxic and targeted at women only, would have contaminated the weights of the men branch of the baseline MTL models by skewing it to make more non-toxic predictions. While this weight-skewing effect of label contamination may result in higher accuracy scores for TradMTL and CSMTL because the majority of the Wilds dataset is nontoxic (85% of the dataset is non-toxic), these models will subsequently acquire a poor understanding of toxicity. Conversely, CondMTL ensures that the demographic group branches learn a more accurate understanding of toxicity and correctly labels more toxic posts as toxic, as illustrated by higher group-specific recall values on the toxic posts in the testing dataset.

Measures of algorithmic fairness and their usage. Frequently, models that seek to improve algorithmic fairness do so by directly considering a fairness measure as part of the loss function, which is often done in the form of a penalty term. In our optimization objective, we do not incorporate any algorithmic fairness measure. We use a variant of weighted Binary Cross Entropy (wBCE) for optimizing the MTL

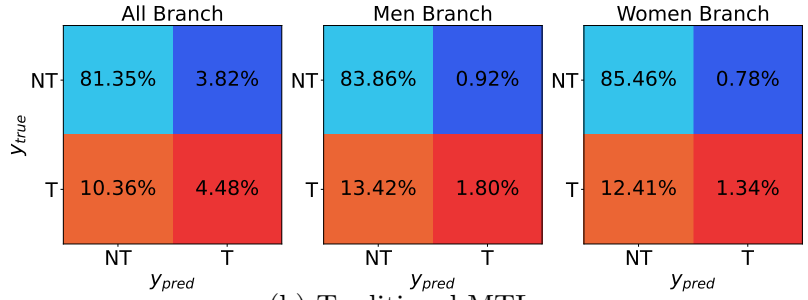
model branches which correlates to giving higher priority in detecting examples from the smaller toxic class. Rather than modifying the loss function as a result of our fairness concerns, we modify the network architecture and labeling schema in a way that enables us to better capture heterogeneity across groups. This approach is suitable for settings in which improving recall for the minority group is a primary fairness consideration. However, if the primary concern is the *difference* across groups, the proposed approach may not always yield improvements, because even if recall improves for both groups, the improvement could be greater for the majority class. In such a scenario, we would like to optimize the network *w.r.t.* a fairness measure, and thus we need to use a differentiable version of that said fairness measure. There exist works in the literature [133] that can optimize a network for a fairness measure. Correspondingly, networks can also be optimized for equal accuracy across groups [57] or equalized odds [120]. The choice of the measure depends on the practitioner's need and the availability of a differentiable version of said measure.

When considering intersectional fairness, *e.g.,* the intersectionality of gender and race, or a more fine-grained grouping of demographics, the dimensions of groups increase. In these cases, the performance of decoupled approaches drops due to data sparsity. We anticipate that the benefits of the shared layer in CondMTL would be even more salient in this setting.

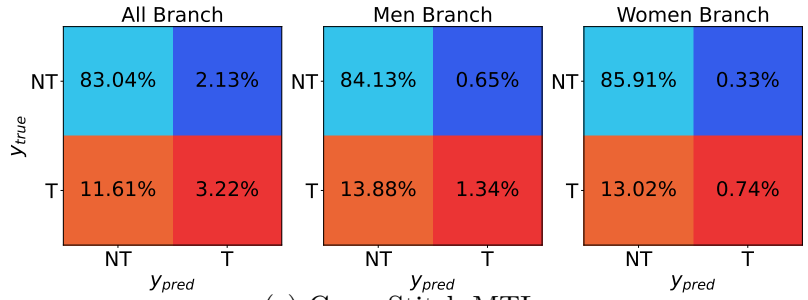
Other stakeholders in toxicity detection. When considering toxicity detection, there are multiple stakeholders who are involved. We have primarily focused on the subject of the post, but other stakeholders include the author of the post and the annotator. Previous work has shown risks of algorithmic bias affecting authors of posts; for instance [129] shows that models may exhibit disporportionately high false positive rates for posts written in African American English. The importance of considering the demographics of annotators involved in labeling data has also been recently emphasized [126]. Using the proposed CondMTL to model the problem in relation to other stakeholders' demographics is a natural extension of this work.



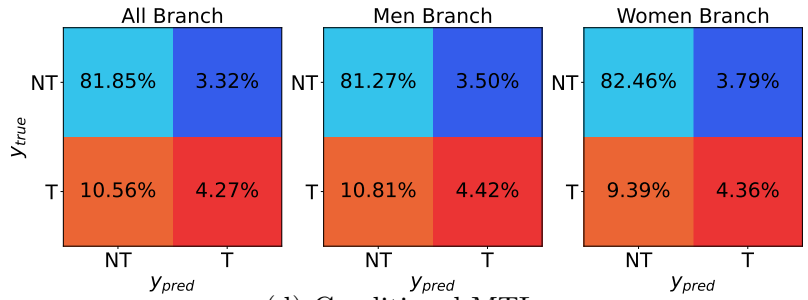
(a) Stacked STL.



(b) Traditional MTL.



(c) Cross Stitch MTL.



(d) Conditional MTL.

Figure 3.8: Confusion matrices of 3 tasks (columns) for different models (rows). Values are shown as percentages in each block *w.r.t.* the number of instances relevant to that branch. CondMTL performs significantly better in the demographic specific *men* and *women* branches, due to training over group relevant examples only.

Chapter 4: Stakeholder-Aware Joint MTL Model

This chapter builds on the stakeholder and architectural modeling as described in **Chapter 3**. While CondMTL only takes into account the target-group of posts, the proposed framework SAJ-MTL (Stakeholder-Aware Joint MTL) framework, in this chapter, builds upon the foundations laid out in the CondMTL chapter by expanding its capabilities to account for multiple interacting stakeholder groups - specifically annotators and target groups - in toxicity detection tasks. By incorporating a joint consideration of annotator and target perspectives, SAJ-MTL ensures that the model can better capture subtle distinctions in annotator's perceptions of toxicity. This extension allows for improved fairness and predictive performance across all groups, addressing the limitations of CondMTL's single-stakeholder focus, and making SAJ-MTL a more comprehensive solution for stakeholder-aware toxicity detection.

TLDR: Work contributions in this chapter are summarized as follows:

1. We propose an extended CondMTL framework — Stakeholder-Aware Joint (SAJ)-MTL — to jointly model the interaction of stakeholder identities, *i.e.,* target group and annotator group demographics.
2. We account for diagreements between annotators in our model training protocols via two variants: Joint-Inter (only inter-group disagreements) and Joint-Intra (both inter-group and intra-group disagreements).
3. We develop a scalable version of the joint model to be memory efficient as the number of stakeholder groups increases, without loss in performance.
4. Results show improved predictive performance of the SAJ-MTL model *w.r.t.* two SoA baselines, while being runtime and memory efficient.

As previously discussed in **Chapter 3**, toxicity detection models can sometimes struggle to distinguish between genuinely harmful content and content that uses certain words or phrases in a non-harmful way [74], depending on the target demographics. By considering the target group, the toxicity models [56, 86] can de-

velop a more nuanced understanding of how cultural variations in language are used [39], which helps protect these groups from hate speech and discrimination. This approach promotes equal participation and representation for all demographic groups. The issue of toxicity perception is further compounded when accounting for the demographic information of the community viewing it [50].

Supervised machine learning models need ground truth labels. Computer vision tasks like object identification, localization *etc.* have attained success due the simplicity of the task *w.r.t.* human visual acuity, and the objective nature of the entity to be recognized. However, the same argument cannot be made for toxicity detection models, wherein the nature of the language itself is complex and the labels are often subjective in nature. The background of the annotators [1, 6, 125] has an important impact on the labels. Furthermore, the diverse pool of annotators might mean more diverse labels for a single post by capturing a broader variety of toxicity perception. Annotators may exhibit observable differences in their labeling patterns when grouped by their self-reported demographic identities, such as race, gender, *etc.* These patterns are termed as annotator identity sensitivities, referring to an annotator's increased likelihood of assigning a particular label on a data sample, conditional on a self-reported identity group [126] or one inferred from platform metadata [30]. This interaction between the target and annotator demographics to infer the final toxicity label for a post forms the basis for our joint MTL framework.

4.1 Stakeholder-Aware Joint (SAJ) MTL

In this section, we explain the need for our Stakeholder-Aware Joint (SAJ) MTL model, describe the data setup, and outline the framework. We also show the workflow of how this model captures the toxicity of a post as a joint interaction between the annotator and the target demographics.

4.1.1 Motivation

The Jury Learning work [50] by Gordon *et al.* provides a critical motivation to jointly model both the target(s) and the annotator(s) demographics of a post in toxic language detection. They argue that annotator subjectivity plays a significant role in toxicity detection, where biases, personal backgrounds, and interpretations of content vary across annotators. It highlights the need to move beyond treating annotator labels as objective truths, instead leveraging the diversity of annotators' perspectives by modeling them explicitly. This insight is particularly relevant in the context of toxicity detection, where judgments about harmful language are often influenced by both who the content is directed at (target demographic) and who is evaluating it (annotator demographic). By integrating both annotator and target identities, their recommendation model can better capture the inherent subjectivity in toxicity labels and reduce bias in detecting toxic language, which is crucial for improving fairness and accuracy across diverse demographic groups. There have been other studies [10, 26, 68, 127, 148] showing the same effect on data collection around toxicity with varied set of annotators. Building on this concept, our work expands the Jury Learning paradigm by using a MTL framework that explicitly incorporates both annotator and target demographic information, addressing the limitations of models that focus solely on target identities.

4.1.2 Problem Statement and Data Setup

RQ (3a). How can we update the architectural pipeline of CondMTL to learn a joint model tailored for specific stakeholder (annotator - target) interactions?

In the current CondMTL framework (Chapter 3), we only considered one stakeholder at a time, *i.e.,* the target. Therefore, the label associated with each post is group conditioned on that stakeholder, *i.e.,* $y = Pr(d|\mathcal{H})$, where the stakeholder

$\mathcal{H}$ is the Target ($\mathcal{H}_{\text{targ}}$) or possibly the Annotator ($\mathcal{H}_{\text{anno}}$).

$$y_{\mathcal{H}_{\text{targ}}} = Pr(d|\mathcal{H}_{\text{targ}}) \qquad \text{Target driven} \tag{4.1}$$

$$y_{\mathcal{H}_{\text{targ}}} = Pr(d|\mathcal{H}_{\text{anno}}) \qquad \text{Annotator driven} \tag{4.2}$$

In our updated content moderation setting, the toxicity of a post is associated not only with the target of the post, but also with the interaction between the target identity and the annotator viewing it.

$$y_{\mathcal{H}_{\text{targ}}} = Pr(d|\mathcal{H}_{\text{anno}}, \mathcal{H}_{\text{targ}}) \qquad \text{Joint Annotator and Target driven} \tag{4.3}$$

Regarding setting up the data and labels for the stakeholder-aware joint framework, we update our labeling schema as per **Eq. 4.3**. Consider a post d, and two groups *Black* and *Latinx* for both annotator and target groups. The schema is then having a annotator-target guide conditional label tuple as $[A_{group}, T_{group}, Label_{cond}]$. For example, if the post only targets the *Black* group and is labeled by two annotators each from both groups, then our schema looks as follows:

$$\begin{aligned} d \rightarrow & [[A_{Black}, T_{Black}, Toxic], [A_{Black}, T_{Black}, Toxic], \\ & [A_{Latinx}, T_{Black}, non-Toxic], [A_{Latinx}, T_{Black}, non-Toxic]] \end{aligned} \tag{4.4}$$

The interpretation of the schema states that the single post d was labeled by four annotators. Both the *Black* annotators marked the post d as being towards the *Black* group, whereas both the *Latinx* annotators marked the post d as being towards the *Black* group. Thus, the label of each post instance is jointly conditioned on the target and annotator demographics, capturing a better view of perceived toxicity.

4.1.3 Framework for Target-Community Interaction

A solution to our joint model is to extend our CondMTL framework (Chapter 3) to take combinations of stakeholder groups and spawn branches for each of them as shown in **Fig. 4.1**. For example, given two annotator ($a = 2$) groups $A1, A2$ and

three target ($t = 3$) groups $T1, T2, T3$, we split out $a \times t = 2 \times 3 = 6$ branches corresponding to all permutes of (Ai, Tj). Being an extension of CondMTL, along with the updated stakeholder-guided labeling schema, we ensure the following: a) we avoid any label contamination issues; b) SAJ-MTL would perform better in terms of model evaluation compared to other Single Task and MTL variants; and c) all the correctness checks of CondMTL translates directly to SAJ-MTL.

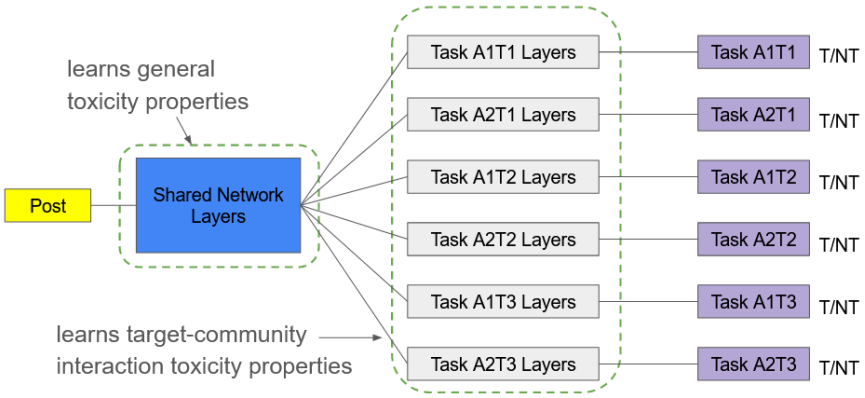


Figure 4.1: SAJ-MTL Architecture Pipeline. Each task specific layer corresponds to a specific combination of annotator target group tuple (A_i, T_j) to treat different demographic interactions independent. While the shared layer still learns general patterns of toxicity that is naturally prevalent, the task specific layers picks up on content that is more finely targeted and demographically contextual.

This SAJ-MTL framework is an end-to-end joint model trained on a single dataset. The proposed model is easy to visualize and conceptually understand, being an extension of the CondMTL framework, where each task-specific branch is a combination of an annotator and target group. Also being an extension of CondMTL, this framework can deal with sparse labeling scenarios of posts. The computational complexity of this model is $\mathcal{O}(at)$, as the number of branches increase multiplicatively *w.r.t.* stakeholder group cardinality, $a \rightarrow$ #annotator groups, $t \rightarrow$ #target groups.

4.2 Accounting for varied Annotator perspectives

From recent works [50, 68, 127, 148], it is well established that annotators often disagree, particularly on subjective tasks, with notable variability both across and within demographic groups. In this section, we discuss the training and evaluation protocols designed to incorporate *inter(across)-Group* and *intra(within)-Group* disagreements of annotators into our SAJ-MTL framework. These protocols ensure that our framework accounts for the diverse perspectives and disagreements that arise during the annotation process, leading to a more robust and equitable model.

RQ (3b). Can we improve model performance and fairness by taking into account annotator disagreements both at the inter-group and intra-group level to better reflect the perception of toxicity?

4.2.1 Joint-Inter Model

Inter-Group Disagreements refer to the variations in how different demographic groups perceive and label toxic content. In the context of toxicity labeling, annotators from different demographic backgrounds - such as race, gender, age, or cultural context — often interpret and judge content through diverse lenses, leading to significant disparities in labeling decisions. For example, certain phrases or expressions may be considered offensive or harmful to one group while being deemed benign or neutral to another. These differences arise due to varying lived experiences, societal norms, and historical contexts that shape each group's sensitivity to specific types of toxic language content. In toxicity detection models, failing to account for these inter-group differences can result in biased outcomes, where the model may disproportionately penalize or overlook content that is offensive to certain demographic groups.

Our SAJ-MTL model by its architecture (Fig. 4.1) accounts for *Inter-Group* disagreements of annotator opinions. The presence of task-specific layers corresponding to specific annotator-target tuple (A_i, T_j) allows this flexibility. In reference to **Eq. 4.4**, the model redirects the first two instances of the post d to the $(\mathbf{A_{Black}} :: \mathbf{T_{Black}})$

branch as $[\mathbf{A_{Black}}, \mathbf{T_{Black}}, \mathbf{Label_{cond}}]$. Similarly, the last two instances of d are passed to the $(\mathbf{A_{Latinx}} :: \mathbf{T_{Black}})$ branch. Once the posts are redirected accordingly, we can take a majority voted label for multiple instances of post d in a specific branch to model how a specific annotator group as a whole perceives toxicity directed towards a specific target group. This protocol allows building models that provide fair and balanced assessments of toxicity, ensuring that no group is unfairly impacted by the model's predictions, by the opinion of the dominant group(s).

4.2.2 Joint-Intra Model

Intra-Group Disagreements refer to the divergences in labeling decisions that occur within a single demographic group when annotators assess toxic content. Even among individuals who share similar backgrounds, subjective interpretations of language can vary marginally. This is particularly evident in toxicity labeling, where personal experiences, individual tolerance levels, and differing social perspectives lead to inconsistent judgments. For example, within a single demographic group, some annotators may perceive a certain comment as offensive, while others may view it as acceptable or non-toxic. These internal disagreements highlight the complexity of labeling subjective content, and failing to account for these intra-group disagreements can introduce ambiguity and reduce the reliability of model predictions.

Given the premise in **Eq. 4.4** of routing examples to resolve *Inter-Group* disagreements of annotator opinions, we provide an additional step in our model training protocol to handle *intra-group* disagreements as well. With the two examples of post d, which were routed to the $(\mathbf{A_{Black}} :: \mathbf{T_{Black}})$ branch as $[\mathbf{A_{Black}}, \mathbf{T_{Black}}, \mathbf{Label_{cond}}]$, we keep the two instances separate instead of majority voting their labels. This allows the SAJ-MTL to distinguish on the fact that even when the same post d is being used to train a branch, its label might not always be the same, rather represents variying personal toxicity perception within (intra) an annotator group. Same holds for the other two instances of d in the $(\mathbf{A_{Latinx}} :: \mathbf{T_{Black}})$ branch, where they are treated as independent instances as well, with their respective labels.

4.3 Scalable Extension of the Joint Stakeholder Model

RQ (3c). How can existing multi-task learning architectures be extended to remain scalable and computationally efficient as the number of task branches (demographic group-pair) increases?

From a computational perspective, the *naive* SAJ-MTL framework design would cause combinatorial explosion when the number of stakeholder groups increase due to the multiplicative complexity of $\mathcal{O}(at)$. All other MTL baselines also suffer from the same scaling issue leading to memory overload in GPU when training the models, an issue widely discussed in the Vision community for MTL [46, 60, 138]. Therefore, we also update the architecture to have linear scaling, *i.e.*, $\mathcal{O}(a+t)$, while at the same time designing numerical correctness checks to validate the workings of this updated *scalable framework* to achieve predictive performance equal to that of the naive model. This ensures that we gain scaling at no cost to predictive performance.

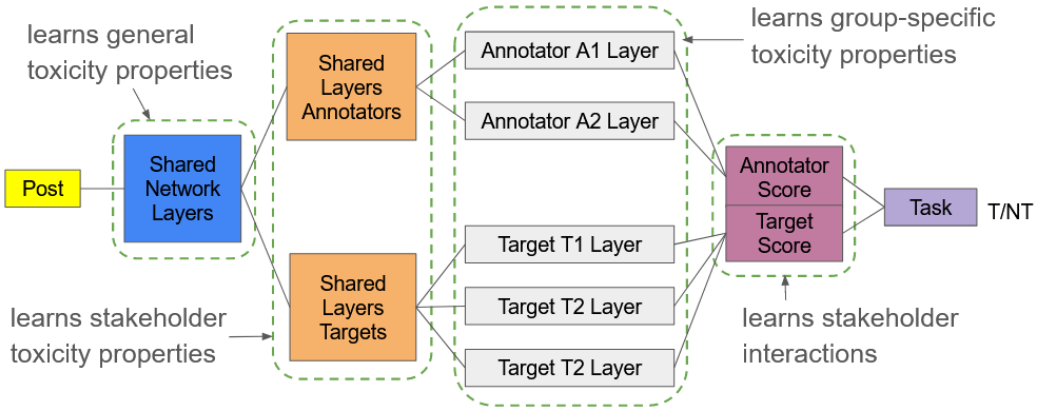


Figure 4.2: Scalable architectural pipeline of SAJ-MTL. Each task specific layer corresponds to a specific stakeholder demographic group. The shared layer still learns general patterns of toxicity that is naturally prevalent. The interaction of the task specific layers in the magenta box picks up on content that is more finely targeted and demographically contextual for (A_i, T_j) pairs.

The multiplicative nature of the *naive framework* arises from making every possible pairs of target and annotator groups and branching off a task-specific layer for each one. In the updated scalable model, as shown in **Fig. 4.2**, we reduce this

complexity to an additive nature to account for model scaling when multiple groups across different stakeholder identities are involved. We have a global shared layer (leftmost) for learning general patterns of toxic language. This layer splits into separate shared layers accounting for each stakeholder identity (annotators and targets), following which we have group-specific layers for each stakeholder group. Therefore, for our previous example having two annotator ($a = 2$) groups $A1, A2$ and three target ($t = 3$) groups $T1, T2, T3$, we split out at least $a + t = 2 + 3 = 5$ branches corresponding to every stakeholder groups, plus intermediate ones. This modification results in linear additive complexity of $\mathcal{O}(a + t)$.

We have an additional **concatenation** gate (magenta) layer of the annotator and target groups, followed by the final softmax layer to provide the overall toxicity score of the post corresponding to all target-annotator pairs (A_i, T_j) that are available for a post. The gates interact depending on the label of each post d as [annotator-group, target-group, conditional-Label], where a label is only allowed to conditionally backpropagate and train the model, *i.e.,* update it's task specific representation weights, **if and only if** it matches the specific annotator-group::target-group flags.

We achieve *equal predictive performance w.r.t. naive baseline*, while resolving the scaling through architectural improvement. We perform numerical correctness checks to achieve scaling without any performance trade-off. This **Scalable SAJ-MTL** model is still a MTL framework, having shared and group-specific layers, though it looks different from a traditional MTL due to the design choices we made to incorporate stakeholder cardinality efficiency. One can extract specific annotator-target pair scores from the model, preserving the multi-headed output logic of MTL.

4.4 Results

In this section, we present the dataset and baseline models for evaluating the predictive performance of SAJ-MTL, along with scalability tests.

4.4.1 Dataset

For experimentation, we use the Hugging Face dataset [37], originally open-sourced in [70] as the MHS corpus that is derived from the DLab dataset [126]. For each post, we have information of the inferred target-demographics, along with self reported demographics of each annotator. The label for each post is binary (Toxic *vs.* non-Toxic) as perceived by the specific annotator marking that post. Although the dataset has seven demographic groups as target, we sub-sample only four demographic groups - Asian, Black, Latinx and White, and ignore three others (Middle-Eastern, Native-American and Pacific-Islander). We do this pre-processing, since the number of samples with relevant annotator-target pairs (A_i, T_j) for the three excluded groups were too small to make any feasible train-test split and draw any meaningful conclusions. Thus, both our targets and annotators belong to mentioned four sample rich groups. Refer to **Table 4.1** for sample size. As there are posts marked by multiple annotators, we pre-process the dataset to ensure that train-test (80% - 20%) split contains distinct posts. Within each split, a post might be annotated multiple times.

Target Groups	Annotator Groups			
	Asian	Black	Latinx	White
Asian	621	721	551	5501
Black	1560	2292	1654	18545
Latinx	638	961	700	6657
White	711	1086	706	7825

Table 4.1: HuggingFace Dlab dataset statistics showing the number of posts that match a specific annotator-target pair (A_i, T_j) over four demographic groups.

4.4.2 Text Augmentation based Approach - SoA Baseline

Following Jury Learning [50], the work by Flesig *et al.* [38] is the first work we are aware of to jointly model the interaction between the target and the annotator groups through a text-augmentation-based approach. Their model relies on additional pieces of information beyond just the post and demographic information itself *i.e.*,

annotator IDs and survey responses. This information is used to update the post itself with additional markers via CLS tokens and a standard loss to learn the conditional labels. The classification layers are pre-trained and ported over from other datasets which can lead to label contamination and wrong distributional inference. For our purposes, we train all parts of the model on one training data and no additional annotator IDs and survey responses as part of text augmentation for the input post.

The other baseline we choose is the Target-only CondMTL (Chapter 3) framework, where we focus on toxicity label directed at a target group by combining labels across all demographics of the annotator.

4.4.3 Performance

RQ (3d). Can a model better capture this joint annotator-target interaction via group-conditioned losses rather than focusing on text-augmentation based approaches?

Table 4.2 present summary statistics of Micro and Macro average performance metric across different models, for F1, Precision and Recall, where higher scores indicate better performance. Our proposed SAJ-MTL Joint-Intra variant emerges as the top-performing model, largely due to its ability to capture both *inter-group* and *intra-group* disagreements among annotators. By taking into account the varying opinions within demographic groups, this variant is able to model subtle distinctions in how toxicity is perceived, resulting in more accurate predictions. This highlights the sensitivity of model learning to intra-group diversity in judgments. In contrast, the Joint-Inter variant, which employs majority voting within groups, underperforms slightly. While it effectively captures *inter-group* disagreements across demographics, it struggles with finer distinctions within groups because majority voting tends to suppress differing perspectives. This often leads to a loss of critical nuances that can significantly impact toxicity labeling, particularly in edge cases where certain expressions may be contentious within a group.

Metric	Level	Joint-Intra	Joint-Inter	Flesig [38]	Target-Only (CondMTL)
F1	Micro	**0.6390**	0.6115	0.5240	0.5000
	Macro	**0.6529**	0.6432	0.5470	0.4879
Precision	Micro	**0.7063**	0.6686	0.6345	0.6041
	Macro	**0.7487**	0.7349	0.6624	0.5896
Recall	Micro	**0.5858**	0.5661	0.4494	0.4299
	Macro	**0.5858**	0.5793	0.4718	0.4235

Table 4.2: Summary Statistics scores over models. **Higher** is **better**. Our SAJ-MTL Joint-Intra variant is the best performing model due to its consideration of intra- and inter-group disagreements between annotators. Due to majority voting inside a group, Joint-Inter fails to learn some subtle distinctions within the group, hence a slightly lower performance. Flesig's performance shows that text-based augmentation approaches for learning group-conditioned toxicity labels might need to rely on several auxilalry data for better detection. Finally, as expected, the Target-only CondMTL framework has the least performance since it tends to learn the majority voted opinion across all groups for a specific target group.

The performance of Flesig's [38] model, which employs text-based augmentation for learning group-conditioned toxicity labels, suggests that while augmentation can help, it may require additional auxiliary data sources to achieve optimal detection, as used in their work. However, given current data collection practices around toxicity tasks, such auxiliary info is rarely present in most publicy available datasets. Without sufficient complementary data, the model struggles to generalize well across diverse groups, which limits its ability to capture the full complexity of group-specific toxic language patterns. Lastly, as anticipated, the Target-only CondMTL [56] framework exhibits the lowest performance. This model focuses on learning the majority opinion across all groups for a specific target group, but this approach overlooks both intra- and inter-group disagreements. As a result, it tends to align with the dominant perspective, failing to accommodate the diversity of opinions that exist across and within demographic groups.

Fig. 4.3 showing the F1 scores for the Toxic class reveal that our proposed Joint-Intra and Joint-Inter models perform comparably to each other and significantly

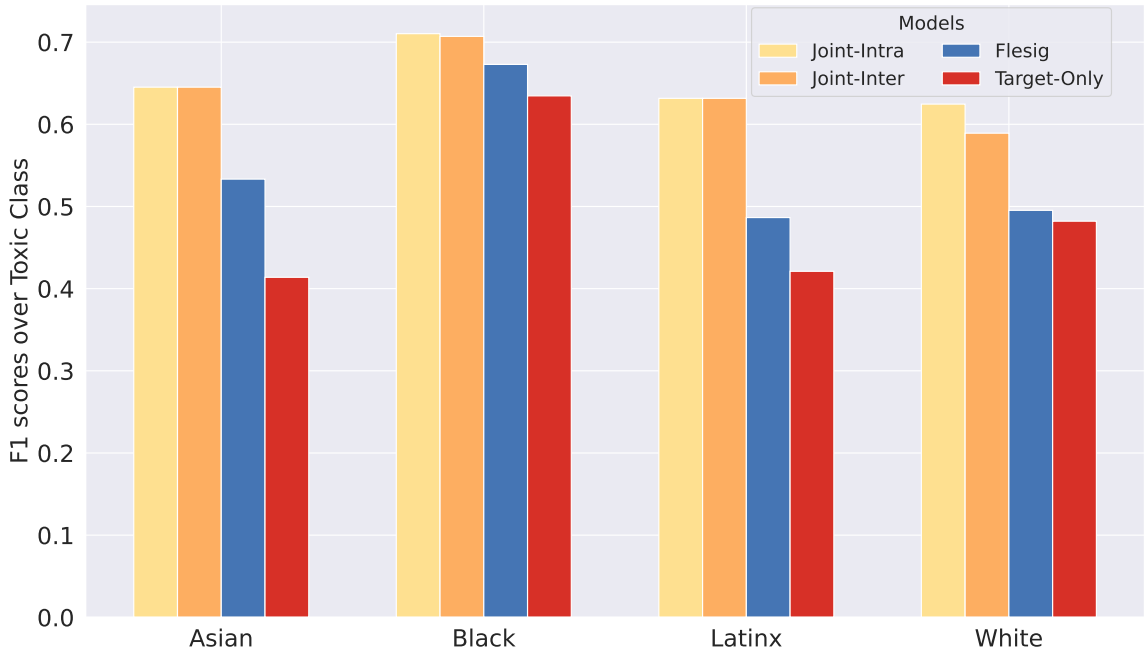


Figure 4.3: F1 scores for the Toxic class. The Joint-Intra and Joint-Inter models perform similarly and outperform the Flesig and Target-Only models by better capturing annotator and target demographics. Notably, Joint-Intra outperforms Joint-Inter for White-targeted posts, with empirical results indicating its effectiveness in handling intra-group disagreements among White annotators.

outperform both the Flesig and Target-Only models across all four groups. By jointly considering both annotator and target demographics through group-conditioned losses, our SAJ-MTL models effectively capture how annotators perceive toxicity based on target groups. Since the *Black*-group in the statistical majority in the dataset, we observe the best F1 scores for that group across the models.

When comparing our models, Joint-Intra and Joint-Inter across different demographic groups, we observe similar performance for the Asian, Black, and Latinx groups. This empirical observation is due to the lack of disagreements between annotators within (intra) each group. The *Asian* and *Latinx* groups have zero disagreements, while the *Black* group has one disagreement. However, for the White group, the Joint-Intra model outperforms Joint-Inter. This difference arises from the dis-

agreements among White annotators on posts targeting the White demographic (31 of 1198 test cases). The Joint-Intra model better handles these internal disagreements by explicitly modeling intra-group variability, leading to more accurate predictions.

4.4.4 Scalability Tests

Model type	# Params	Δ	Time(s)	Δ
Target Only (CondMTL)	763,653	-	2200	-
Flesig	67,075,795	+8684%	4800	+118%
Naive SAJ-MTL (4 branches)	763,653	+0%	2400	+9%
Naive SAJ-MTL (16 branches)	2,354,612	+208%	3300	+50%
Scalable SAJ-MTL (Ours)	626,133	-18%	2850	+30%

Table 4.3: Memory (trainable parameters) and training time (10 epochs) comparison between models. We treat the Target-Only model as baseline for comparison, since it does not account for annotator views. While the model size of Target-Only and Naive SAJ-MTL are the same, SAJ-MTL takes longer time to train due to varaition in data. The scalable SAJ variant is the best performing model in terms of model training time.

We report the memory (number of trainable parameters) and runtime (model training for 10 epochs) efficiency to analyze the Scalable SAJ-MTL variant *vs.* the various models in **Table 4.3**. We treat the Target-Only CondMTL model as the baseline here since it does not jointly model the annotator-target interactions. The Flesig [38] model requires the largest parameter and runtime amongst all the models. Being a text-augmentation based model, the DistilBERT layers were made trainable to learn updated feature representation from the augmented post. This unfrozen DistilBERT layer causes the massive increase in the parameter space. We trained the naive models in two versions. The SAJ-MTL (4 branches) corresponds to modeling the annotator-target pairs whose demographics match. Since our processed dataset has four demographic groups, this leads to four branches in the SAJ-MTL model. As such this model has the same number of trainable parameters as the Target-Only CondMTL model, with 9% additional runtime to converge. If we consider the interaction of all annotator-target pairs as reported in Table 4.1, we get the SAJ-MTL

(16 branches) model. Due to the presence of 16 task-specific layers in this model, we get a massive increase in space and runtime, highlighting the multiplicative complexity of the joint interaction of stakeholders in our SAJ-MTL framework.

Finally, the Scalable SAJ-MTL model only has eight branches corresponding to its additive architecture in Fig. 4.2. Although having comparable results *w.r.t.* naive SAJ-MTL models, this proposed scalable pipeline scales well in terms of both memory and runtime. We can observe that even with the four demographic groups (*Asian, Black, Latinx and White*) and two stakeholder (Annotator and Target) interactions, the scalable variant is -18% faster than the baseline compared to the 16 branches $+208\%$. The runtime increase of $+30\%$ in this low cardinality setup expected since the scalable model has eight branches as compared to just four in the baseline model. Thus, from a computational viewpoint, as the cardinality of groups increases in any stakeholder identities, the Scalable model would be able to accommodate model training within the same machine, without GPU memory saturation.

4.4.5 Discussion

Modeling Annotator Disagreements. As suggested in Jury Learning [50], there is substantial disagreement on what the correct label ought to be for toxicity labeling tasks, indicating that it is impossible to create a classifier that makes every user happy. The quantity of interest, *i.e.,* the final label is rarely just a question of how many people disagree, but who disagrees, to highlight the differences in labels produced when the same post is reviewed by people of different demographics. These differences are present both at the individual and group levels. By accounting for the Inter-group differences and Intra-group disagreements, we observe that our Joint-Intra variant indeed performs the best empirically across the evaluation measures *vs.* other models, supporting the hypothesis in Jury Learning.

Is Toxicity Detection Solved? Referring to values in Table 4.2, we can observe that the best recall obtained is still below 0.6, meaning that we are only able to detec-

tion 60% of the toxic posts. This emphasizes that more modeling and data collection needs to be done around this task. As the Hugging Face dataset specifically focused on collecting posts that targeted the *Black* group, the model is able to score the best (0.7 Recall) for that group. For the *White* group, the performance gap between Joint-Intra and Joint-Inter also highlights the variability of opinions of perceived toxicity by *White* annotators when the post is targeting *White* group. So, while we are able to demonstrate that toxicity detection cannot be captured by a one-size-fits-all classifier, even our own model, with improved performance, still lags behind of what might be considered an acceptable performance value for a standard ML classification problem. Thus, there needs more data per group, both quality and quantity wise, for our model to learn more on the viewpoints of individuals and groups.

Chapter 5: Pareto Manifold Tracer for MOO

Multi-Objective Optimization (MOO) problems require balancing multiple objectives, often competing with one another under constraints [35, 145]. This chapter summarizes the ideas, methodologies and findings across **three of my MOO related works** [53, 54, 135] (**arxiv, ICITR 21, UAI 22**).

TLDR: Work contributions in this chapter are sectioned as follows:

1. We propose a PINN based network, Hybrid Neural Pareto Front (HNPF), based on Fritz John Conditions, to learn a manifold over weak Pareto points.
2. We provide application-driven scenarios and numerical correctness checks to distinguish between Pareto Front *vs.* Dataset Optima.
3. We develop a scalable solver, Scalable HNPF (SUHNPF), that acts as Hypernetwork to trace approximate Pareto manifold over large scale models.

We adopt Pareto definitions from [94]. A general MOO problem can formulated as follows:

$$\begin{aligned} &\text{optimize} \quad F(x) = (f_1(x), \ldots, f_k(x)) \\ &\text{s.t.} \quad x \in \mathcal{S} = \{x \in \mathbb{R}^n | G(x) = (g_1(x), \ldots, g_m(x)) \leq 0\} \end{aligned} \tag{5.1}$$

with n variables $(x_1, \ldots, x_n)$, k objectives $(f_1, \ldots, f_k)$, and m constraints $(g_1, \ldots, g_m)$. Here, S is the feasible set, *i.e.,* the set of input values x that satisfy the constraints $G(x)$. For an MOO problem optimizing $F(x)$ subject to $G(x)$, the solution is usually a manifold as opposed to a single global optimum, therefore, one must find the set of all points that satisfy the chosen definition for an optimum. A solution point x is optimal, if it is a stationary point of the function f, *i.e.,* its gradient at x is zero $(\nabla_x f = 0)$. A *Pareto optimal* solution [109] defines the set of all saddle points [35] such that no objective can be further improved without penalizing at least one other objective.

Type	Method	Finds Only Pareto points	Handles Constraints	Scalable Neural MOO
Operations Research (OR)	NBI [24]	✓	✓	✗
	mCHIM [47]	✓	✓	✗
	PK [113]	✓	✓	✗
	HNPF [135]	✓	✓	✗
Multi-Task Learning (MTL)	MOOMTL [132]	✗	✗	✓
	PMTL [84]]	✗	✗	✓
	EPO [91]	✗	✗	✓
	EPSE [90]	✗	✗	✓
	PHN [106]	✗	✗	✓
Ours	**SUHNPF**	✓	✓	✓

Table 5.1: SUHNPF *vs.* existing Operations Research (OR) and Multi-Task Learning (MTL) methods. OR methods account for both objectives and constraints, produce Pareto points only, and are known to find true Pareto points for non-convex MOO problems. However, these methods do not scale to high-dimensional neural MOO problems. In contrast, MTL methods scale well but typically do not support constraints and can struggle with non-convexity.

5.1 Hybrid Neural Pareto Front (HNPF)

HNPF learns a neural Pareto manifold from training data. With HNPF, Pareto points are acquired from training data via FJC by treating the loss function like a Physics Informed Neural Network (PINN).

This section is based on the work reported in "A Hybrid 2-stage Neural Optimization for Pareto Front Extraction", Gupta, Singh, Lease and Dawson. Arxiv edition: https://arxiv.org/pdf/2101.11684

5.1.1 Fritz Jon Conditions (FJC)

Let the objectives and constraints in Eq. (5.1) be differentiable once at a decision vector $x^* \in \mathcal{S}$. The Fritz-John [80] necessary conditions for x^* to be *weak* Pareto optimal is that vectors must exists for $0 \leq \lambda \in \mathbb{R}^k$, $0 \leq \mu \in \mathbb{R}^m$ and $(\lambda, \mu) \neq (0, 0)$ (not identically zero) *s.t.* the following holds:

$$\sum_{i=1}^{k} \lambda_i \nabla f_i(x^*) + \sum_{j=1}^{m} \mu_j \nabla g_j(x^*) = 0 \tag{5.2}$$
$$\mu_j g_j(x^*) = 0, \forall j = 1, \ldots, m$$

Gobbi et al. [48] present an L matrix form of FJC:

$$L = \begin{bmatrix} \nabla F & \nabla G \\ \mathbf{0} & G \end{bmatrix} \quad [(n+m) \times (k+m)] \tag{5.3}$$

$$\nabla F_{n\times k} = [\nabla f_1, \ldots, \nabla f_k]$$

$$\nabla G_{n\times m} = [\nabla g_1, \ldots, \nabla g_m]$$

$$G_{m\times m} = diag(g_1, \ldots, g_m)$$

comprising the gradients of the functions. The matrix equivalent of FJC for x^* to be Pareto optimal is to show the existence of $\delta = (\lambda, \mu) \in \mathbb{R}^{k+m}$ (*i.e.,* δ not identically zero) in Eq. (5.2):

$$L \cdot \delta = 0 \quad \text{s.t.} \quad L = L(x^*), \delta \geq 0, \delta \neq 0 \tag{5.4}$$

Therefore the non-trivial solution for Eq. (5.4) is:

$$det(L^T L) = 0 \tag{5.5}$$

Remark. If f_is and g_js are continuous and differentiable once, then the set of weak Pareto optimal points are $x^* = \{x | det(L(x)^T L(x)) = 0\}$, $\delta \geq 0$ for a non-square matrix $L(x)$, and is equivalent to $x^* = \{x | det(L(x)) = 0\}$, $\delta \geq 0$, for a square matrix $L(x)$. See [54] for a proof of the above for the unconstrained setting only.

5.1.2 Framework

HNPF's neural network first identifies *weak Pareto* points via feed-forward layers to smoothly approximate the *weak* Pareto optimal solution manifold $M(X^*)$ as $\tilde{M}(\tilde{X}, \Phi)$. The last layer of the network has two neurons with *softmax* activation for binary classification of Pareto *vs.* non-Pareto points, distinguishing sub-optimal points from the *weak* Pareto points. The network loss is representation driven, since the Fritz John discriminator (Eq. 5.5), described by the objective and constraints, explicitly classifies each input data point X_i as being *weak* Pareto or not. After identifying weak Pareto points, HNPF uses an efficient Pareto filter to find the subset of *non-dominated* points.

HNPF's scalability bottleneck lies in how it samples variable domain points to test for Pareto optimality in model training. If there are any direct constraints on

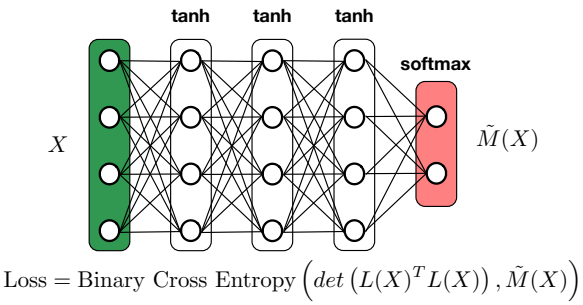


Figure 5.1: Caption

variable values, this naturally restricts the feasible domain for sampling. However, lacking any prior distribution on where to find Pareto optima, HNPF performs uniform random sampling in the variable domain to ensure broad coverage for locating optima. For small benchmark problems with known variable domains, this suffices. However, it is infeasible to apply this to find optimal parameters for a neural MOO model.

5.2 Pareto Front *vs.* Dataset Optima

Our review of related work in Information Retrieval (IR) suggests several questions are of great interest. Firstly (1), what is the true Pareto front induced by specified objective functions? Secondly (2), given a finite dataset, what are the best solutions that can be achieved on that dataset? Finally (3), how close or far are the best dataset solutions from the true Pareto front?

This section is based on the work reported in "Pareto Solutions vs Dataset Optima: Concepts and Methods for Optimizing Competing Objectives with Constraints in Retrieval", Gupta, Singh, Das and Lease, ICTIR 21.
Online edition: https://dl.acm.org/doi/abs/10.1145/3471158.3472248

To find the true Pareto front (1), a tremendous body of prior work in optimization [32, 94] is readily available, supporting both convex and non-convex MOO problems, as well as optimization under constraints. Prior work here also includes

canonical benchmark problems on which correctness of methods is routinely validated. Regarding (2), we wish to know the best solutions that can be achieved on a given dataset. Intuitively, one can simply evaluate objective functions on each data point and return the maxima achieved for any single objective. To identify the "frontier" of discrete points achieving optimal trade-off between objectives, classic methods can be applied to induce the non-dominated dataset *hull* [97]. Finally (3), how close or far are the best dataset solutions from the true Pareto front? Given the true Pareto front found in (1) and the functional domain hull for a given dataset (2), we can measure the distance from points on the dataset hull to the true Pareto front. Application specific datasets or models can strive towards minimizing the distance between the domain hull and the true Pareto front.

We define several IR inspired practical scenarios [45, 53] around *relevant and diverse document retrieval* and illustrate the difference between true front *vs.* data hull achieved on limited sampling of data. Cases include non-/competing objectives under both un-constrained and constrained settings, with verifiable numerical correctness checks with tunable trade-off between relevance *vs.* diversity.

5.3 Scalable HNPF

To address HNPF's scalability bottleneck, we introduce SUHNPF, a scalable variant of HNPF for finding weak Pareto points with an arbitrary density and distribution of initial data points. This is achieved via a scalable unidirectional FJC-guided double-gradient descent algorithm that encompasses HNPF's neural manifold estimator. Given continuous differentiable loss functions, SUNHPF's guided double gradient descent strategy efficiently searches the variable domain to find Pareto optimal points in the function domain. This enables SUHNPF to learn an ϵ-bounded approximation $\tilde{M}(\Theta^*)$ to the weak Pareto optimal manifold.

This section is on the work reported in "Learning a neural Pareto manifold extractor with constraints", Gupta, Singh, Bollapragada and Lease, UAI 22. Online edition: https://proceedings.mlr.press/v180/gupta22a/gupta22a.pdf

Constructing a classification manifold of Pareto *vs.* non-Pareto points requires a set of feasible points to represent both classes. Since the Pareto manifold is unknown *a priori*, feasible points are drawn from a random distribution (lacking an informed prior) to initialize both classes. We then refine the points in the Pareto class $\mathcal{P}1$ while holding the non-Pareto points $\mathcal{P}0$ constant. We assume an equal-sized sample set of P points for each class, which helps to address class imbalance for harsh cases. For benchmark problems where the feasible set over the variable domain is known, we randomly sample points over this feasible domain to initialize $\mathcal{P}1$ and $\mathcal{P}0$. Given these input points x, held constant for $\mathcal{P}0$ and used as initial seed values for $\mathcal{P}1$, **Alg. 2** specifies our FJC-guided double-gradient descent algorithm. The algorithm iteratively updates $\mathcal{P}1$ towards the Pareto manifold via FJC-guided descent. The training dataset D is the union of $\mathcal{P}0 \cup \mathcal{P}1$. The algorithm iterates over Steps 5-9 until the error (err) converges to the user-specified error tolerance (ϵ_{outer}).

$$err = \sum_{p \in \mathcal{P}1} \left(det(L^T L)\right)^2 \tag{5.6}$$

Algorithm 2 FJC-guided descent of variable domain

1: **Input**: Data $D = \mathcal{P}0 \cup \mathcal{P}1$ ▷ Training Data
2: **Input**: Functions F and Constraints G
3: **Input**: Error tolerance ϵ_{outer}, ϵ_{inner}
4: **while** $err > \epsilon_{outer}$ **do** ▷ Run until convergence
5: Train network using D as data for e epochs
6: Compute current error err
7: Compute $\nabla_p det = \frac{\partial det(L^T L)}{\partial p}$, $\forall p \in \mathcal{P}1$
8: $\mathcal{P}1 \leftarrow \mathcal{P}1 - \eta \nabla det$ ▷ Update points in $\mathcal{P}1$
9: $D = \mathcal{P}0 \cup \mathcal{P}1$ ▷ Update Training Data
10: **Output**: Weak Pareto manifold $\tilde{M}$

Chapter 6: Group Accuracy Parity

Toxic language in social media is often associated with various risks and harms: cyber bullying, discrimination, mental health problems, and even hate crimes. Given the massive volume of user generated content online, manual review of all posts by human moderators simply does not scale. Consequently, natural language processing (NLP) methods have been developed to fully or partially automate toxicity detection [130]. Prior work has achieved high Accuracy and F1 scores on TL detection (*e.g.,* [158]) across various model architectures: *e.g.,* convolutional (CNN) [44], sequential (BiLSTM) [52], and transformer (BERT) [29]. However, studies have also found that model accuracy can vary greatly across sensitive demographic attributes, such as race or gender [25, 110, 129]. For example, a BERT-based classifier obtains 90.4% *vs.* 84.5% accuracy for White *vs.* African American author groups on Davidson's dataset [27] when just optimized for overall accuracy, independent of author groups.

This chapter is based on the work: "Learning Optimal Accuracy vs. Fairness Tradeoffs for Hate Speech Detection", Gupta, Kovatchev, Das and Lease. Arxiv edition: https://arxiv.org/pdf/2204.07661 (Unpublished)

TLDR: Work contributions in this chapter are sectioned as follows:

1. We propose a differentiable variant of the Accuracy Parity (AP) [160] fairness measure — Group Accuracy Parity (GAP) — to optimize for equal accuracy over binary groups, with strict mapping between GAP and AP.
2. We use a MOO hypernetwork — SUHNPF [55] — to study Pareto trade-off for overall accuracy *vs.* GAP (group fairness), across three neural models.
3. We empirically show better performance using GAP *vs.* two other differentiable measures, *w.r.t.* author and target groups across two datasets.

6.1 Group Accuracy Parity (GAP)

While recent years have seen a rapid progress in fairness research, it is often measured in a post-hoc manner and optimization is often indirect (*e.g.,* by improving the training data via pre or post-processing) [129]. A particular challenge is that most existing measures are non-differentiable and thus cannot be optimized directly via gradient descent. While one can optimize differentiable, surrogate loss functions instead, this risks *metric divergence* between the optimization criteria used in training *vs.* the actual metrics of interest [98, 101, 140, 157]. While a standard Cross Entropy based loss will maximize performance for the overall/majority groups, we also need to provide protections to minority groups as well, from a fairness perspective. Therefore, our intended measure should be able to optimize for comparable performance across all groups involved. As Friedler et al. [43] and others have noted, different worldviews lead to conflicting definitions of fairness that are mutually incompatible. Since one cannot have it all, specific fairness measures must be selected (that are suitable to the given task, context, and stakeholders at hand). In this work, we adopt a popular fairness objective of optimizing a model to provide balanced accuracy across demographic groups [9, 25, 61, 100] *i.e., Accuracy Parity* (AP) [160].

6.1.1 Related Work

Toxicity Detection and Fairness. Detection of toxic language in its various forms [40] has attracted significant attention due to its prevalence in online social media platforms. To date, many datasets have been created to support model training and evaluation [27, 41, 74, 115, 117, 146, 150, 151]. While NLP/ML methods tend to optimize for overall performance, recent studies highlight the racial bias induced in such classification tasks, when group identifiers are not considered during model training [28, 129, 155]. Here, fairness concerns with respect to multiple stakeholders arise, including the author of the post [129] and groups targeted by a post. We focus our attention on fairness with respect to demographic groups targeted in the content.

To address the problem of bias in toxic language detection, a variety of work has sought to improve the training and testing data [51, 62, 77, 110, 122, 126, 129], with the expectation that fairer data will lead to fairer models. Sap et al. [129] propose to resolve the problem using race and dialect priming during annotation. Park et al. [110] propose a data-focused fairness approach for the closely related problem of gender bias. They create contrastive examples by using gender swap data augmentation. Röttger et al. [122] framed a set of functional tests and argued that automated toxicity detection models should be evaluated on all functionalities.

Fairness Measures. Measures based on confusion matrices (*e.g.,* accuracy, precision, recall, and F1) help measuring performance of machine learning systems. However, several studies unveil ML systems can be discriminatory based on different sensitive attributes (race, gender *etc.*). The amplification of systemic unfairness through AI applications has been pronounced across different critical application areas such as hiring, finance, legal applications, and content moderation [3]. To address this, several methods have been introduced to quantitatively measure and mitigate unfairness in machine learning systems borrowing from legal literature on anti-discrimination [36]. Friedler et al. [43] show that these different worldviews can lead to conflicting statistical targets, which makes it *impossible* to simultaneously achieve conflicting fairness targets. Because fairness measures can be at odds with each other based on the underlying assumptions and statistical choices [104], selecting an appropriate fairness metrics often depends on the task, use case, and stakeholder priorities [7, 42, 66].

Differentiable Fairness Losses. Typically, toxicity detection systems are trained with a single objective of minimizing cross-entropy [41, 110, 122], which is differentiable. For imposing a fairness constraint in the optimization, a standard approach is to add a differentiable regularization term with a hyper-parameter λ, as shown in **Eq. 6.1**. While decreasing cross-entropy leads to decreasing Overall Error (OE), effectively increasing Overall Accuracy (OA), we typically need a fairness loss function

whose decrease leads to increase in a corresponding fairness evaluation measure.

$$\min \quad \text{Cross-Entropy loss}\,(f_1) + \lambda \cdot \text{Fairness loss}\,(f_2) \tag{6.1}$$

While many fairness evaluation measures exists, our survey of existing fairness-related loss functions, that strictly optimizes for an equivalent measure during model training, finds only few variants. For *e.g.,* the CLA [133] loss has one-on-one correspondence for optimizing False Negative Rates across groups. Other adversarial losses like ADV [155] tries to optimize for False Positive Rates, but suffers from metric divergence. Ranking literature suffers from lack of equivalent optimizers owing to the discontinuous nature of rank order, hence they focus on designing surrogates with close asymptotic bounds [107, 140].

Same Usage, Very Many Monikers. In this work we focus on optimizing for *Accuracy Parity* (AP) [160], which refers to equal performance of a model across different demographic groups, ensuring that accuracy remains consistent irrespective of group membership. This is a popular fairness measure and has been proposed in literature, yet with very many different names. The ones we found include *Accuracy Equity* [31], *Equality of Accuracy* [61], *Equal Accuracy* [100], *Overall Accuracy Equality* [9] and *Accuracy Difference* [25]. However, note that all these work have used AP as an evaluation measure in their classification protocol to evaluate degrees of un-fairness.

RQ (4a). Can we design a differentiable fairness measure corresponding to *Accuracy Parity*, which accounts for balanced accuracy across groups?

6.1.2 Accuracy Difference

While AP is an equality condition, we still need to quantify the deviation from equality in cases of unequal performance across groups. We therefore use *Accuracy difference* (AD) Das et al. [25], a continuous version of AP to measure this deviation. AD is shown in (**Eq. 6.2**), where $\hat{y}, y, g$ are the predicted label, true label, and group

attribute respectively. Thus AD is defined based on the confusion matrix. Since the formulation is probabilistic in nature, *i.e.,* ratio of numbers over the dataset, and not distribution over variable, AD becomes non-differentiable. That is, AD can only be used in a post-hoc manner and cannot be directly used for gradient-based back propagation. Furthermore, Eq. 6.2 inherently assumes that the majority group accuracy ($g = 1$) will always be higher than the vulnerable group ($g = 0$), which might not always hold true, resulting in potential negative values of AD in the range [-1,1]. Naturally, as a post-hoc measure, AD is disconnected from the optimization objective of the model used during training.

$$AD = \underbrace{P[\hat{y} = y|g = 1]}_{\text{Acc Group 1 (g=1)}} - \underbrace{P[\hat{y} = y|g = 0]}_{\text{Acc Group 0 (g=0)}} \tag{6.2}$$

These limitations motivated us to define a differentiable, non-probabilistic form of AD we refer to as *group accuracy parity* (GAP), which allows any descent-based model during training to optimize close to equal accuracy across sensitive attribute classes, and addresses the range issue of AD.

6.1.3 GAP Formulation

Cross-Entropy (CE) loss is commonly used as a loss function in classification tasks and is designed to measure the difference between probability distribution ($\hat{y}$) predicted by the model and the true distribution (y) of the data. CE is a general differentiable loss that can be used to optimize over the entire data or independently across groups (g). Although not a strict one-to-one correspondence, it is generally observed that minimizing CE leads to minimizing Overall Error (OE), thereby maximizing Overall Accuracy (OA), due to CE providing non-asymptotic guarantees and placing an upper bound on the estimation error of the actual loss [93].

$$CE(y, \hat{y}) = \sum_{c \in \text{class}} y(c) \log(\hat{y}(c)) \tag{6.3}$$

For balanced classification across groups, *e.g.,* demographic information of post subject, we formulate our GAP loss function as follows: we first calculate the CE across each group, then minimize the difference across them and finally frame it as a Single Objective Optimization problem corresponding to Eq. 6.1. The GAP loss function in **Eq. 6.4** is equal to overall cross entropy loss ($\mathbf{OE} = \mathbf{CE}(\mathbf{y}, \hat{\mathbf{y}})$) in Eq. 6.3 only when both CE errors are equal across the groups.

$$GAP = OE + \lambda \| \underbrace{CE(g=1)}_{\text{err Group 1 (g=1)}} - \underbrace{CE(g=0)}_{\text{err Group 0 (g=0)}} \|_2^2 \tag{6.4}$$

Remark. GAP optimizes for AP, to reduce the accuracy gap across groups, therefore minimizing this type of disparate impact across groups. Additionally, GAP formulation is flexible with different weighted variants of accuracy *w.r.t.* chosen entropy, depending on the evaluation need.

The formulation of the GAP loss in Eq. 6.4 is generalizable to different weighted variants of accuracy. For *e.g.,* Binary Cross Entropy (BCE) in **Eq. 6.5** is used as a loss function for optimizing a binary classifier *i.e.,* labels of 0s and 1s. BCE, however, does not take into account the label imbalance.

$$BCE = -\frac{1}{N} \sum_{N} y \log(\hat{y}) + (1-y) \log(1-\hat{y}) \tag{6.5}$$

Weighted Binary Cross Entropy (wBCE) re-weights the error for the class labels proportional to their inverse frequency in the data. The class re-weighting strategy $\mathbf{w}(\cdot)$ in **Eq. 6.6** is available in packages like SkLearn [112] and discussed in detail by Lin et al. [83]. Since most real-world datasets often have label imbalance, wBCE aims to penalize both labels (**1**s and **0**s) equally. Thus, minimizing wBCE leads to maximizing balanced accuracy (BA), a better measure than accuracy, in presence of label imbalance.

$$wBCE = -\frac{1}{N} \sum_{N} w(y) \cdot y \log(\hat{y}) + w(1-y) \cdot (1-y) \log(1-\hat{y}) \tag{6.6}$$

Remark. In this implementation we choose the weighted Binary Cross Entropy ($wBCE$) variant in the GAP terms to account for binary label imbalance within each group. Consequently, we use Balanced Accuracy (BA) to evaluate performance of each group, to maintain functional mapping.

We summarize key properties of GAP:

1. **GAP maps to AD.** GAP has a strict mapping to AD *i.e.,* minimizing GAP also minimizes AD.

2. **GAP is differentiable.** GAP is defined as the summation of overall error and the squared 2-norm difference between the wBCE across the groups. Since wBCE is differentiable, so is the 2-norm difference. Hence GAP can be optimized for any descent based model.

3. **GAP is smooth.** GAP has a 2-norm formulation, thus the range of attainable evaluation values are within $[0, 1]$, avoiding the negativity issue in AD. Being a squared 2-norm measure, the surface of GAP is smoother than other comparable measures like CLA [133], which uses 1-norm. A smoother surface leads to better descent rates [15].

6.2 Optimizing Competing Objectives - Pareto Trade-off

Typically, TL detection systems are trained with the single objective of maximizing OA [41, 110, 122] or a custom defined objective [155]. In contrast, we frame TL detection as a Multi-Objective Optimization (MOO) problem. It is important to highlight the distinction between an M(Multi)OO *vs.* S(Single)OO formulation and their interpretation. Consider the two objectives as f_1: Cross-Entropy and f_2: Fairness. Traditional fair classifiers operate by adding a penalty term corresponding to

Fairness to the main objective Entropy with a hyper-parameter λ in **Eq. 6.7**.

$$\begin{aligned} &\min \quad f_1 + \lambda f_2 \\ &\min \quad \text{Cross-Entropy loss} + \lambda\, \text{Fairness loss} \end{aligned} \tag{6.7}$$

The reader is requested to note that such optimization process does not have any control over the range of λ, and it can vary between $(0, \infty)$. During the optimization process, we tune λ till we get a desired performance in SOO setting. Furthermore, there is no explicit requirement of the scale of f_1 and f_2 to be the same. Thus, there is no simple correlation between the amount of Fairness we want *vs.* the value of λ.

An unconstrained MOO problem with two competing loss objectives is defined in **Eq. 6.8**. Note that this is a joint min-min problem instead of a single min problem. The objectives here need to be at the same scale. If a linear trade-off is expected between them, the linear scalarized form of the MOO problem with trade-off $\alpha \in [0, 1]$, minimizes both objectives simultaneously in **Eq. 6.9**. Solving this reformulated MOO problem would achieve balance between Entropy and Fairness, with α holding strict mathematical interpretation of linear trade-off. Decreasing Entropy causes Fairness to increase, while decreasing Fairness causes Entropy to increase.

$$\min \min \quad f_1\,,\, f_2 \tag{6.8}$$

$$\begin{aligned} &\min \quad \alpha f_1 + (1-\alpha) f_2 \\ &\min \quad \alpha\, \text{Cross-Entropy loss} + (1-\alpha)\, \text{Fairness loss} \end{aligned} \tag{6.9}$$

Note that there are multiple mathematically optimal solutions to **Eq. 6.9**. Every optimal solution corresponding to each value of α in Eq. 6.9 is a member of the Pareto optimal solution set *i.e.,* the Pareto front contains the set of optimal model parameters given the dataset and the model. To solve this MOO problem, we adopt the SUHNPF Pareto framework [55] as a HyperNetwork [58] to learn optimal TL detection neural model parameters over trade-offs. Hypernetworks train one neural model to generate effective weights for a second, target model.

SUHNPF efficiently learns the entire Pareto manifold of feasible trade-off values during training. This empowers users to then choose any solution point they prefer on the manifold, *a posteriori*, and extract the classifier weights configuration as per their desired trade-off α, without retraining the model for that α. Training the same model for K different α's, with R being the time for a single run, would result in total runtime of $K \times R$ *i.e.,* linear on the number of runs. Using the Hypernetwork to learn the manifold is computationally much more efficient *i.e.,* taking a constant time $c \times R$, $1 < c \ll K$ over feasible α's, rather than for each value of α.

RQ (4b). How we use existing MOO frameworks to approximately and efficiently trace out the trade-off space of competing measures?

6.3 Experimental Results

6.3.1 Datasets

We consider two datasets: Davidson *et al.* [27] for author demographics and the *Civil Comments* [13] portion of *Wilds* [74] for target demographics (**Table 6.1**). In each case, we frame the task as a binary classification problem (Toxic *vs.* non-Toxic, or "safe") with binary group attributes (Majority *vs.* Minority groups). Note that "Majority" and "Miniroty" in our work simply refers to the representation of the group in the data and does not carry any social or cultural meaning.

Dataset	Group	Toxic	Safe	Total
Davidson	Minority	8,725	302	9,027 (36%)
	Majority	11,895	3,861	15,756 (64%)
Wilds	Minority	5,973	33,762	39,735 (44%)
	Majority	6,832	42,950	49,782 (56%)

Table 6.1: Statistics of the two datasets used in this work. For Davidson *et al.* [27], we consider the author demographics AAE *vs.* SAE as group attribute for minority *vs.* majority group attributes. For Wilds [74], we consider the binary group target gender as male *vs.* female for minority *vs.* majority group atttributes.

Author Demographics Dataset We consider fair moderation of posts writ-

ten by authors from different demographic groups in [27]. Prior studies [4, 129] have empirically demonstrated the existence of bias towards author demographics in TL classification. The sensitive attribute in this dataset is *race*, as identified by the dialect of the tweets. Following prior work, we apply Blodgett *et al.* [12]'s model to automatically-detect dialect labels for each of the tweet as African-American English (AAE) or Standard American English (SAE), representing *Minority* and *Majority* groups, respectively. We acknowledge both that dialect is only a weak surrogate representation of demographic race, and that automatic detection of dialect will naturally incur noise. However, in this we follow established practices from prior work. Our fairness methods are agnostic as to the sensitive attribute labeled in the data, and our results are only intended to attest to the capabilities of our proposed methods, rather than provide findings regarding protection of any specific vulnerable population. Davidson *et al.* [27]'s data includes 24,783 Twitter posts labeled as Hate, Offensive, or Normal. Following prior work [110], we set the class label to 1 (Toxic) if the post contains hate speech or offensive language, and 0 otherwise. We note that tweets from *Minority* authors are annotated as toxic in 96% of the cases, compared to 75% for the tweets by *Majority* authors. While these statistics suggest an important risk of annotation bias in this dataset, dataset debiasing lies beyond the scope of our work. Our focus in this work is restricted to balancing accuracy across the groups, given the dataset as it is annotated.

Target Identity Dataset To assess fair protection of different groups targeted in posts, we use the *Civil Comments* [13] portion of *Wilds* [74]. This dataset has 448,000 training tweets labeled as Toxic or non-Toxic. Each tweet has explicit annotation for the demographics, gender, or religion of the target entity. We select tweets where more than 50% of annotators agreed on the gender of the target. In this work, we include only female (majority) and male (minority) genders in order construct a binary sensitive attribute for our experiments. In doing so, we fully acknowledge both the non-binary nature of gender and individual freedom of self-identification. Our method is agnostic as to the sensitive attribute in the data, and our inclusion

of only two genders merely reflects a convenient way to assess the capabilities of our proposed method in regard to balancing accuracy across a binary groups.

6.3.2 Neural Models Considered

To assess the generality of GAP, we evaluate across three distinct neural architectures: CNN [44], BiLSTM [52] and BERT [29]. For all three models, we freeze the feature representation layers and optimize the weights of the classification layer. In general, GAP loss optimization and the SUHNPF hypernetwork [55] support such generalization across any models that can be trained via gradient descent.

6.3.3 Baseline Loss Functions

We compare against two baseline loss functions. The first baseline [133] seeks to balance False Negative Rate (FNR) across protected groups [21], also known as equality of opportunity [59]. To do so, they propose a differentiable measure referred to as "CLAss-wise equal opportunity" (CLA). CLA by the nature of its formulation has a strict correspondence to its intended fairness evaluation.

$$CLA = \sum_{y \in C} \sum_{g \in G} |BCE(y, g) - BCE(y)| \tag{6.10}$$

The second baseline [155] is an adversarial approach to demoting unfairness, which we denote as ADV. It seeks to provide false positive rate (FPR) balance [21], otherwise known as *predictive equality* (*ibid.*). Being adversarial, this method and others [20] do not have any correspondence with any evaluation measure. Thus, users should be cautious of possible metric divergence while using such techniques.

$$ADV = \beta BCE + (1 - \beta)(adversary(y, g) - 0.5) \tag{6.11}$$

6.3.4 Experimental Setup

We have two experimental setup with Weighted Cross Entropy (WCE) as f_1 and Fairness criteria as f_2. First, we optimize the fairness measure directly as a

SOO problem following Eq. 6.7 under a penalization setting, as used in CLA [133]. Secondly, we use the MOO setting to find the trade-off between WCE and fairness measure following Eq. 6.9, with the SOO *vs.* MOO distinction shown in **Sec 6.2**.

6.3.5 Evaluation Measures

Our focus in this work is the tension between minimizing *accuracy difference* (AD) [25] and maximizing overall accuracy (OA). We thus evaluate on four post-hoc measures: OA over the dataset (majority and minority groups together), accuracy of each group separately, and AD observed between groups.

6.3.6 Existing Bias in CNN, BiLSTM, BERT

Table 6.2 presents results for three different TL classifiers optimized to maximize OA (*i.e.,* WCE) on Davidson *et al.* [27]'s dataset. The *Majority* class consistently shows 6-7% higher accuracy than the *Minority* class, across models and five random initialization. Such imbalance serves as motivation for our work to optimize OA/AD across demographic groups. This inequaility behavior in TL detection is consistent across all three neural models and both datasets. Due to space restrictions, in the rest of the paper we present only the results for the BERT-based classifier. However, our findings also apply to BiLSTM and CNN networks.

Models	Overall %	Majority %	Minority %	AD %
CNN	87.52 ± 0.3	89.12 ± 0.2	82.88 ± 0.3	6.24 ± 0.2
BiLSTM	87.60 ± 0.2	89.37 ± 0.2	82.46 ± 0.1	6.91 ± 0.3
BERT	88.84 ± 0.2	90.35 ± 0.2	84.47 ± 0.1	5.88 ± 0.1

Table 6.2: Baseline accuracy results on Davidson *et al.* [27]'s dataset when maximizing overall accuracy only. Results show consistent bias of higher accuracy for the Majority.

Table 6.3 shows the baseline results on the Wilds [74] dataset. The performance of the classifiers are similar *w.r.t.* Table 6.2, where due to focus on Overall Accuracy (OA), there is a gap between the group specific accuracies. This shows the existing bias across the three neural models, with the BERT based model performing

relatively better than the rest.

Models	Overall %	Majority %	Minority %	AD %
CNN	83.90 ± 0.2	86.11 ± 0.1	81.27 ± 0.2	4.84 ± 0.2
BiLSTM	83.94 ± 0.1	85.98 ± 0.2	81.52 ± 0.2	4.46 ± 0.1
BERT	84.71 ± 0.3	86.53 ± 0.1	82.49 ± 0.2	4.04 ± 0.2

Table 6.3: Baseline accuracy results on Wilds [74] dataset when maximizing overall accuracy (OA) only. Results show consistent bias of higher accuracy for the Majority.

6.3.7 Single Objective Optimization (SOO)

Table 6.4 shows the results for the SOO experimental setup. The baseline BERT model optimized via Cross Entropy obtains 88.84 OA and 5.88 AD on Davidson *et al.* [27] and 84.68 OA and 3.88 AD on Wilds [74]. All three loss functions successfully reduce the AD on both datasets. As expected, the improvement in fairness comes at the cost of lower OA. We evaluate the different optimization metrics by looking at both the change in AD and in OA.

ADV performs the worst of the three measures, most notably due to its relatively large drop in OA. Optimizing for GAP and CLA gives the same OA, where the two losses show no significant difference across 5 initialization. However, in terms of reducing AD, our GAP measure outperforms CLA by 0.9 on Davidson and 1.5 on Wilds. Looking at the results, we can conclude that GAP is the best performing measure in terms of reducing Accuracy Difference. The results are consistent across both datasets. These results show the value in optimizing a measure that correctly reflects the desired notion of fairness, as well as the benefit from directly optimizing the measure of interest, rather than surrogate or approximate loss functions, to avoid metric divergence.

6.3.8 Multi Objective Optimization (MOO)

In Section 6.3.7 we used GAP, CLA, or ADV to directly optimize fairness. However, the reduced AD comes at the cost of lower OA. In order to find the optimal

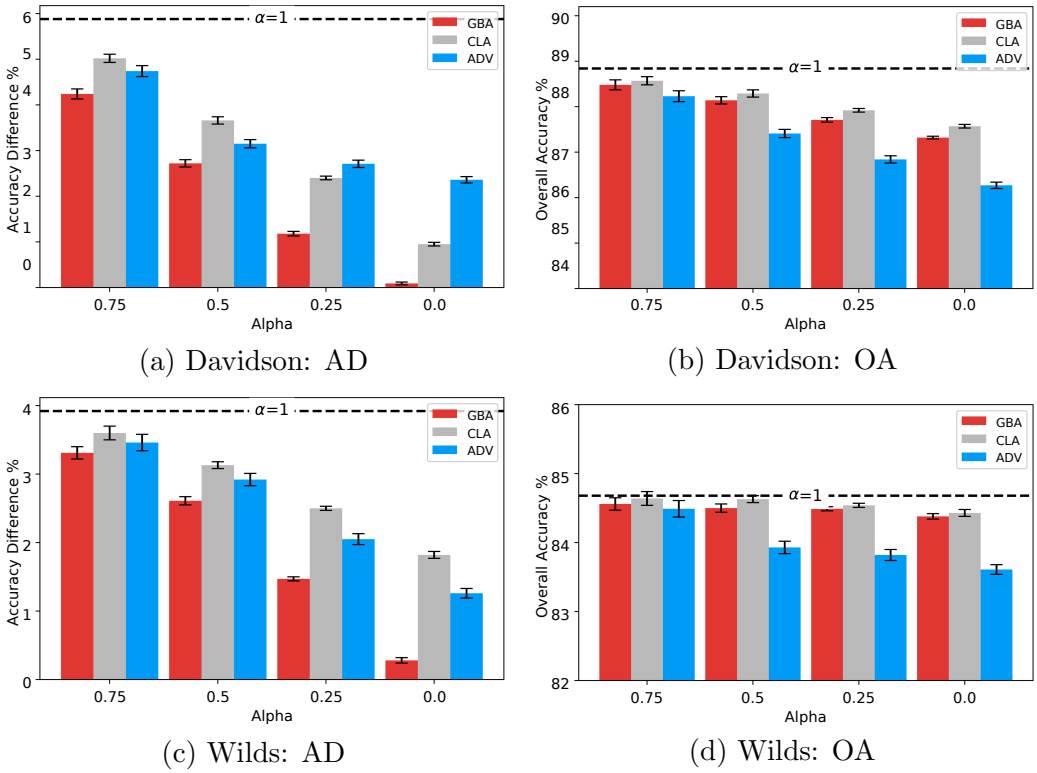
Measure	Overall %	Majority %	Minority %	AD %
	Davidson			
Baseline	88.84 ± 0.2	90.35 ± 0.2	84.47 ± 0.1	5.88 ± 0.1
GAP (Ours)	87.32 ± 0.1	87.35 ± 0.1	87.26 ± 0.1	0.09 ± 0.0
CLA	87.57 ± 0.2	87.82 ± 0.1	86.87 ± 0.1	0.95 ± 0.0
ADV	86.27 ± 0.4	86.88 ± 0.2	84.52 ± 0.3	2.36 ± 0.1
	Wilds			
Baseline	84.68 ± 0.3	86.41 ± 0.2	82.49 ± 0.1	3.88 ± 0.2
GAP (Ours)	84.38 ± 0.1	84.51 ± 0.1	84.23 ± 0.0	0.28 ± 0.0
CLA	84.43 ± 0.1	85.23 ± 0.1	83.41 ± 0.0	1.82 ± 0.1
ADV	83.61 ± 0.2	84.17 ± 0.1	82.91 ± 0.1	1.26 ± 0.1

Table 6.4: Optimizing fairness in a SOO setup. We compare a BERT-based model trained using cross entropy (Baseline) with models trained using different fairness measures. Our proposed measure (GAP) obtains the best results in reducing AD while maintaining high overall accuracy.

trade-offs between fairness and accuracy, we use the SUHNPF framework in a MOO experimental setup. We use a BERT-based classifier and three different pairs of objective functions: WCE *vs.* GAP; WCE *vs.* CLA; and WCE *vs.* ADV, learning a linear MOO trade-off between the two competing objectives.

Fig. 6.1 shows the results of the MOO experiments. The SUHNPF allows us to control how important is each objective (accuracy *vs.* fairness) by choosing the value of α. At $\alpha = 1$, we optimize only for Accuracy, and at $\alpha = 0$, only for fairness. We illustrate the different trade-offs at 4 points of the Pareto front ($\alpha = 0$, 0.25, 0.5, and 0.75). We can observe that with decreasing α, both AD and OA decrease. For ADV we can see that the drop in AD is comparable to the drop in OA, which is not an efficient trade-off between accuracy *vs.* fairness. GAP and CLA maintain a relatively consistent OA, while GAP reduces AD far more than CLA, yielding the best trade-off for each α. We can conclude that GAP is consistently the best metric, across SOO and MOO experimental setups and across different values of α for MOO.

Table 6.5 reports the the Accuracy Difference (AD) and Overall Accuracy



(a) Davidson: AD (b) Davidson: OA

(c) Wilds: AD (d) Wilds: OA

Figure 6.1: Trade-offs between Accuracy Difference (AD) and Overall Accuracy (OA), on the BERT based model with SUHNPF acting as hypernetwork for three methods — GAP (ours), CLA, and ADV – across the two datasets for $\alpha \in [0, 1]$, with $\alpha = 0$ optimizing AD only and $\alpha = 1$ optimizing OA only. GAP achieves lower AD consistently across α settings and datasets, while a more modest drop in OA is observed across methods as AD is reduced.

(OA) values achieved for the different trade-off configurations of the Bert model, across three loss measures. This is a tabulated version of **Fig. 1 (main text)**. Note that for trade-off $\alpha = 1$, only OA is maximized, hence none of the losses play any part, thus a common number across three columns, for each dataset. As the trade-off takes into account each of the loss measures, we empirically observe GAP to be performing best *w.r.t.* the other measures, since it is being optimized *w.r.t.* minimizing AD.

CLA is designed to optimize for Equal Opportunity *i.e.,* False Negative Rate across each group of sensitive attribute (g), follows similar trajectory to GAP. As these measures operate on different sections of the confusion matrix, and optimizing

α	Accuracy Difference			Overall Accuracy			F1		
	GAP (Ours)	CLA	ADV	GAP (Ours)	CLA	ADV	GAP (Ours)	CLA	ADV
				Davidson					
1.00	5.9 ± 0.1			88.9 ± 0.2			0.71 ± 0.02		
0.75	4.2 ± 0.1	5.0 ± 0.1	4.7 ± 0.1	88.5 ± 0.3	88.6 ± 0.2	88.2 ± 0.4	0.70 ± 0.01	0.69 ± 0.01	0.68 ± 0.00
0.50	2.7 ± 0.1	3.7 ± 0.1	3.2 ± 0.1	88.1 ± 0.5	88.3 ± 0.5	87.4 ± 0.6	0.69 ± 0.02	0.67 ± 0.01	0.65 ± 0.01
0.25	1.2 ± 0.1	2.4 ± 0.0	2.7 ± 0.1	87.7 ± 0.2	87.9 ± 0.4	86.8 ± 0.6	0.67 ± 0.01	0.65 ± 0.00	0.64 ± 0.01
0.00	0.1 ± 0.0	0.9 ± 0.0	2.4 ± 0.1	87.3 ± 0.1	87.6 ± 0.2	86.3 ± 0.4	0.66 ± 0.00	0.64 ± 0.02	0.61 ± 0.01
				Wilds					
1.00	3.9 ± 0.2			84.7 ± 0.3			0.65 ± 0.02		
0.75	3.3 ± 0.1	3.6 ± 0.1	3.5 ± 0.1	84.6 ± 0.2	84.6 ± 0.1	84.5 ± 0.3	0.63 ± 0.02	0.62 ± 0.01	0.62 ± 0.02
0.50	2.6 ± 0.1	3.1 ± 0.1	2.9 ± 0.1	84.5 ± 0.4	84.6 ± 0.6	83.9 ± 0.4	0.62 ± 0.0	0.61 ± 0.01	0.60 ± 0.01
0.25	1.5 ± 0.0	2.5 ± 0.0	2.0 ± 0.1	84.5 ± 0.1	84.5 ± 0.2	83.8 ± 0.5	0.60 ± 0.01	0.60 ± 0.01	0.57 ± 0.01
0.00	0.3 ± 0.0	1.8 ± 0.1	1.3 ± 0.1	84.4 ± 0.1	84.4 ± 0.1	83.6 ± 0.2	0.58 ± 0.02	0.58 ± 0.01	0.55 ± 0.02

Table 6.5: Performance of GAP *vs.* CLA, ADV across two datasets in terms of Accuracy Difference (AD) and Overall Accuracy (OA). GAP achieves lower AD consistently across α settings and datasets, while a more modest drop in OA is observed across methods. $\alpha = 1$ minimizes WCE over labels only, hence same error across the three measures.

for some values in them leads to better numbers in other parts of the table, since the total number of samples are fixed. ADV, on the other hand, tries to balance False Positive Rate across each sub-population of sensitive attribute (g). The performance of ADV however deviates a lot from the trajectory of both GAP and CLA, since their adversarial setup is not strictly optimizing for FPR, and similar deviations can be seen in their original work [155] as well.

There are various ways to define fairness and over 80 [7] different post-hoc measures for fairness, corresponding to different use-cases. We obtained the best results when using GAP: a measure designed specifically for achieving Overall Accuracy Equality (OAE). Other fairness measures such as CLA and ADV can improve the OAE to a certain degree, but are nowhere near as efficient as GAP. Because no fairness measure is universal [104], it is important to pick a loss function that corresponds to the intended fairness goal.

Chapter 7: Multi-Group GAP for Target Detection

While **Chapter 6** reviews the concept of Accuracy Parity (AP) and a develops it's corresponding differnetiable loss GAP, we did not highlight practical use case scenarios where we can justify the importance of such a group-fairness measure. In this chapter we aim to provide such practical use-case driven justification to emphasize the need for GAP, specially around fair target-group detection. We also provide extend GAP to a multi-group setting to accommodate the target-detection task.

This chapter is based on: "Fairly Accurate: Optimizing Accuracy Parity in Fair Target-Group Detection", Gupta, Kovatchev, De-Arteaga and Lease. Arxiv edition: https://arxiv.org/pdf/2407.11933 (Unpublished)

TLDR: Work contributions in this chapter are summarized as follows:

1. We emphasize the need for balanced accuracy across groups for tasks with symmetric error costs, specifically around target-group detection.
2. We extend the GAP measure to handle multiple (beyond binary) groups.
3. We show an impossibility between Equalized Odds and AP, clarifying the common misconception that satisfying EO guarantees satisfying AP.
4. We empirically show the effectiveness of GAP for balancing accuracy across groups, over 7 target demographics, with minimal drop in overall accuracy.

7.1 Need for symmetric errors in Target Detection Task

RQ (5a). From a fairness use-case, how to we ground the importance for such measure in target-group detection task?

Algorithmic fairness tasks that have received most attention in the past years are typically associated with allocating goods or burdens (*e.g.,* college admission is a good, and denying bail is a burden). In such settings, it is easy to identify a "positive"

and a "negative" class, and errors typically have asymmetric costs. For example, errors in being mistakenly granted admission (false positive) *vs.* being mistakenly denied (false negative) are not equal. However, target detection presents a multi-label prediction task where labels correspond to demographic groups (*e.g.*, *Latinx*, *Black*, and *Native American*). As such, there is no notion of "positive" and "negative" label, and the motivation to provide equal treatment to all groups results in considering errors as symmetric. As discussed Chapter 1, if a toxic post targets group-*Latinx* but is mistakenly detected as targeting group-*Black*, this would be equally undesirable as a toxic post targeting group-*Black* but mistakenly detected as targeting group-*Latinx*. Thus, a fair target detection model involves equalizing accuracy across all groups, *i.e.,* *Accuracy Parity* (AP) [160]. Specifically, for any given platform, user demographics may be highly skewed, and enforcing equal accuracy for every demographic group *may* require a trade-off in which accuracy for dominant group(s) is reduced.

7.2 Extension to Beyond Binary Groups

RQ (5b). What are feasible extensions on the proposed measure to account for multiple demographic groups (beyond binary)?

Referring to **Eq. 6.4**, one can notice that the current formulation takes into account only two groups ($g = 0, 1$) due to the binary nature of the motivating fairness measures AD [25] or Equality of Accuracy [61]. We extended the current formulation of Eq. 6.4 to include multiple groups (beyond binary), such that the equivalence the original fairness notion still holds mathematically, and not heuristic guided. **Eq. 7.1** defines the revised formulation. In particular, we sum the Cross Entropy (CE) losses over all pairs of distinct groups. This updated formulation also reaches zero error when the error differences across all the groups are minimized, *i.e.,* accuracy levels

for all demographics groups are optimized to be equal.

$$GAP = OE + \lambda \sum_{i,j \in [G], i \neq j} \| \underbrace{CE(g=i)}_{\text{err Group i (g=i)}} - \underbrace{CE(g=j)}_{\text{err Group j (g=j)}} \|_2^2 \tag{7.1}$$

The multi-group extension of GAP in **Eq. 7.1** has ${}^{G}C_2$ (G choose 2) terms in the regularization factor, corresponding to all combinations of distinct group pairs $(i, j) \in [G], i \neq j$ within the dataset of group cardinality G, while being smooth in nature due to the squared 2-norm. This formulation allows us to account for practical scenarios where a post can potentially target multiple group(s) simultaneously.

7.2.1 Code Flow

Alg. 3 presents the logic of the $wBCE$ loss in Eq. 6.6, where the weights $w[g]$ represent inversely scaled values of labels (0s and 1s), within each group g. Thus $err\_grp[g]$ equivalently maps to Balanced Accuracy (BA), an evaluation measure widely used in datasets with label imbalance [16, 65]. Absence of this weighting term would strictly map to standard accuracy, following the mapping from BCE. The overall error $err\_overall$ is then defined as summation of errors across groups. Note that we do not weight groups while adding their errors since: a) we want to treat all groups equally; and b) TensorFlow's *bce* function is scale independent, *i.e.,* it produces same error value for equal ratios of mispredictions *w.r.t.* total samples, irrespective of sample size. For *e.g., bce* value over 5 samples with 1 misprediction is equal to *bce* value over 15 samples with 3 misprediction. **Alg. 4** shows the logic of our proposed GAP loss. After computing overall error via Alg. 3, GAP computes the ${}^{G}C_2$ (G choose 2) group-pairwise errors. The errors are squared to enforce positive values and allow for a smooth loss surface. The final SOO error as per Eq. 6.1 is the summation of the overall error and a regularized sum of group-pair errors.

Algorithm 3 $overall\_loss(y\_true, y\_pred, w[g])$
Loss function for optimizing Overall Error

Input: y_true, y_pred ▷ True and Predicted Labels
Input: $w[g]$ ▷ Balanced weights of Group g
$y\_true\_lab = y\_true[:, 0]$ ▷ Label Info
$y\_true\_dem = y\_true[:, 1]$ ▷ Demographic Info
for each group $grp[g] \in G$ **do**
    $pos\_grp[g] = group(y\_true\_dem == g)$ ▷ Find indices
    $y\_true[g] = y\_true\_lab[pos\_grp[g]]$ ▷ True group labels
    $y\_pred[g] = y\_pred[pos\_grp[g]]$ ▷ Predicted group labels
    $err\_grp[g] = bce(y\_true[g], y\_pred[g], w[g])$ ▷ wBCE
$err\_overall = \sum^{G} err\_grp[g]$ ▷ summation of group loss
Output: $err\_overall$

Algorithm 4 $gba\_loss(y\_true, y\_pred, w[g])$
Loss function for optimizing Group Balanced Error

Repeat Steps 1-10 of **Alg. 3**
for group pairs $[i, j] \in G, [i \neq j]$ **do** ▷ ${}^{G}C_2$ (G choose 2) iterations
    $err\_group\_pairs = \sum(err\_grp[i] - err\_grp[j])^2$
$err\_balanced = err\_overall + \lambda \cdot err\_group\_pairs$
Output: $err\_balanced$

7.3 Incompatibility of Equalized Odds and Accuracy Parity

RQ (5c). c) Are Equalized Odds and Accuracy Parity mutually incompatible?

We next present an impossibility theorem between Equalized Odds (EO) and Accuracy Parity (AP), that, to the best of our knowledge, has not been previously identified. It challenges the common assumption that satisfying EO inherently ensures AP. While EO aims to equalize error rates, such as true positive rates (TPR) and false positive rates (FPR) across different demographic groups, it does not guarantee balanced accuracy for those groups. EO is concerned with the consistency of error rates, focusing on reducing disparate treatment across groups in terms of misclassifications. AP, on the other hand, prioritizes balanced detection accuracy across all groups, ensuring that no group is disproportionately disadvantaged in terms of correct classifications. Thus, a model can achieve EO while still exhibiting unequal detection

performance, leaving some groups with significantly lower detection accuracy than others. To formalize this incompatibility, we present the following theorem:

Theorem 7.1. *Consider a fairness problem where the goal is to simultaneously satisfy Equalized Odds and Accuracy Parity. The only scenarios in which this is feasible are when the base rates (the proportion of positive labels) are equal across all groups or when the model engages in random prediction.*

This theorem illustrates that EO and AP are fundamentally misaligned in most practical scenarios. This finding underscores the importance of selecting fairness metrics that align with the desired outcomes, particularly in tasks involving sensitive group detection, where accuracy disparities can exacerbate existing biases.

Proof. From basic definitions, we know the following:

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN}$$
$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}$$
$$\text{Accuracy (Acc)} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{Total}$$

Assume two groups A and B. Let the number of positive and negative examples in group A be P_A and N_A respectively. Similarly, P_B and N_B for group B respectively.

Base Rate: Proportion of positive to negative examples in a group, *i.e.,* $\frac{P_A}{N_A}$ and $\frac{P_B}{N_B}$.

Since we are satisfying Equalized Odds, we have:

$$TPR_A = TPR_B = TPR$$
$$FPR_B = FPR_B = FPR$$

For Group A, we have the following:

$$
\begin{aligned}
TP_A &= TPR * P_A \\
FN_A &= P_A - TP_A = P_A * (1 - TPR) \\
FP_A &= FPR * N_A \\
TN_A &= N_A - FP_A = N_A * (1 - FPR) \\
\therefore Acc_A &= \frac{TP_A + TN_A}{P_A + N_A} \\
&= \frac{TPR * P_A + N_A * (1 - FPR)}{P_A + N_A}
\end{aligned} \tag{7.2}
$$

For Group B, we have the following:

$$
\begin{aligned}
TP_B &= TPR * P_B \\
FN_B &= P_B - TP_B = P_B * (1 - TPR) \\
FP_B &= FPR * N_B \\
TN_B &= N_B - FP_B = N_B * (1 - FPR) \\
\therefore Acc_B &= \frac{TP_B + TN_B}{P_B + N_B} \\
&= \frac{TPR * P_B + N_B * (1 - FPR)}{P_B + N_B}
\end{aligned} \tag{7.3}
$$

Hence, we can observe that group accuracies under Equalized Odds setting are dependent on proportion of positive and negative examples in each group. Therefore, guaranteeing Equalized Odds does not guarantee Equalized Accuracy.

Special Case 1: Equal Base Rates across groups, *i.e.*, $\frac{P_A}{N_A} = \frac{P_B}{N_B}$, $P_A = \alpha P_B, N_A = \alpha N_B, \forall \alpha \in (0, \infty)$. Under this special setting of equalized base rates, Eq. 7.2 and Eq. 7.3 becomes the same:

$$
\begin{aligned}
Acc_A &= \frac{TPR * P_A + N_A * (1 - FPR)}{P_A + N_A} \\
&= \frac{TPR * \alpha P_B + \alpha N_B * (1 - FPR)}{\alpha P_B + \alpha N_B} \\
&= \frac{\alpha (TPR * P_B + N_B * (1 - FPR))}{\alpha (P_B + N_B)} \\
&= Acc_B
\end{aligned}
$$

Special Case 2: Unequal Base Rates across group, but TPR and FPR sum to one. $TPR + FPR = 1$. Under this special setting, Eq. 7.2 and Eq. 7.3 becomes the same. This amounts to random prediction by the model.

$$
\begin{aligned}
Acc_A &= \frac{TPR * P_A + N_A * (1 - FPR)}{P_A + N_A} \\
&= \frac{TPR * P_A + N_A * TPR}{P_A + N_A} \\
&= \frac{TPR(P_A + N_A)}{P_A + N_A} \\
&= TPR
\end{aligned}
\qquad
\begin{aligned}
Acc_B &= \frac{TPR * P_B + N_B * (1 - FPR)}{P_B + N_B} \\
&= \frac{TPR * P_B + N_B * TPR}{P_B + N_B} \\
&= \frac{TPR(P_B + N_B)}{P_B + N_B} \\
&= TPR
\end{aligned}
$$

In summary, the feasibility of simultaneously achieving Equalized Odds and Accuracy Parity is highly constrained. In most real-world scenarios, the base rates are inherently imbalanced across groups, making *Special Case 1* difficult to satisfy. The only alternative is for the model to resort to random predictions (*Special Case 2*), which undermines the utility and reliability of the system. This impossibility result highlights the need for practitioners to carefully choose and prioritize fairness criteria based on the context and constraints of their application.

7.4 Results of GAP around Target Group Detection

To assess fair target-group detection we use the HuggingFace DLab dataset [37] as used in Chapter 4. The dataset has 135,556 posts, where each post has an explicit annotation for the target group(s) *i.e.,* demographics of the target entity (*target_race*). We select posts having the target-demographics flag as *True*, where a post targets one or more groups, irrespective of the toxicity label of the post. We have boolean scores on annotator consensus for seven demographic groups: Asian, Black, Latinx, Middle-Eastern, Native-American, Pacific-Islander and White. **Fig. 7.1** shows the split of the targeted groups by posts. **Fig. 7.2** shows the split of posts by the number of groups targeted. We do a 80%-20% split for training and testing data respectively. Similar to Chapter 4, we ensure that posts do not repeat across splits.

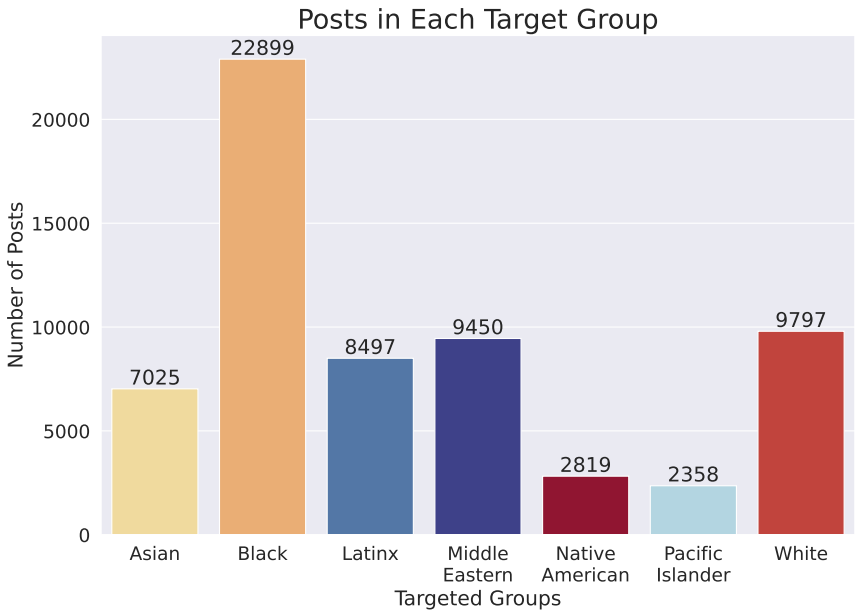


Figure 7.1: Statistics of posts targeting various demographic groups in the DLab dataset [70]. The Black community is the statistical majority in the dataset, emphasizing that they are most highly targeted in posts.

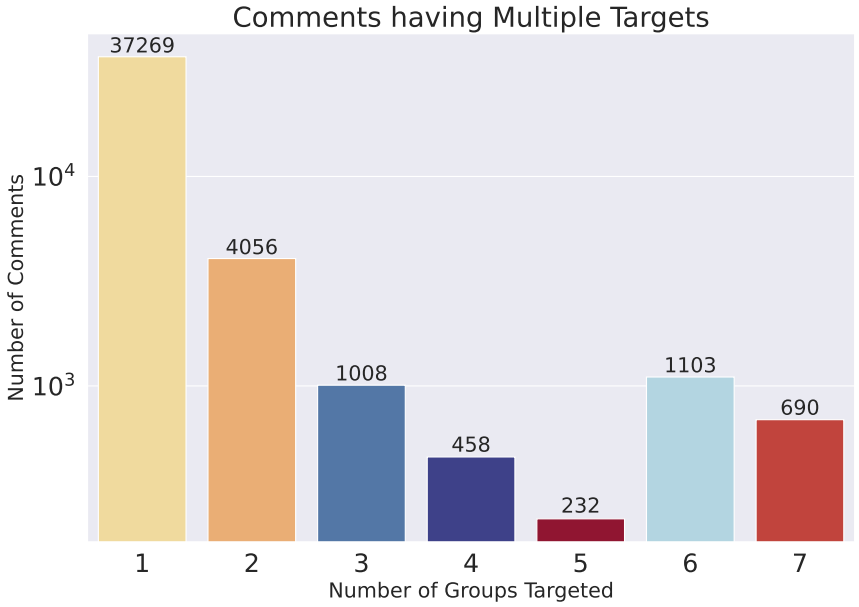


Figure 7.2: Statistics of posts targeting multiple groups.

7.4.1 Baselines Compared

We use the same loss functions as before from Section 6.3.3. Note that while CLA, EO are applicable to multi-group setting, ADV by design is only for two-group setting. In our implementation, we replace BCE with wBCE in Eqs. 6.10, 6.11 to make a fair comparison under label imbalance.

7.4.2 Evaluation Measures Considered

Balanced Accuracy (BA) Unlike standard accuracy, which can be misleading in the presence of label imbalance, BA provides a more reliable model assessment when dealing with imbalanced datasets. It computes the average accuracy of each label, thereby offering a balanced perspective to account for the unequal label distribution. By considering both the sensitivity (*TPR: true positive rate*) and specificity (*TNR: true negative rate*) of each label, BA effectively captures the model's ability to correctly classify instances across all labels, regardless of their prevalence.

$$BA = (TPR + TNR)/2.0 \tag{7.4}$$

Average Balanced Accuracy (Avg. BA). When optimizing accuracy across groups, we report the average over the group-specific BAs (known as macro-averaging) as a summary statistics. The Avg. BA (macro) treats each group equally, ensuring that the classifier's performance is evaluated in a balanced manner across all demographic groups.

$$Avg.\, BA = \frac{1}{G}\sum_{g=1}^{G} BA(g) \tag{7.5}$$

Hamming Loss It is a widely employed metric for assessing the performance of multi-label classifiers. Formally, for a dataset with N instances and G labels, the Hamming Loss quantifies the fraction of incorrectly predicted labels across all instances in the dataset, with $hamming(y_i(g), \hat{y}_i(g))$ as an indicator function of 1, if

the g-th label for instance i is incorrectly predicted, 0 otherwise. Specifically, it measures the average fraction of labels that are misclassified in comparison to the true label set. Hamming and Subset Accuracy Loss are comparable under small label cases [154], hence we just report HL.

$$Hamming = \frac{1}{NG} \sum_{i=1}^{N} \sum_{g=1}^{G} hamming(y_i(g), \hat{y}_i(g)) \tag{7.6}$$

Other measures of interest Although we are strictly optimizing for similar BA across groups, it is imperative to state that the gain in fairness does not come at a strict trade-off to other measures of interest like Precision, Recall, $F1$. Given our multi-label setup, we report the macro variants (average over group-specific numbers) of the mentioned measures.

$$Prc_{macro} = \frac{\sum_{g \in G} Prc(g)}{G} \tag{7.7}$$

$$Rec_{macro} = \frac{\sum_{g \in G} Rec(g)}{G} \tag{7.8}$$

$$F1_{macro} = \frac{2 \cdot Prc_{macro} \cdot Rec_{macro}}{Prc_{macro} + Rec_{macro}} \tag{7.9}$$

7.4.3 Evaluation and Loss Performance

The values presented in **Table 7.1** illustrate the achieved BA values across various groups during a single run for the test set, for the two baseline losses (OE and CLA) *vs.* our GAP. Notably, the Black group, constituting the statistical majority in the dataset as outlined in Kennedy et al. [70], demonstrates the maximum BA values for the three losses. We show the maximum difference (Max. Diff.) between the groups in **Table 7.1**, to highlight the performance gap when the optimization method does not align with the intended evaluation. Specifically, optimizing for overall error (OE) fails to account for variations in group performance, resulting in the highest difference values (Max. Diff. of 21.9). In contrast, both the GAP and CLA approaches incorporate considerations of group performance alongside overall error,

Balanced Accuracy (BA) (↑)									
Loss	Asian	Black	Latinx	Middle Eastern	Native American	Pacific Islander	White	**Max. Diff. (↓)**	**Avg. BA (↑)**
OE	80.31	**86.91**	81.07	**84.87**	64.99	67.91	75.01	21.9±1.3	77.29±0.29
CLA	82.51	85.34	81.67	84.62	73.91	74.92	80.44	11.4±0.8	80.49±0.14
GAP	**83.18**	83.86	**83.47**	83.42	**78.95**	**78.32**	**82.58**	**5.5**±0.5	**81.97**±0.13

Table 7.1: Balanced Accuracy (BA) achieved by each loss function (OE and CLA baselines *vs.* our GAP) over the 7 demographic groups (on test data) for one run. For each loss, we also color which group exhibits the maximum and minimum BA values achieved for that the loss over the 7 groups, with the difference between this maximum *vs.* minimum shown in the Max. Diff. column (for BA, we want to minimize this difference). GAP achieves a lower maximum difference (Max. Diff. = 5.54) than either baseline loss function, evident from the visualization in *Fig. 7.3*. GAP also achieves the highest (macro) average BA across groups (Avg. BA = 81.97). We also report the standard deviation of Max. Diff and Avg. BA over five runs, with CLA and GAP having similar deviation.

leading to substantially lower Max. Diff. compared to OE. Notably, given that GAP optimizes for balanced error rates across groups, it exhibits the smallest difference (Max. Diff. of 5.5), indicating least disparities across groups.

We report the average BA (macro) score obtained in all three losses in **Table 7.1**, with GAP having the highest value (Avg. BA = 81.97). We hypothesize that by incorporating the additional group information in the loss while model training, both the group-informed losses (CLA and GAP) are able to find better local optima compared to OE. The Best BA is a group performance measure indicating how many amongst the 7 groups are performing best across the three losses. For this measure, we observe OE to have the highest group BA values for 2 groups (*Black, Middle Eastern*), while GAP performs best for rest of the five groups. This also emphasizes the fact that GAP does not prioritize the performance of one group over others in its optimization criteria, thereby being the best performing loss across the groups.

To further highlight performance disparities between demographic groups in our target-group detection setting, we present **Fig. 7.4**. This figure displays the pairwise absolute differences in evaluated Balanced Accuracy (BA) across various demographic groups (|*left* - *bottom*|). Notably, as each group is equivalent to itself,

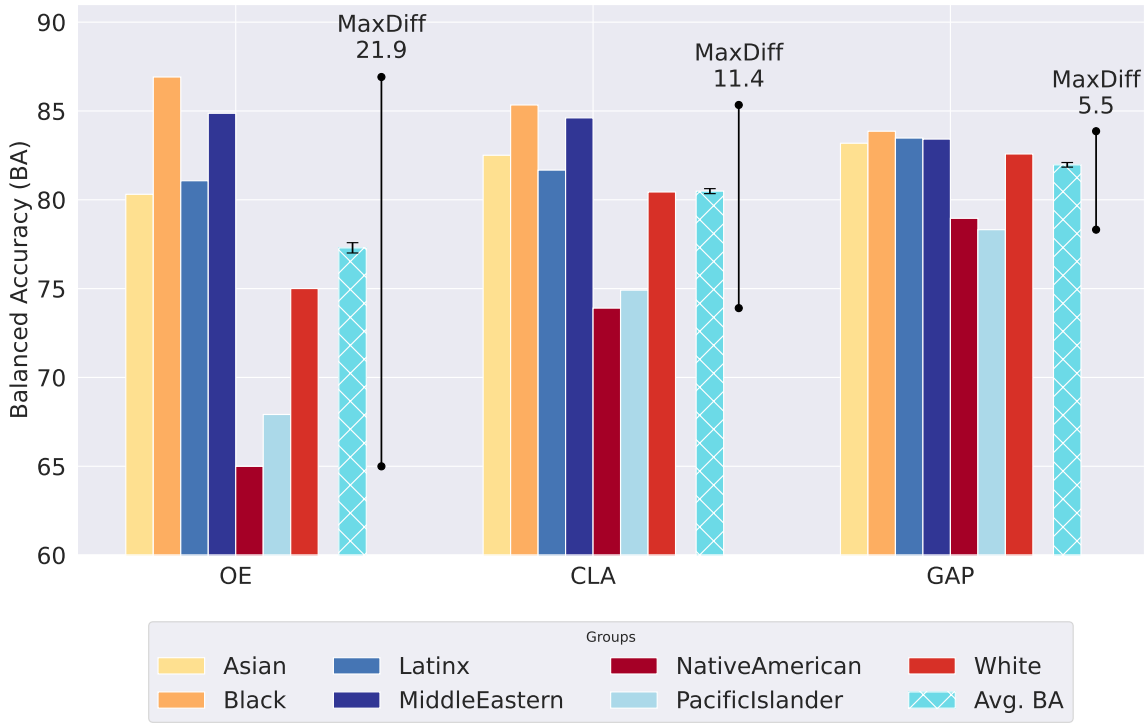


Figure 7.3: Visualization of the BA values achieved by each loss function over the 7 demographic groups. The maximum difference (Max. Diff.) between the maximum and minimum BA achieved for each loss across groups is also shown. See Table 7.1 for additional detail and discussion. GAP performs best with lowest Max Diff. of 5.5.

all diagonal entries are 0.0. Higher values in the heatmap indicate the classifier's bias towards one group compared to another. Through the color gradient in the heatmap, we observe consistent patterns of unequal group performance, particularly evident in the optimization for overall error rates (OE). This illustrates that solely optimizing for overall performance may result in disproportionate and inequitable performances across the internal groups within the dataset.

Apart from the (*Black, Native American*) pair which has a Max. Diff. of 21.9, we see other group pairs as well with a wide range of performance disparity, when group indicators aren't considered in OE. Two key observations can be drawn from the figure: a) The *Black* group being the statistical majority has a dominant performance gap over the *Native American* group being one of the statistical minorities. b)

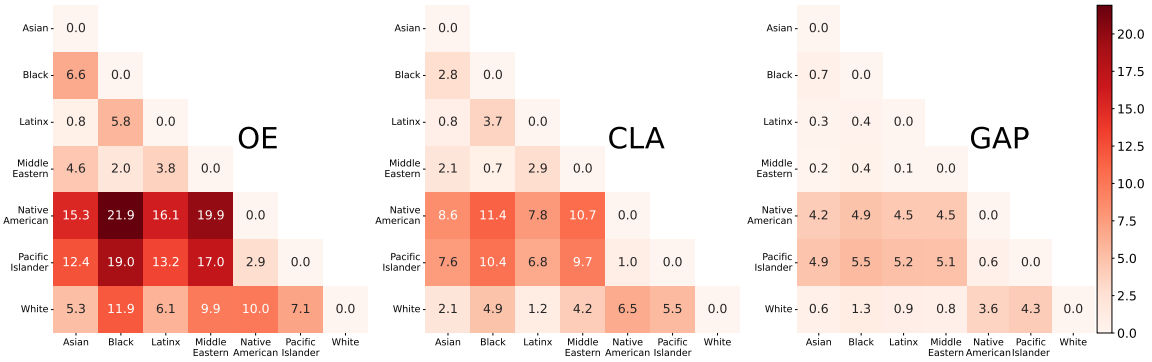


Figure 7.4: Heatmap of pairwise absolute difference of BA across groups in test set as an indicator for bias and disparate impact. OE has the highest performance gap (Max Diff = 21.9) across groups as indicated by the extremes of color, not only across one group- pair but consistently across multiple group pairs. GAP has the least spread in pairwise error values (Max Diff = 5.5), evident from the flatness of color, indicating least disparate impact across groups.

Even for group pairs with similar statistical population, for *e.g.,* the (*Middle Eastern, White*) pair, there can be performance disparities (BA gap of 9.9) because one group might be simply easier to classify than the other. CLA, which optimizes for FNR, has improved performance over OE, however, since it's optimization criteria (minimizing false negatives)does not align with out intended evaluation (having similar performance across groups) its heatmap lies in-between that of OE and GAP.

In contrast, our GAP loss, which is explicitly designed to achieve similar (balanced) performance across groups while optimizing overall performance, shows substantially fewer extremes in group performance gaps. The heatmap reveals smoother transitions between groups, indicating a more equitable distribution of the performance of the classifier. Moreover, the extreme values of Maximum Difference (Max. Diff.) presented in Table 7.1 are reflected as outliers in these heatmaps, corresponding to specific group pairs.

We present the Hamming Loss (**Eq. 7.6**) values in **Table 7.2** as a summary statistic in our multi-label classification setup. The Hamming loss metric quantifies the average fraction of misclassified labels, with lower values indicating enhanced clas-

	OE	CLA	GAP
Hamming Loss % (↓)	7.65	12.10	**6.85**

Table 7.2: HL values across different losses for test data. **Lower** values are **better**. GAP optimizes for jointly over TPR and TNR, thereby achieving lowest values, indicating better classifier performance.

sifier performance. While the CLA loss prioritizes minimizing FNR, which inherently involves asymmetry, there may be instances where it disproportionately optimizes for this aspect at the expense of other performance metrics. Consequently, CLA may exhibit poorer Hamming performance compared to OE loss. In contrast, the GAP loss maintains symmetry, resulting in a balanced trade-off while jointly minimizing TPR and TNR. As a result, GAP consistently achieves the lowest Hamming values, indicative of superior classifier performance.

Loss	Prc_{macro} (↑)	Rec_{macro} (↑)	$F1_{macro}$ (↑)
OE	0.7083	0.5808	0.6383
CLA	0.5418	**0.7143**	0.6162
GAP	**0.7854**	0.6837	**0.7310**

Table 7.3: Summary statistics of other evaluation measures. Since CLA strictly optimizes for minimizing FNR, it indeed achieves the highest Recall (1 - FNR). However, this comes at the cost of losing out on Precision. Since GAP jointly minimizes both TPR and TNR, it performs best both in terms of Precision and F1 scores across three losses.

Table 7.3 presents summary statistics of evaluation measures - Precision, Recall and F1 - in our multi-label classification at the macro level. The CLA approach, characterized by its emphasis on minimizing the False Negative Rate (FNR), inherently yields the highest Recall, however, this optimization strategy comes at the expense of Precision and F1, both being lower than OE. Our GAP loss, by jointly minimizing True Positive Rate (TPR) and True Negative Rate (TNR), emerges as the optimal performer in terms of Precision and F1 score across the evaluated losses.

7.4.4 Runtime Performance

The runtime and model convergence numbers shown in **Table 7.4** are over the training dataset of 36k posts, where we report the average time per epoch, epochs till convergence, total runtime, and the extra time (Δ) for losses compared to OE.

	Avg. Time Per Epoch (s)	Epochs till Convergence	Runtime Total (s)	Δ (s) *w.r.t.* OE
OE	154	21	3234	0
CLA	158	41	6478	3244
GAP	163	27	4401	1167

Table 7.4: Runtime Analysis. While OE takes the least time, GAP gains more in optimizing performance across groups for an extra of $\sim 9s$ per epoch. The smoothness of GAP loss (27 epoch) also allows faster convergence compared to CLA (41 epoch).

Since OE in Alg. 3 is weighted Binary Cross Entropy (wBCE), it takes the least amount of time per epoch and also number of epoch to converge. GAP in Alg. 4 takes all the steps of Alg. 3 for computing the overall loss in addition to calculating ${}^{G}C_{2}$ (G choose 2) losses and the balanced loss. Thereby GAP takes additional compute time for solving its intended optimization. The same argument holds true for CLA and ADV as well, since all of them are variants of the SOO format in Eq. 6.1.

Although GAP does the ${}^{G}C_{2}$ extra computation, hence the extra runtime ($\Delta = 1167s$), it is not that significant (extra 9s per epoch) compared to OE, while gaining much more in terms of optimization improvement. While OE and GAP operate on smooth losses (Eq. 6.6, 6.4) their convergence epoch is relatively fast ($\sim$21, $\sim$27). CLA uses a 1-norm loss (Eq. 6.10), hence the empirical loss surface is not as smooth as the previous two. As such, it is observable that CLA on average takes more epochs ($\sim$41) to converge with a higher Δ.

7.4.5 Analysis and Discussions

Differentiable Measures. While important fairness measures have been proposed in the literature, catering to evaluate different scenarios, many lack an equivalent differentiable loss, making these measures difficult to optimize. Model training with approximate loss functions might lead to potential metric divergence [98, 101] between optimization criteria used in training *vs.* evaluation measure of interest. "*No matter what measure is chosen for optimization, an inexact metric necessarily leads to a divergence between the goal and the metric in the tail.*" [92]. Continuing formulation of equivalent differentiable loss functions *w.r.t.* other important fairness measures could yield better performance for them.

Goodhart's Law and Over-optimization. *Goodhart's Law* states that "When a measure becomes a target, it ceases to be a good measure" [49]. Thomas and Uminsky [142] highlight how an over-reliance on metrics can lead to unintended consequences across AI systems, and fairness measures are no exception. Over-optimizing for any one metric in isolation risks degrading performance on others. For example, Table 7.3 shows that CLA's focus on minimizing FNR leads to large drops in Precision and F1, where it underperforms *w.r.t.* to both OE and GAP. As Friedler et al. [43] and others have noted, different worldviews lead to conflicting definitions of fairness that are mutually incompatible. Since one cannot have it all, specific fairness measures must be selected (suitable to the given task, context, and stakeholders). In this work, given the nature of the symmetric errors, we optimize a model to provide balanced Accuracy Parity (AP) across demographic groups [160] via our GAP loss function.

Balanced Measure *vs.* Overall Performance. As its name reflects, OE optimizes for cross-entropy (wBCE); it does not consider group (sub-population) performance. Overall accuracy will be driven by several factors. First, under-represented groups may suffer at the cost of benefiting the over-represented groups (*i.e.,* group

prevalance). Second, even when groups are balanced, some groups may be intrinsically more difficult to model for a given task, and thus sacrificed in optimization to benefit other groups. Since we train directly on the data (no over-/under- sampling), using the training objective to achieve AP across groups, we are able to accommodate both of the the cases above. Readers are referred to 2-group setting: *e.g., Asian vs. Black* and *White vs. Latinx* pairs.

Improved Overall Performance with Balancing. For the 7-group setting in Table 7.1, we notice that by optimizing for group-related errors alongside overall error (OE), both GAP and CLA driven classifiers achieve better overall performance (in terms of Avg. BA), as well as achieving their intended group-specific objective. This observation goes a bit beyond traditional ML where the nature of Single Objective Optimization (Eq. 6.1) forces one objective to be better at the expense of the other objective, in absence of any alternate dominated solution sets [94]. Given that our problem setup is Multi-Label classification (and correspondingly the architectural setup is a series of one-vs rest classifier nodes), we hypothesize that the group indicator gives an extra feature dimension for the classifier to consider, boosting it to learn something more about the data than it would have without the group label. By considering the group-associated terms, both loss functions have a modified surface compared to OE, allowing convergence to a better optima. We see this pattern emerging in some of the 2-group setting: *e.g.,* the *Latinx vs. Middle Eastern* group.

Model Multiplicity [11] highlights the ability of a task to have variability in the predictions generated by different models, although they perform with equal accuracy. Such effects arise due to factors like data imbalance, inherent biases *etc.* A simple case in our target-group detection setting could have three classifiers — trained with three losses (OE, CLA and our GAP) — performing equally well in terms of overall accuracy, yet differing widely in group-accuracy performance. In such a scenario, AP provides valuable insight as a fairness measure by highlighting the amount of

disparate impact across groups, with GAP having the least pairwise performance gap across groups, amongst the three losses.

We present the performance across all four losses (OE, CLA, ADV and our GAP) for some of the 2-group case to show different scenarios and performances. Best values are bolded (higher for BA, lower for Diff.). For each loss, we also color which group exhibits the maximum and minimum BA values achieved.

Balanced Accuracy (BA)				
Loss	Latinx	Middle Eastern	Avg. BA	Diff.
OE	90.98	83.67	87.33	7.31
CLA	91.52	88.73	90.12	2.79
ADV	91.04	84.20	87.62	6.84
GAP	**92.34**	**91.79**	**92.06**	**0.55**

Table 7.5: Optimizing GAP results in improving overall error, indicating the group label provides extra dimension for the loss to stabilize at a better local optima.

Balanced Accuracy (BA)				
Loss	Asian	Black	Avg. BA	Diff.
OE	86.59	**92.32**	89.46	5.73
CLA	87.02	92.22	89.62	5.20
ADV	87.49	92.10	89.79	4.61
GAP	**89.65**	90.82	**90.23**	**1.17**

Table 7.6: Performance varies across groups due to their population size, where the statistically major group dominates.

ADV [155] is an approximate adversarial loss for balancing FPR rates across groups. We notice similar issues of convergence instability as observed in Xia et al. [155]. Consequently, let ADV run for a fixed epochs and report the best BA value achieved over iterations.

Balanced Accuracy (BA)				
Loss	White	Latinx	Avg. BA	Diff.
OE	**88.89**	82.19	85.54	6.70
CLA	88.20	85.55	86.87	2.65
ADV	88.18	82.19	85.18	5.99
GAP	88.36	**86.23**	**87.29**	**2.13**

Table 7.7: Groups have similar population, but performance varies due to one group being more difficult to model.

Task Agnostic Measure. While in this work we explicitly focus around fair target-group detection, our proposed measure GAP is model, task, and dataset agnostic *i.e.,* it is designed to push for equal accuracy numbers across groups with arbitrarily defined groups. Thus, balancing between accuracy *vs.* fairness is not exclusive to the task of Toxic Language Detection, and can be extended to other problems, datasets, and models. It can also be used to for classifiers involving groups, sets or categories.

Exploring Fairness tasks with Symmetric Errors. By recognizing and incorporating symmetric error (*i.e.,* type I and type II errors are equally harmful) considerations into fairness tasks, we not only enhance the fairness of our target-group detection model but also open avenues to explore real-life scenarios addressing similar scenarios, challenges and needs in other domains. By acknowledging and addressing the symmetric nature of errors across groups, we can achieve a balanced perspective on fairness and more equitable outcomes in algorithmic decision-making processes.

Author Demographics *vs.* Target Demographics. While author demographics [12] focuses on identifying group tags about the post's author, identifying Target Demographics involves determining group tags of post when its directed towards specific groups or communities [37, 74]. In scenarios involving sensitive topics or potentially toxic language, such group identification becomes crucial. For example, a

post containing racially charged language maybe indicative of targeting a particular demographic group as a slur. However, the interpretation of such language may vary depending on the context of interaction. If both the author and the target belong to the same demographic, the use of such language may be considered as a friendly banter or colloquial communication within that group only. Conversely, if the author does not belong to the targeted group, the same language may be considered more likely as toxic, reflecting potential discriminatory behavior. Thus, both author and target demographics need to be considered jointly to combat toxic language.

Target group identification and Large Language Models. The task of fair target group detection can also be relevant in the context of training and deploying Large Language Models (LLMs). Existing studies have found that LLMs contain bias with respect to protected characteristics such as gender and race [76, 105]. Explicitly incorporating target group detection during training and fine-tuning via reinforcement learning from human feedback (RLHF) is a promising direction for reducing LLM bias.

Chapter 8: Conclusion

In this dissertation, I explored two core and parallel challenges in the domain of toxic language detection: (1) improving predictive performance and fairness in toxicity detection across multiple demographic groups using Multi-Task Learning (MTL); and (2) balancing competing objectives between a specific variant of group fairness and overall accuracy. While my application focus was exclusively on toxic language detection, the frameworks and methodologies developed here are more broadly applicable to other NLP tasks involving fairness, scalability and trade-offs in optimization.

By addressing the limitations of traditional approaches in capturing demographic specific patterns of toxicity and balancing competing fairness objectives, the research here lays the groundwork for more equitable and effective NLP systems. Our findings underscore the importance of considering both shared and unique aspects of toxic language across demographic groups, while also providing tools for navigating the complex trade-offs between accuracy and fairness. As toxic language detection continues to evolve as a field, the frameworks and insights presented in this dissertation offer a foundation for building more nuanced and fair NLP models. These contributions lay the groundwork for more robust, scalable, and fair NLP models that extend beyond toxic language detection to other domains with similar challenges.

8.1 Summary of Methodological Contributions

We provide a summarization of the methodological contributions. Although in this thesis, we explicitly focus on toxicity detection as the downstream task, our proposed methods are more general and can be applied to a wide variety of NLP tasks or similar domains in physical and medical sciences.

Framework 1: Multi-Task Learning for Group-Targeted Classification. We developed classification models where the expression of data varies significantly across groups, making a one-size-fits-all approach suboptimal. To address these challenges, we developed the Conditional Multi-Task Learning (CondMTL) framework, which leverages both shared and group-specific model layers. This design strikes a balance between generalization and specialization, enabling the model to perform well across multiple groups despite the variations in how toxicity manifests across them. Building on CondMTL, we proposed SAJS-MTL (Stakeholder-Aware Joint Scalable MTL), which extends the model to account for the joint interaction between different stakeholders. This framework captures both inter-group (across groups) and inter-group (within a group) disagreements, ensuring more nuanced predictions. Furthermore, SAJS-MTL is optimized for scalability, ensuring computational efficiency even as the cardinality of stakeholder group grows. These frameworks go beyond the conventional one-size-fits-all approaches by jointly modeling multiple tasks and groups while ensuring fairness, scalability, and predictive performance across diverse groups. The improved performance of these models compared to state-of-the-art (SoA) baselines highlights their potential for improving both the fairness and accuracy of toxicity detection. NLP applications where our MTL frameworks are applicable include Sentiment Analysis [63], Fake News [79], Misinformation [78], Content Moderation [136], Healthcare [64] *etc.* Physical, chemical and medical sciences domains like Climate Modeling [116], Material Design [108], Drug Discovery [82], Structural Engineering [149], Energy Systems Optimization [141], Biomedical Imaging and Disease Detection [87], Robotics and Control Systems [156] *etc.* can also gain advantage from such group-specific modeling along with joint interaction of stakeholders.

Framework 2: Balancing Fairness and Accuracy through Multi-Objective Optimization. The second problem tackled in this work focuses on fairness measures and Pareto trade-off via Multi Objective Optimization. We developed a differentiable variant of the Accuracy Parity fairness measure called Group Accuracy

Parity (GAP), which can be used as a loss function to train a descent-based model to optimize for balanced accuracy across groups. To trade-off between competing objectives, we introduced the HNPF (Hypernetwork-based Multi-Objective Optimization Framework), which enables a model to explore the trade-off space during training, offering flexibility for users to adjust trade-offs at runtime. We further extended this framework to SUHNPF, a scalable variant capable of handling large-scale neural models. Our GAP fairness measure is applicable in various recommendation systems involving groups [96]. The SUHNPF hypernetwork is highly generalizable and can be applied in various domains such as economics [121], healthcare [73], finance [102], criminal justice [88] *etc.* Through scalable optimization, these methods provide practitioners with the flexibility to adapt trade-offs based on the specific needs and constraints, helping to build more equitable and accountable AI systems.

8.2 Summary of Domain Contributions

The proposed work advances the field of toxicity detection by addressing key challenges related to group-specific modeling, predictive performance, fairness and scalability. Traditional toxicity detection models often treat all demographic groups similarly, which overlooks the fact that toxicity manifests differently across groups. Additionally, such models are prone to bias when trained on datasets with skewed demographic distributions, leading to performance degradation for minority groups. This dissertation makes several important contributions to mitigate these challenges:

1. Group-Specific Modeling through Multi-Task Learning (MTL). The development of the Conditional Multi-Task Learning (CondMTL) framework allows for simultaneous learning of both shared and group-specific features of toxic language. This framework enables the model to generalize across diverse demographic groups while capturing the unique ways toxicity is expressed toward specific target groups. This adaptive modeling ensures that groups with distinct patterns of toxic

language are not overlooked, resulting in more nuanced and accurate predictions. This contribution addresses a major gap in previous models that applied a one-size-fits-all approach, improving both fairness and predictive performance. The proposed conditional labeling schema establishes an accurate way of labeling group-targeted examples without causing unintended label bias, resulting in accurate modeling.

2. Stakeholder-Aware Joint Modeling for Real-World Applications. The extension of CondMTL to SAJ-MTL (Stakeholder-Aware Joint MTL) incorporates the interactions between different stakeholders (*e.g.,* annotators and targets) into the model. This framework accounts for both inter-group and intra-group disagreements, which is essential for real-world platforms where multiple communities coexist. Furthermore, SAJ-MTL is optimized for scalability, ensuring that the model can handle a large and growing number of demographic groups without sacrificing computational efficiency. This makes the proposed work applicable to dynamic, large-scale systems such as social media platforms and online content moderation tools.

3. Balancing Fairness and Accuracy with Group Accuracy Parity (GAP). A contribution of this work is the introduction of the Group Accuracy Parity (GAP) measure, which ensures Accuracy Parity across demographic groups by balancing model performance evenly across all groups. Unlike conventional fairness metrics, GAP addresses the challenge of symmetric error costs, ensuring that misclassifications between any two demographic groups (*e.g., Black vs. Latinx*) are treated equally. The GAP measures provides a practical loss function for enforcing group fairness in classification while addressing the biases inherent in skewed datasets.

4. Multi-Objective Optimization with HNPF and SUHNPF. To balance between competing objectives, we introduce the HNPF framework, which allows the model to learn and explore the trade-off space between fairness and accuracy during training, providing users with the flexibility to adjust these trade-offs in real-time

based on evolving needs. The scalable extension, SUHNPF, ensures that the optimization framework can handle large neural models and datasets with multiple demographic groups, making it suitable for high-throughput content moderation systems.

5. Theoretical Insights into Fairness Metrics. A key theoretical contribution of this work is the discovery of an impossibility theorem between Accuracy Parity and Equalized Odds, addressing a common misconception that achieving Equalized Odds will automatically ensure satisfying Accuracy Parity. This insight highlights the inherent trade-offs between fairness measures and helps practitioners choose appropriate metrics based on their specific context and goals. This contribution ensures that future work in toxicity detection is better aligned with real-world fairness constraints.

6. Impact on Platform Moderation and Fair Content Delivery. Our proposed frameworks offer significant improvements for platforms engaged in content moderation and automated toxicity detection. By incorporating group-specific patterns, stakeholder interactions, and scalable fairness mechanisms, the models developed in this work help ensure fairer content moderation across diverse demographic groups. This work equips platforms with the tools needed to balance fair treatment across communities while maintaining sufficient accuracy for majority groups, ensuring that both minority and dominant communities are served equitably.

8.3 Remaining Challenges and Open Questions

8.3.1 Modeling Annotator Labels as Confidence Scores

A promising avenue for future work involves treating annotator labels as confidence scores rather than fixed binary labels. In toxicity detection tasks, annotators may often express varying degrees of certainty in assigning labels, as they are not always fully confident about categorizing content as definitively toxic or non-toxic [111, 134]. By modeling annotator labels as soft labels or probabilistic confidence

scores, the model can better capture the inherent uncertainty and subjectivity of these annotations, thereby learning from the variability across annotators [114]. This strategy could yield more robust predictions by accounting for cases where content might be perceived as ambiguously toxic or non-toxic, depending on the annotator's perspective. Techniques such as label smoothing, probabilistic modeling, or fuzzy labels could be applied to operationalize this concept [14]. Moreover, incorporating annotator confidence scores could also help mitigate biases, particularly when some annotators are more sensitive or lenient in their judgments, by adjusting the weight of their contributions during model training [119].

8.3.2 Modeling Target Labels as Confidence Scores

Another important and parallel direction is modeling target group(s) as confidence scores, rather than treating them as fixed, binary attributes. In toxicity detection tasks, target group(s) identity (*e.g.,* race, gender, religion) are often inferred automatically (through a noisy oracle) or provided based on imperfect annotations (inference based on content of post), which can introduce uncertainty and noise [28]. By representing these identity scores as soft labels or probabilistic confidence scores, the model can better reflect the uncertainty in identifying the true target of a toxic comment. This approach would enable the system to assign varying levels of confidence to each potential target identity, rather than forcing a hard decision. Such probabilistic modeling could also improve the robustness of the toxicity detection pipeline by allowing the model to account for cases where the target identity is ambiguous or unclear, *e.g.,* a comment may seem toxic towards multiple groups, or its target might not be explicitly stated [129].

8.3.3 Graphical Modeling of the MTL Pipeline

While Multi Task Learning (MTL) frameworks are effective for capturing task relationships in toxicity detection, explicitly modeling the dependencies and uncer-

tainties between these tasks using probabilistic graphical models could significantly enhance performance [75]. By leveraging graphical models such as Bayesian networks or Conditional Random Fields, we can capture the joint probabilistic dependencies between the toxic language detection task and the target identity detection task, allowing the model to reason over both tasks simultaneously.

Incorporating graphical models into the pipeline would enable the system to explicitly represent uncertainties at various stages, such as in the prediction of target identities or the classification of group-conditioned toxic posts. This approach is especially important in cases where target identity detection itself is a noisy process. For example, a comment may target multiple groups, or the target might be inferred with limited confidence. Jointly modeling these tasks in a probabilistic framework would allow the system to propagate uncertainty from the target detection pipeline to the toxicity detection pipeline, making the toxicity predictions more robust.

Furthermore, graphical modeling enables more efficient inference and decision-making, particularly when the pipelines operate in parallel. By structuring the dependencies between tasks, the modeler can prioritize certain tasks or weight the output of the target detection pipeline based on the certainty of its predictions. Techniques such as variational inference or Markov Chain Monte Carlo could be applied to make this system scalable to large datasets [103]. This modeling approach would be especially valuable in real-world scenarios where both toxicity and target identification are prone to errors, helping to build a more accurate and fair toxicity detection system.

For instance, one can design a Graphical Model for the Multi-Task Learning framework by treating the input features, shared representations, and task-specific outputs as random variables with associated probability distributions. Let's define the prior distributions, likelihood functions, and posterior distributions as follows:

- Input Features (X): Feature vector of the tweet, with prior distribution $P(X)$ as a multivariate distribution, $P(X) \sim \mathcal{N}(\mu_X, \Sigma_X)$, where μ_X and Σ_X are the mean vector and covariance matrix of the input features.

- Shared Latent Variable (Z): Shared latent representation derived from the input features X, with prior distribution $P(Z)$, where Z is a non-linear transformation (ϕ) with additive Gaussian noise as $Z = W\phi(X) + \epsilon_z$, where W is a weight matrix and $\epsilon_Z \sim \mathcal{N}(0, \Sigma_Z)$. The conditional distribution is then $P(Z|X) \sim \mathcal{N}(W_X\phi(X) + b_X, \Sigma_Z)$, where W_X is the weight matrix mapping X to the latent space Z, b_X is the bias term, and the covariance matrix Σ_Z models the uncertainty in Z.

- Group-Specific Output Variables (Y_i): Binary output (toxic/non-toxic) for each demographic group G_i, with likelihood function $P(Y_i|Z)$, where Y_i can be modeled using a logistic regression $P(Y_i = 1|Z) = \sigma(W_i^T Z + b_i)$, where σ is sigmoid function, and W_i and b_i are the task-specific weight and bias parameters. The likelihood can be modeled as a Bernoulli distribution where Y_i represents the probability of the tweet being toxic towards group G_i. $Y_i \sim Bernoulli(p_i)$.

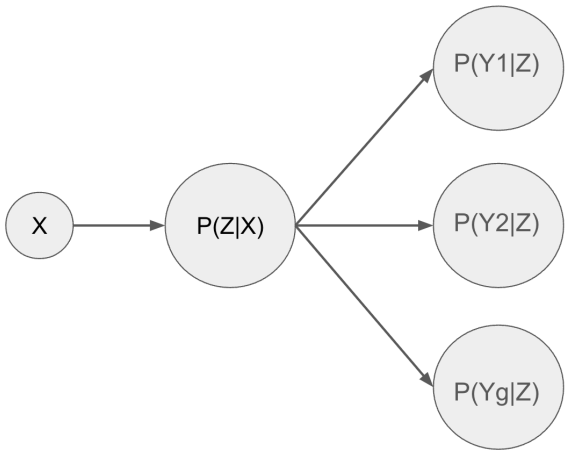


Figure 8.1: A simplistic Graphical model to represent the workings of the Conditional MTL model. X, Z, Y represents the input features, shared latent variable and group-specific output variables respectively.

The posterior distribution for each Task-Specific Outputs (Y_i) given the input X and the latent variable Z can be derived using Bayes' theorem:

$$P(Y_i|X) \propto P(Y_i|Z)P(Z|X)$$

To perform inference, we can estimate the posterior distribution $P(Y_i|X)$ for each group-specific output by sampling Z from its posterior distribution $P(Z|X)$ and computing the likelihood $P(Y_i|Z)$ for each task i, and finally aggregating results to obtain the posterior predictive distribution for each Y_i.

However, we need accompanying data to support simulations, that the proposed graphical modeling requires. For our given dataset, we have limited samples to infer any hyper-parameters or have a good estimate of the likelihoods and posteriors. Therefore, we keep the verification of this modeling approach as an open question.

8.3.4 Exploring Other Fairness Metrics and Constraints

Another important area is the exploration of alternative fairness metrics and constraints for toxicity detection. While Accuracy Parity and Equalized Odds were central to this dissertation, different fairness notions may be more appropriate depending on the use case or platform requirements. Toxic language detection operates in high-stakes settings, where balancing fairness across groups (*e.g.,* racial or gender identities) requires careful consideration of biases introduced by both model predictions and underlying datasets [100].

A promising research is individual fairness, which requires treating similar individuals similarly [33]. In the context of toxicity detection, this could involve ensuring that two comments with similar levels of toxicity (but directed at different groups) are treated consistently by the model. Implementing individual fairness would require similarity-based regularization constraints to align predictions for similar inputs, which could complement the group-based fairness metrics currently used.

Another alternative is Subgroup Fairness, which ensures fair treatment not just for large demographic groups (*e.g.,* 'race' or 'gender') but also for intersectional subgroups (*e.g.,* Black women, queer Latinx *etc.*). This is essential because intersectional groups may experience distinct forms of discrimination that are not captured by traditional group-level metrics [17]. Future work could extend the current frame-

works to intersectional multi-task learning architectures, ensuring fairness across both primary and intersectional subgroups.

8.3.5 Expansion to Multi-Lingual and Cross-Domain Toxicity Detection

A potential avenue is the expansion of the MTL framework to multi-lingual and cross-domain settings, where linguistic, cultural, and contextual variations influence the expression and perception of toxic language. Since toxicity is expressed differently across languages and domains, incorporating language-specific task branches along with domain-aware optimization objectives could further enhance the robustness and applicability of these models. Potential application areas include social media platforms, forums, or news outlets with wide demographic coverage.

In this context, language-agnostic word embeddings such as those from mBERT or XLM-R can be used to represent text, allowing shared knowledge transfer across languages [22]. The SAJS-MTL framework could be extended to include both target-group branches and language-specific branches, enabling simultaneous handling of demographic, community, and linguistic variations. In cross-domain settings, the type and prevalence of toxic language often vary depending on the platform or context (*e.g.,* Reddit vs. Twitter), and models trained on one platform tend to struggle on others due to domain shift [139]. This motivates the need for domain-aware learning, where the MTL framework could include domain-specific task branches while still using shared layers to capture general toxic patterns across platforms.

8.3.6 LLMs for Group Targeted Toxicity Detection

Large language models (LLMs) such as GPT-4 have demonstrated the ability to learn patterns and language use specific to different demographics, including subtle forms of toxicity. These models capture nuanced linguistic variations across different communities, making them promising tools for demographic-specific toxicity detection. By learning from diverse datasets, LLMs can detect how toxic language

manifests uniquely for various social groups, such as microaggressions toward women, slurs directed at ethnic minorities, or discriminatory remarks against religious groups.

However, the extent to which an LLM can effectively learn demographic-specific forms of toxicity depends heavily on the quality, diversity, and representativeness of the training data. If the training data lacks coverage for specific groups or contains disproportionate examples of toxicity for certain communities, the model may perform unevenly across demographics, thereby reinforcing systemic biases. Careful curation and balancing of datasets that reflect the experiences of marginalized groups, while minimizing harmful stereotypes is essential for building reliable models [8].

Ethical considerations must guide the development and deployment of LLMs for demographic-specific toxicity detection. Training models on data that contains toxic language, even for detection purposes, risks reinforcing or amplifying harmful stereotypes and biases [12]. As such, rigorous evaluation for fairness, bias, and unintended consequences is crucial. Model evaluation should consider it's ability to handle intersectional identities without propagating biased outputs. Moreover, explainability techniques can be integrated to ensure that stakeholders understand the decisions made by the model and can identify cases of bias or unintended harms [118].

Works Cited

[1] Gavin Abercrombie, Valerio Basile, Sara Tonelli, Verena Rieser, and Alexandra Uma. Proceedings of the 1st workshop on perspectivist approaches to nlp@ lrec2022. In *Proceedings of the 1st Workshop on Perspectivist Approaches to NLP@ LREC2022*, 2022.

[2] G Stoney Alder and Joseph Gilbert. Achieving ethics and fairness in hiring: Going beyond the law. *Journal of Business Ethics*, 68:449–464, 2006.

[3] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of Data and Analytics*, pages 254–264. Auerbach Publications, 2016.

[4] Aymé Arango, Jorge Pérez, and Barbara Poblete. Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 45–54, 2019.

[5] Ananth Balashankar, Alyssa Lees, Chris Welty, and Lakshminarayanan Subramanian. What is fair? exploring pareto-efficiency for fairness constrained classifiers. *arXiv preprint arXiv:1910.14120*, 2019.

[6] Valerio Basile, Federico Cabitza, Andrea Campagner, and Michael Fell. Toward a perspectivist turn in ground truthing for predictive computing. *arXiv preprint arXiv:2109.04270*, 2021.

[7] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: an extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. arxiv preprint. *arXiv preprint arXiv:1810.01943*, 2018.

[8] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

[9] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50(1):3–44, 2021.

[10] Federico Bianchi, Stefanie Anja Hills, Patricia Rossini, Dirk Hovy, Rebekah Tromble, and Nava Tintarev. " it's not just hate": A multi-dimensional perspective on detecting harmful speech online. *arXiv preprint arXiv:2210.15870*, 2022.

[11] Emily Black, Manish Raghavan, and Solon Barocas. Model multiplicity: Opportunities, concerns, and solutions. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 850–863, 2022.

[12] Su Lin Blodgett, Johnny Wei, and Brendan O'Connor. A dataset and classifier for recognizing social media english. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 56–61, 2017.

[13] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019.

[14] Abdelhamid Bouchachia and Witold Pedrycz. Enhancement of fuzzy clustering by mechanisms of partial supervision. *Fuzzy Sets and Systems*, 157(13):1733–1759, 2006.

[15] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[16] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE, 2010.

[17] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[18] US Census Bureau, July 2022. URL https://www.census.gov/quickfacts/fact/table/US/PST045222.

[19] R Caruana. Multitask learning: A knowledge-based source of inductive bias1. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Citeseer, 1993.

[20] Tong Chen, Danny Wang, Xurong Liang, Marten Risius, Gianluca Demartini, and Hongzhi Yin. Hate speech detection with generalizable target-aware fairness. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 365–375, 2024.

[21] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017.

[22] A Conneau. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

[23] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

[24] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, 8(3):631–657, 1998.

[25] Sanjiv Das, Michele Donini, Jason Gelman, Kevin Haas, Mila Hardt, Jared Katzman, Krishnaram Kenthapadi, Pedro Larroy, Pinar Yilmaz, and Muhammad Bilal Zafar. Fairness measures for machine learning in finance. *The Journal of Financial Data Science*, 3(4):33–64, 2021.

[26] Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10: 92–110, 2022.

[27] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2017.

[28] Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. Racial bias in hate speech and abusive language detection datasets. *arXiv preprint arXiv:1905.12516*, 2019.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[30] Mark Díaz, Ian Kivlichan, Rachel Rosen, Dylan Baker, Razvan Amironesei, Vinodkumar Prabhakaran, and Emily Denton. Crowdworksheets: Accounting for individual and collective identities underlying crowdsourced dataset annotation. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2342–2351, 2022.

[31] William Dieterich, Christina Mendoza, and Tim Brennan. Compas risk scales: Demonstrating accuracy equity and predictive parity. *Northpointe Inc*, 7(4), 2016.

[32] Saykat Dutta and Kedar Nath Das. A survey on pareto-based eas to solve multi-objective optimization problems. *Soft Computing for Problem Solving*, pages 807–820, 2019.

[33] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Conference on Innovations in Theoretical Computer Science (ITCS)*, 2012.

[34] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on fairness, accountability and transparency*, pages 119–133. PMLR, 2018.

[35] Matthias Ehrgott and Margaret M Wiecek. Saddle points and pareto points in multiple objective programming. *Journal of Global Optimization*, 32(1):11–33, 2005.

[36] Michael D Ekstrand, Anubrata Dass, Robin Burke, Fernando Diaz, et al. Fairness in information access systems. *Foundations and Trends® in Information Retrieval*, 16(1-2):1–177, 2022.

[37] Hugging Face. ucberkeley-dlab/measuring-hate-speech, 2022. URL https://huggingface.co/datasets/ucberkeley-dlab/measuring-hate-speech.

[38] Eve Fleisig, Rediet Abebe, and Dan Klein. When the majority is wrong: Modeling annotator disagreement for subjective tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6715–6726, 2023.

[39] Paula Fortuna and Sérgio Nunes. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30, 2018.

[40] Paula Fortuna, Juan Soler, and Leo Wanner. Toxic, hateful, offensive or abusive? what are we really classifying? an empirical analysis of hate speech datasets. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6786–6794, 2020.

[41] Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior, 2018. URL https://open.bu.edu/handle/2144/40119.

[42] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 329–338, 2019.

[43] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. The (im) possibility of fairness: Different value systems require different mechanisms for fair decision making. *Communications of the ACM*, 64(4):136–143, 2021.

[44] Björn Gambäck and Utpal Kumar Sikdar. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, pages 85–90, 2017.

[45] Ruoyuan Gao and Chirag Shah. How fair can we go: Detecting the boundaries of fairness optimization in information retrieval. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, 2019.

[46] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11543–11552, 2020.

[47] A Ghane-Kanafi and E Khorram. A new scalarization method for finding the efficient frontier in non-convex multi-objective problems. *Applied Mathematical Modelling*, 39(23-24):7483–7498, 2015.

[48] Massimiliano Gobbi, F Levi, Gianpiero Mastinu, and Giorgio Previati. On the analytical derivation of the pareto-optimal set with applications to structural design. *Structural and Multidisciplinary Optimization*, 51(3):645–657, 2015.

[49] Charles AE Goodhart. *Problems of monetary management: the UK experience*. Springer, 1984.

[50] Mitchell L Gordon, Michelle S Lam, Joon Sung Park, Kayur Patel, Jeff Hancock, Tatsunori Hashimoto, and Michael S Bernstein. Jury learning: Integrating dissenting voices into machine learning models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2022.

[51] Nitesh Goyal, Ian D Kivlichan, Rachel Rosen, and Lucy Vasserman. Is your toxicity my toxicity? exploring the impact of rater identity on toxicity annotation. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2): 1–28, 2022.

[52] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *International conference on artificial neural networks*, pages 799–804. Springer, 2005.

[53] Soumyajit Gupta, Gurpreet Singh, Anubrata Das, and Matthew Lease. Pareto Solutions vs Dataset Optima: Concepts and Methods for Optimizing Competing Objectives with Constraints in Retrieval. In *Proceedings of the The 7th ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR)*, 2021.

[54] Soumyajit Gupta, Gurpreet Singh, Raghu Bollapragada, and Matthew Lease. Learning a neural pareto manifold extractor with constraints. In *Uncertainty in Artificial Intelligence*, pages 749–758. PMLR, 2022.

[55] Soumyajit Gupta, Gurpreet Singh, Raghu Bollapragada, and Matthew Lease. Learning a Neural Pareto Manifold Extractor with Constraints. In *Proceedings of the 38th International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2022.

[56] Soumyajit Gupta, Sooyong Lee, Maria De-Arteaga, and Matthew Lease. Same same, but different: Conditional multi-task learning for demographic-specific toxicity detection. In *Proceedings of the ACM Web Conference 2023*, pages 3689–3700, 2023.

[57] Soumyajit Gupta, Venelin Kovatchev, Maria De-Arteaga, and Matthew Lease. Fairly accurate: Optimizing accuracy parity in fair target-group detection. *arXiv preprint arXiv:2407.11933*, 2024.

[58] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=rkpACe1lx.

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

[60] Xiao He, Francesco Alesiani, and Ammar Shaker. Efficient and scalable multi-task regression on massive number of tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3763–3770, 2019.

[61] Hoda Heidari, Michele Loi, Krishna P Gummadi, and Andreas Krause. A moral framework for understanding fair ml through economic models of equality of

opportunity. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 181–190, 2019.

[62] Danula Hettiachchi, Indigo Holcombe-James, Stephanie Livingstone, Anjalee de Silva, Matthew Lease, Flora D. Salim, and Mark Sanderson. How Crowd Worker Factors Influence Subjective Annotations: A Study of Tagging Misogynistic Hate Speech in Tweets. In *Proceedings of the 11th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, pages 38–50, 2023.

[63] Dirk Hovy. Demographic factors improve classification performance. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (Volume 1: Long papers)*, pages 752–762, 2015.

[64] Ming-En Hsieh and Vincent Tseng. Boosting multi-task learning through combination of task labels-with applications in ecg phenotyping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7771–7779, 2021.

[65] Guanlan Hu, Mavra Ahmed, and Mary R L'Abbé. Natural language processing and machine learning approaches for food categorization and nutrition quality prediction compared with traditional methods. *The American Journal of Clinical Nutrition*, 117(3):553–563, 2023.

[66] Lily Hu and Yiling Chen. Welfare and distributional impacts of fair classification. *arXiv preprint arXiv:1807.01134*, 2018.

[67] John E Hunter, Frank L Schmidt, and Ronda Hunter. Differential validity of employment tests by race: A comprehensive review and analysis. *Psychological Bulletin*, 86(4):721, 1979.

[68] Nan-Jiang Jiang and Marie-Catherine de Marneffe. Investigating reasons for disagreement in natural language inference. *Transactions of the Association for Computational Linguistics*, 10:1357–1374, 2022.

[69] Prashant Kapil and Asif Ekbal. A deep neural network based multi-task learning approach to hate speech detection. *Knowledge-Based Systems*, 210:106458, 2020.

[70] Chris J Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*, 2020.

[71] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Algorithmic fairness. In *AEA P&P*, 2018.

[72] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Algorithmic fairness. In *Aea papers and proceedings*, volume 108, pages 22–27. American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203, 2018.

[73] Marek Kochańczyk and Tomasz Lipniacki. Pareto-based evaluation of national responses to covid-19 pandemic shows that saving lives and protecting economy are non-trade-off objectives. *Scientific reports*, 11(1):2425, 2021.

[74] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[75] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[76] Hadas Kotek, Rikker Dockum, and David Sun. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Collective Intelligence Conference*, CI '23, page 12–24, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701139. doi: 10.1145/3582269.3615599. URL https://doi.org/10.1145/3582269.3615599.

[77] Deepak Kumar, Patrick Gage Kelley, Sunny Consolvo, Joshua Mason, Elie Bursztein, Zakir Durumeric, Kurt Thomas, and Michael Bailey. Designing toxic content classification for a diversity of perspectives. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*, pages 299–318, 2021.

[78] Rina Kumari, Nischal Ashok, Tirthankar Ghosal, and Asif Ekbal. Misinformation detection using multitask learning with mutual learning for novelty detection and emotion recognition. *Information Processing & Management*, 58(5): 102631, 2021.

[79] Rina Kumari, Nischal Ashok, Tirthankar Ghosal, and Asif Ekbal. A multitask learning approach for fake news detection: Novelty, emotion, and sentiment lend a helping hand. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[80] Francesco Levi and Massimiliano Gobbi. An application of analytical multiobjective optimization to truss structures. In *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 6975, 2006.

[81] Qing Liao, Heyan Chai, Hao Han, Xiang Zhang, Xuan Wang, Wen Xia, and Ye Ding. An integrated multi-task model for fake news detection. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5154–5165, 2021.

[82] Shaofu Lin, Chengyu Shi, and Jianhui Chen. Generalizeddta: combining pretraining and multi-task learning to predict drug-target binding affinity for unknown drug discovery. *BMC bioinformatics*, 23(1):367, 2022.

[83] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[84] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in neural information processing systems*, 32: 12060–12070, 2019.

[85] Zachary Lipton, Julian McAuley, and Alexandra Chouldechova. Does mitigating ml's impact disparity require treatment disparity? *Advances in neural information processing systems*, 31, 2018.

[86] Han Liu, Pete Burnap, Wafa Alorainy, and Matthew L Williams. Fuzzy multi-task learning for hate speech type identification. In *The world wide web conference*, pages 3006–3012, 2019.

[87] Mingxia Liu, Jun Zhang, Ehsan Adeli, and Dinggang Shen. Joint classification and regression via deep multi-task multi-channel learning for alzheimer's disease diagnosis. *IEEE Transactions on Biomedical Engineering*, 66(5):1195–1206, 2018.

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

[89] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumor and stance jointly by neural multi-task learning. In *Companion proceedings of the the web conference 2018*, pages 585–593, 2018.

[90] Pingchuan Ma, Tao Du, and Wojciech Matusik. Efficient continuous pareto exploration in multi-task learning. In *International Conference on Machine Learning (ICML)*, pages 6522–6531, 2020.

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

[92] David Manheim and Scott Garrabrant. Categorizing variants of goodhart's law. *arXiv preprint arXiv:1803.04585*, 2018.

[93] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *International Conference on Machine Learning*, pages 23803–23828. PMLR, 2023.

[94] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6): 369–395, 2004.

[95] Natalia Martinez, Martin Bertran, and Guillermo Sapiro. Minimax pareto fairness: A multi objective perspective. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[96] Judith Masthoff. Group recommender systems: Combining individual models. In *Recommender systems handbook*, pages 677–702. Springer, 2010.

[97] Achille Messac, Amir Ismail-Yahaya, and Christopher A Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and multidisciplinary optimization*, 25(2):86–98, 2003.

[98] Donald Metzler and W.B̃ruce Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[99] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3994–4003, 2016.

[100] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. Algorithmic fairness: Choices, assumptions, and definitions. *Annual Review of Statistics and Its Application*, 8:141–163, 2021.

[101] William Morgan, Warren Greiff, and John Henderson. Direct maximization of average precision by hill-climbing, with a comparison to a maximum entropy approach. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 93–96, 2004.

[102] Amitabha Mukerjee, Rita Biswas, Kalyanmoy Deb, and Amrit P Mathur. Multi–objective evolutionary algorithms for the risk–return trade–off in bank loan management. *International Transactions in operational research*, 9(5):583–597, 2002.

[103] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[104] Arvind Narayanan. 21 fairness definitions and their politics: A tutorial. In *Proceedings of the ACM FAccT Conference on Fairness, Accountability and Transparency*, New York, NY, USA, 2018. Association for Computing Machinery. https://shubhamjain0594.github.io/post/tlds-arvind-fairness-definitions/.

[105] Roberto Navigli, Simone Conia, and Björn Ross. Biases in large language models: origins, inventory, and discussion. *ACM Journal of Data and Information Quality*, 15(2):1–21, 2023.

[106] Aviv Navon, Aviv Shamsian, Gal Chechik, and Ethan Fetaya. Learning the pareto front with hypernetworks. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=NjF772F4ZZR.

[107] Harrie Oosterhuis and Maarten de Rijke. Differentiable unbiased online learning to rank. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1293–1302, 2018.

[108] Runhai Ouyang, Emre Ahmetcik, Christian Carbogno, Matthias Scheffler, and Luca M Ghiringhelli. Simultaneous learning of several materials properties from incomplete databases with multi-task sisso. *Journal of Physics: Materials*, 2 (2):024002, 2019.

[109] Vilfredo Pareto. Manuale di economica politica, societa editrice libraria. milan. *English translation as Manual of Political Economy, Kelley, New York*, 1906.

[110] Ji Ho Park, Jamin Shin, and Pascale Fung. Reducing gender bias in abusive language detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2799–2804, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1302. URL https://aclanthology.org/D18-1302.

[111] Ellie Pavlick and Tom Kwiatkowski. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7: 677–694, 2019.

[112] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[113] Behzad Pirouz and Esmaile Khorram. A computational approach based on the ε-constraint method in multi-objective optimization problems. *Advances and Applications in Statistics*, 49(6):453, 2016.

[114] Barbara Plank, Dirk Hovy, Anders Sogaard, et al. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of EACL*. Association for Computational Linguistics, 2014.

[115] Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, 55(2):477–523, 2021.

[116] Minghui Qiu, Peilin Zhao, Ke Zhang, Jun Huang, Xing Shi, Xiaoguang Wang, and Wei Chu. A short-term rainfall prediction model using multi-task convolutional neural networks. In *2017 IEEE international conference on data mining (ICDM)*, pages 395–404. IEEE, 2017.

[117] Md Mustafizur Rahman, Dinesh Balakrishnan, Dhiraj Murthy, Mucahid Kutlu, and Matthew Lease. An Information Retrieval Approach to Building Datasets for Hate Speech Detection. In *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS): Datasets and Benchmarks Track*, 2021.

[118] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. Closing the ai accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 33–44, 2020.

[119] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[120] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. Fairbatch: Batch selection for model fairness. In *9th International Conference on Learning Representations*, 2021.

[121] Peter Roosen, Stefan Uhlenbruck, and Klaus Lucas. Pareto optimization of a combined cycle power system as a decision support tool for trading off investment vs. operating costs. *International Journal of Thermal Sciences*, 42(6): 553–560, 2003.

[122] Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. HateCheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.4. URL https://aclanthology.org/2021.acl-long.4.

[123] Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, Janet Pierrehumbert, et al. Hatecheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58. Association for Computational Linguistics, 2021.

[124] Jennifer D Rubin, Lindsay Blackwell, and Terri D Conley. Fragile masculinity: Men, gender, and online harassment. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–14, 2020.

[125] Pratik Sachdeva, Renata Barreto, Geoff Bacon, Alexander Sahn, Claudia Von Vacano, and Chris Kennedy. The measuring hate speech corpus: Leveraging rasch measurement theory for data perspectivism. In *Proceedings of the 1st Workshop on Perspectivist Approaches to NLP@ LREC2022*, pages 83–94, 2022.

[126] Pratik S Sachdeva, Renata Barreto, Claudia von Vacano, and Chris J Kennedy. Assessing annotator identity sensitivity via item response theory: A case study

in a hate speech corpus. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1585–1603, 2022.

[127] Marta Sandri, Elisa Leonardelli, Sara Tonelli, and Elisabetta Ježek. Why don't you do it right? analysing annotators' disagreement in subjective tasks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2428–2441, 2023.

[128] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[129] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. The risk of racial bias in hate speech detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1668–1678, 2019.

[130] Anna Schmidt and Michael Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media*, pages 1–10, 2017.

[131] Anna Schmidt and Michael Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media*, pages 1–10, 2017.

[132] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, pages 525–536, 2018.

[133] Aili Shen, Xudong Han, Trevor Cohn, Timothy Baldwin, and Lea Frermann. Optimising equal opportunity fairness in model training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4073–4084, Seattle, United States, July 2022. Association for Computational Linguistics.

doi: 10.18653/v1/2022.naacl-main.299. URL https://aclanthology.org/2022.naacl-main.299.

[134] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622, 2008.

[135] Gurpreet Singh, Soumyajit Gupta, Matthew Lease, and Clint Dawson. A hybrid 2-stage neural optimization for pareto front extraction. *arXiv preprint arXiv:2101.11684*, 2021.

[136] Donghyun Son, Byounggyu Lew, Kwanghee Choi, Yongsu Baek, Seungwoo Choi, Beomjun Shin, Sungjoo Ha, and Buru Chang. Reliable decision from multiple subtasks through threshold optimization: Content moderation in the wild. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 285–293, 2023.

[137] Timo Spinde, Jan-David Krieger, Terry Ruas, Jelena Mitrović, Franz Götz-Hahn, Akiko Aizawa, and Bela Gipp. Exploiting transformer-based multitask learning for the detection of media bias in news articles. In *Information for a Better World: Shaping the Global Future: 17th International Conference, iConference 2022, Virtual Event, February 28–March 4, 2022, Proceedings, Part I*, pages 225–235. Springer, 2022.

[138] Gjorgji Strezoski, Nanne van Noord, and Marcel Worring. Many task learning with task routing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1375–1384, 2019.

[139] Steve Durairaj Swamy, Anupam Jamatia, and Björn Gambäck. Studying generalisability across abusive language detection datasets. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*, pages 940–950, 2019.

[140] Robin Swezey, Aditya Grover, Bruno Charron, and Stefano Ermon. Pirank: Scalable learning to rank via differentiable sorting. *Advances in Neural Information Processing Systems*, 34:21644–21654, 2021.

[141] Zhongfu Tan, Gejirifu De, Menglu Li, Hongyu Lin, Shenbo Yang, Liling Huang, and Qinkun Tan. Combined electricity-heat-cooling-gas load forecasting model for integrated energy system based on multi-task learning and least square support vector machine. *Journal of cleaner production*, 248:119252, 2020.

[142] Rachel L Thomas and David Uminsky. Reliance on metrics is a fundamental challenge for ai. *Patterns*, 3(5):100476, 2022.

[143] Ameya Vaidya, Feng Mai, and Yue Ning. Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 683–693, 2020.

[144] Ana Valdivia, Javier Sánchez-Monedero, and Jorge Casillas. How fair can we go in machine learning? assessing the boundaries of accuracy and fairness. *International Journal of Intelligent Systems*, 36(4):1619–1643, 2021.

[145] M Van Rooyen, X Zhou, and Sanjo Zlobec. A saddle-point characterization of pareto optima. *Mathematical programming*, 67(1):77–88, 1994.

[146] Bertie Vidgen and Leon Derczynski. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *Plos one*, 15(12):e0243300, 2020.

[147] Emily A. Vogels. The state of online harassment. *Pew Res. Center, Washington, DC, USA, Tech. Rep*, 2021.

[148] Ruyuan Wan, Jaehyung Kim, and Dongyeop Kang. Everyone's voice matters: Quantifying annotation disagreement using demographic information. In *Pro-*

ceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 14523–14530, 2023.

[149] Zhenyu Wang, Jian Zhou, and Kang Peng. The potential of multi-task learning in cfdst design: Load-bearing capacity design with three mtl models. *Materials*, 17(9):1994, 2024.

[150] Zeerak Waseem. Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 138–142, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-5618. URL https://aclanthology.org/W16-5618.

[151] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-2013. URL https://aclanthology.org/N16-2013.

[152] Zeerak Waseem, James Thorne, and Joachim Bingel. Bridging the gaps: Multi task learning for domain transfer of hate speech detection. *Online harassment*, pages 29–55, 2018.

[153] Chris Wotton. 'same same, but different': The origins of thailand's tourist catchphrase, Jan 2019. URL https://theculturetrip.com/asia/thailand/articles/same-same-but-different-the-origins-of-thailands-tourist-catchphrase/.

[154] Guoqiang Wu and Jun Zhu. Multi-label classification: do hamming loss and subset accuracy really conflict with each other? *Advances in Neural Information Processing Systems*, 33:3130–3140, 2020.

[155] Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. Demoting racial bias in hate speech detection. In *Proceedings of the Eighth International Workshop on*

Natural Language Processing for Social Media, pages 7–14, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.socialnlp-1.2. URL https://aclanthology.org/2020.socialnlp-1.2.

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

[157] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278, 2007.

[158] Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*, 2020.

[159] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

[160] Han Zhao, Amanda Coston, Tameem Adel, and Geoffrey J Gordon. Conditional learning of fair representations. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[161] Peng Zhao, Shiyi Zhao, Xuyang Zhao, Huiting Liu, and Xia Ji. Partial multi-label learning based on sparse asymmetric label correlations. *Knowledge-Based Systems*, 245:108601, 2022.