Same Same, But Different: Conditional Multi-Task Learning for Demographic-Specific Toxicity Detection

Soumyajit Gupta, Sooyong Lee*
Dept. of Computer Science
The University of Texas at Austin
[smjtgupta,sooyonglee]@utexas.edu

Maria De-Arteaga
McCombs School of Business
The University of Texas at Austin
dearteaga@mccombs.utexas.edu

Matthew Lease
School of Information
The University of Texas at Austin
ml@utexas.edu

ABSTRACT

Algorithmic bias often arises as a result of *differential subgroup validity*, in which predictive relationships vary across groups. For example, in toxic language detection, comments targeting different demographic groups can vary markedly across groups. In such settings, trained models can be dominated by the relationships that best fit the majority group, leading to disparate performance. We propose framing toxicity detection as multi-task learning (MTL), allowing a model to specialize on the relationships that are relevant to each demographic group while also leveraging shared properties across groups. With toxicity detection, each task corresponds to identifying toxicity against a particular demographic group. However, traditional MTL requires labels for all tasks to be present for every data point. To address this, we propose **Cond**itional **MTL** (CondMTL), wherein only training examples relevant to the given demographic group are considered by the loss function. This lets us learn group specific representations in each branch which are not cross contaminated by irrelevant labels. Results on synthetic and real data show that using CondMTL improves predictive recall over various baselines in general and for the minority demographic group in particular, while having similar overall accuracy.

KEYWORDS

Multi-task learning, differential subgroup validity, conditional loss

ACM Reference Format:
Soumyajit Gupta, Sooyong Lee, Maria De-Arteaga, and Matthew Lease. 2023. *Same Same, But Different*: Conditional Multi-Task Learning for Demographic-Specific Toxicity Detection. In *Proceedings of the ACM Web Conference 2023 (WWW '23), May 1–5, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3543507.3583290

1 INTRODUCTION

In developing natural language processing (NLP) models to detect toxic language [1, 44, 49], we typically assume that toxic language manifests in similar forms across different targeted groups. For example, HateCheck [36] 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). For this reason, a "one-size-fits-all" modeling approach may yield sub-optimal performance and more specifically raise concerns of algorithmic fairness [1, 32, 43]. 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*" [53].

More formally, such heterogeneity of toxic language targeting different groups can be conceptually framed in terms of *differential subgroup validity* [17]: 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 [9]. Some methods, *e.g.*, adversarial fairness approaches [55], 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 [20, 25], a reason that is tightly linked to the presence of differential subgroup validity. Prior work [7] has shown that differential subgroup validity can be addressed by training models that learn group-specific idiosyncratic patterns, such as decoupled classifiers [10]. 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

*Soumyajit Gupta and Sooyong Lee contributed equally to this work.

WWW '23, May 1–5, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04.
https://doi.org/10.1145/3543507.3583290

can benefit from training across posts targeting multiple groups, while task-specific layers are trained only on posts that target each respective group. For *e.g.,* 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.

A key limitation of prior MTL work [26] is the assumption that each training point is labeled for all tasks. In our context, this means that a post targeting group A is labeled as non-toxic for group B. However, this risks contaminating group B data because toxic terms or phrases contained in such posts would then be associated with a non-toxic label. We refer to this as *label contamination.*

In this work, we propose a novel **Cond**itional **MTL** (CondMTL) approach that is capable of learning group specific representation of toxic language over sparse labels. Crucially, we abandon the traditional MTL assumption that each training instance is labeled for all tasks. Instead, instances labeled only for one task influence shared-layer training without contaminating training of task-specific layers for other tasks. We achieve this by introducing a task-specific, conditional representation of labels, combined with a novel conditional loss function for MTL training that is a selective variant of the widely-used binary cross-entropy loss.

To evaluate CondMTL for toxic language detection with differential subgroup validity, we use the *Civil Comments* [4] portion of the *Wilds* [21] dataset, which provides toxicity labels with target demographics. We center the potential harms affecting the targets and distinguish *misses* (*i.e.,* an undetected toxic post) *vs. false alarms* (*i.e.,* a non-toxic post that is erroneously flagged) because the potential harm of different types of errors is different. Recall, which is tightly linked to the algorithmic fairness measure of Equality of Opportunity [14], is thus a central measure of interest in our study. Our results show that CondMTL matches accuracy of baselines while boosting recall across demographic target groups. For CondMTL, we also observe improved Equality of Opportunity between the groups. Further analysis investigates how the group specific nature of our problem affects model predictions (Section 7). We also present analysis of model performance *vs.* a prior baseline (Section 6.3) and conduct additional synthetic experiments to verify model correctness (Appendix D). Finally, we show that CondMTL reduces time and space required *vs.* other MTL baselines.

Key contributions of our work are as follows:

- We propose a novel Conditional MTL framework that supports conditional labels in general and specifically addresses differential subgroup validity in the context of toxic language detection.
- Results show that CondMTL performs better than existing MTL variants across demographic groups considered.
- We provide theoretical and empirical justification with benchmarking scenarios for mathematically checking model validity.
- For reproducibility and adoption, we share our TensorFlow-based source code for CondMTL[1].

2 RELATED WORK

Multi-task learning (MTL) has widespread use in many applications of machine learning, from computer vision to natural language processing and from hate speech detection to media bias diagnosis. We present a focused overview of the existing MTL works as applicable to our study. For detailed analysis, readers are referred to surveys by Ruder [38], Crawshaw [8], and Vandenhende et al. [50].

MTL architectures broadly fall under two categories of a) hard; and b) soft parameter sharing of hidden layers. Hard parameter sharing shares the hidden encoder layers between all tasks while keeping several task-specific output layers. It is the most commonly used approach [18, 22, 28] to MTL and empirically reduces the risk of over-fitting [3]. Soft parameter sharing maintains several independent task-specific layers with either some form of regularization in loss [54] or introduces correlation units across tasks [30]. It can learn an optimal combination of shared and task-specific representations as shown in [11, 27, 39].

2.1 MTL in Computer vision

Works in Architecture. Long et al. [28] place matrix priors on the fully connected layers, allowing the MTL model to learn relationships between tasks. They rely on a pre-defined structure for sharing, which may be well-studied for computer vision problems but less adaptable for non-vision tasks. Lu et al. [29] follows a bottom-up approach starting with thin layers and dynamically widens them greedily during training by promoting grouping of similar tasks. This greedy grouping may inhibit the model from learning similarities between tasks. Cross Stitch Networks [30] maintain correlation units across tasks specific layers, called cross stitch units. These units use a linear combination of the output of the previous layers across tasks, allowing the units to learn appropriate weights if the tasks are correlated to each other.

Works in Loss Function. These methods do not alter the MTL architecture, but rather use modified loss functions. Kendall et al. [18] use a Gaussian likelihood with a task-dependant uncertainty inspired loss function that assigns relative weights to each task based on its difficulty. Chen et al. [5] constrains the task-specific gradients to be of similar magnitude, thereby forcing all tasks to learn at an equal trajectory. Guo et al. [13] assigns higher weights to difficult tasks, forcing the network to learn better values on these tasks. However, this requires manual parameter tuning. Liu et al. [27] aims to maintain equal trajectory in losses across tasks, irrespective of their difficulty, by readjusting weights across tasks after each gradient update. Tasks with slower drop rate are assigned more weights for the next update.

2.2 MTL in Natual Language Processing

In order to find better hierarchies across NLP tasks, methods are developed to account for losses, not only at the outermost layer of the model, but at intermediate layers as well. Søgaard and Goldberg [46] empirically show that using supervision losses at the earlier stages of an NLP pipeline improved model performance for tasks such as part-of-speech tagging and named entity recognition on a deep bi-directional RNN architecture. Hashimoto et al. [15] uses the aforementioned idea to apply losses at different levels of the hierarchical NLP pipeline. They propose a joint model for MTL with supervising losses at the word, syntactic, and semantic levels.

2.3 MTL in Hate Speech

Liu et al. [26] use a fuzzy MTL setup for hate speech detection to identify hate speech from single-labeled data. They employ a fuzzy

[1] **https://github.com/smjtgupta/CondMTL**

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* (see Section 4.1). Plaza-del Arco et al. [34] observe that invoking labels of sentiment, emotion, and target of hate speech jointly improves detection of hate speech. They employ a MTL model where each tweet has labels corresponding to the mentioned attributes, using transfer learning. Samghabadi et al. [41] develop a MTL model to jointly predict aggression and misogyny across datasets. Vaidya et al. [49] use a MTL model to jointly predict toxicity and identity of target. Morgan et al. [31] frame the identification of toxic, engaging, and fact-claiming comments on social media as a MTL problem. Awal et al. [2] jointly learns hate speech detection with sentiment classification as primary and target identification as secondary task, and employ a model with independent task-specific layers and a parallel shared layer. The output vectors from these layers are merged using a gate fusion mechanism, which is a linear combination unit between the shared and task specific vectors.

2.4 Applied MTL

Given the media's ability to shape individuals' actions and beliefs, prior work has sought to improve media bias detection to identify underlying biases and their influences. Spinde et al. [47] use a MTL model which learns to detect bias using task-specific layers associated with specific datasets, yielding performance that generally surpasses their baseline single-task methods. The proliferation of misinformation has driven prior work in automated fact-checking. Vasileva et al. [51] use a MTL model to learn from multiple fact-checking organizations simultaneously. Their MTL model yields sizeable improvements over their single-task learning baseline, indicating a benefit to jointly learning identification of fact-check worthy claims for multiple news sources. In the context of data annotation, collected labels may often be dominated by views of the majority. To address this, Gordon et al. [12] introduce jury learning to model individual labelers in the dataset and incorporate dissenting voices when forming juries of diverse demographics.

Variations in language across posts targeting different groups is at the core of our motivation. We aim to improve model performance and fairness by learning group-specific variations. As we explained in Section 1, this contrasts with approaches such as adversarial fairness, which aim to *prevent* a model from relying on group-specific variations. For instance, Huang and Paul [16] argue that there is no unified use of language across different demographics including gender, age, country and region. With the goal of improving generalization across groups, they propose an adversarial training approach; the adversarial branch predicts demographics of the document, while the main branch does standard text classification. In doing this, their objective is to learn patterns that generalize across groups. Meanwhile, our goal is to learn group-specific patterns, *e.g.,* the model should be able to recognize demeaning terms that are only used against one minority group.

3 PROBLEM STATEMENT

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 assume binary labels for this dataset as $\mathcal{Y} \in \mathbb{R}^{N}$, where each label ($y$) corresponding to a post (d) can be either Non-Toxic (0) or Toxic (1). Furthermore, we are given the K demographic groups pertaining to the targets of each post. Thus each post can be mapped to an overall (group-agnostic) toxicity label as well as multiple group-targeted toxicity labels as $d \rightarrow y$ and $d \rightarrow y_k, \forall k \in K$,where the overall label $y = \bigcup_K y_k$ considering the data to be toxic/non-toxic, irrespective of the group. Due to the nature of the K independent groups, we have the combined dataset $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \ldots \cup \mathcal{D}_K$ as the union of the demographic specific data points $\mathcal{D}_k \in \mathbb{R}^{N_k \times F}$.

If our objective is to optimize a certain performance metric, *i.e.,* minimizing Binary Cross Entropy (BCE), we can do it over: a) the entire dataset $\mathcal{D}$; or b) demographic group specific data subsets $\mathcal{D}_k, k \subseteq K$. The Single Task (STL) model has independent classifiers for each split $\mathcal{D}_k$, while the Multi Task Learning (MTL) variant has one joint classifier with K task-specific branches for each split $\mathcal{D}_k$.

4 CONDITIONAL MTL

Conditional loss is intuitive: for each demographic branch, we should compute error across toxic and non-toxic class labels only for examples that are relevant to that branch's demographic group.

4.1 Labeling Schema

Traditional MTL (TradMTL) approaches assume that each training point has labels for all tasks. A post targeting group Green, irrespective of toxicity label, is assumed to be non-toxic towards group Orange 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 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*. For extended illustration, refer to Appendix C.

Post	Traditional MTL Labels		Correct Labels	
	Green	Orange	Green	Orange
"I hate Green"	Toxic	Non-Toxic	Toxic	•
"I love Green"	Non-Toxic	Non-Toxic	Non-Toxic	•

Table 1: Label contamination occurs in a Traditional MTL label assignment when posts that target a given group (Green) are assumed to be non-toxic toward any other group (*e.g.,*Orange). Red denotes unsupported label assignments, while (•) correctly denotes that these posts do not contain a label wrt. the Orange target group.

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. To illustrate the reasoning for the schema, we show a series of example post templates and their corresponding labels in **Table 2**. Note that in the traditional labeling schema, as proposed in [26] 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, and compute the loss over the relevant examples only.

Hypothetical Post	TradMTL Label Men	TradMTL Label Women	CondMTL Label Men	CondMTL Label Women
"I hate everybody"	T	T	T	T
"I hate men"	T	**NT**	T	•
"I hate women"	**NT**	T	•	T
"I love everybody"	NT	NT	NT	NT
"I love men"	NT	**NT**	NT	•
"I love women"	**NT**	NT	•	NT

Table 2: CondMTL group-specific labels *vs.* TradMTL labels for some hypothetical posts. T and NT denote toxic and non-toxic labels, The label (t) denotes the toxicity t 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.

4.2 CondMTL Algorithm

We follow the same architecture as TradMTL (shown in Fig. 3), but instead of standard weighted Binary Cross Entropy (wBCE), we use a Conditional weighted BCE loss function. wBCE is a variation of BCE that re-weights the error for the different classes proportional to their inverse frequency in the data [24]. This strategy is available in popular packages like SkLearn [33] 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. For example, if groups correspond to a simplified version of gender, with two possible group labels, then we would have three tasks: T1) given a post, is it toxic?; T2) given a post, is it toxic towards men? and T3) given a post, is it toxic towards women? All the examples in $\mathcal{D}$ are passed through the network, where the T1 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 (T2 and T3) as well, CondMTL only allows back propagation for the relevant instances. For instance, only the N_1 examples of $\mathcal{D}_1$ that are targeted towards women demographics would be considered by the women branch.

The Conditional Loss Function CondMTL 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 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*).

When considering the template posts from **Table 2**, the TradMTL model with its labeling schema [26] would correctly backpropagate its losses for the all, men, and women branches for the example *I hate everybody*. Given that this post does target both the men and women groups and is toxic, the traditional label (T, T, T) is equivalent to the conditional label (T, T, T). However, the template post *I hate men* 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 women branch due to the women label in the traditional label (T, T, NT) being marked as non-toxic (NT). The traditional MTL model would erroneously learn that a post which is toxic towards men is nontoxic if it were targeted at women, ultimately confusing the model. In contrast, the CondMTL model avoids back-propagating the loss which may confuse the model by examining the demographic flag corresponding to the label (T, T, •) and using it to compute the loss only for the men branch.

5 EXPERIMENTAL DETAILS

We describe the dataset used for validation along with a stacked single task model baseline and other MTL models for comparison with CondMTL. Implementation details regarding setup, loss curves, and class balancing strategy are enumerated in **Appendix A**.

5.1 Data

Target Identity Dataset. To assess differential subgroup validity in toxic language detection, we focus on toxicity and gender [37, 52]. We use the *Civil Comments* [4] portion of *Wilds* [21]. 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.

We consider posts where either group (women: 22,149 and men: 15,305) or both groups (both: 11,134) are targeted. **Table 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.

	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: Statistics of the Wilds [21] 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.

5.2 Baselines

For a single task (**STL**) baseline, we use a DistilBERT [42] 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. 1**). For our experiments, we freeze the weights of the DistilBERT [42] 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.

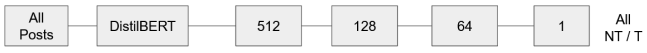


Figure 1: Architecture for Single Task, where all posts are passed though a neural network and get classified as toxic *vs.* non-toxic.

Stacked STL model (**Fig. 2**) 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.

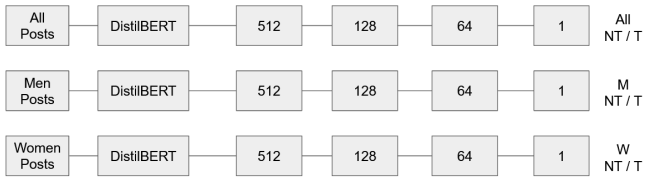


Figure 2: Architecture for stacked STL contains three independent single task models, one for each portion of the data.

Traditional Multi Task (TradMTL) model (**Fig. 3**) 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.

Cross Stitch Multi Task (CSMTL) model (**Fig. 4**) is similar to the stacked STL model (Fig. 2) with Cross Stitch (CS) units [30] 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.

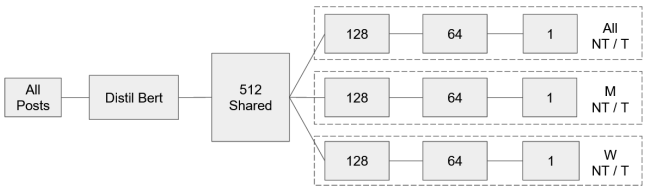


Figure 3: Architecture for TradMTL and CondMTL, where the 512 dense neurons are shared across all three tasks while maintaining independent task specific layers (mark by dashed boxes).

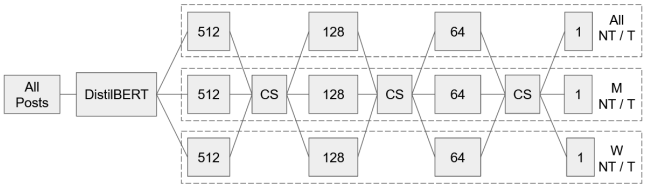


Figure 4: Architecture for CSMTL, which is replica of the Stacked STL model, with cross stitch (CS) units between each dense layers, allowing them to share weights across tasks for task similarity.

5.3 Inference

All models discussed (STL, TradMTL, CSMTL, and CondMTL) assume group labels at inference time. Each post in the *Civil Comments* [4] portion of *Wilds* [21] is human-annotated for the demographics, which includes the gender group of the target entity of each post. We acknowledge that human annotations may contain errors.

6 RESULTS

We present the performance of CondMTL *w.r.t.* the stacked STL baseline and other MTL variants in terms of trainable parameters and training runtime. We examine post hoc classification performance and recall disparities and compare confusion matrices for all of the models. We also present analysis of our CondMTL and CSMTL in terms of label contamination and contamination of weights.

6.1 Architecture and Runtime

Table 4 shows trainable parameters for the baseline models and for CondMTL. Results shown are only over one run. For mean and variance reports over multiple runs, please refer to **Appendix B**.

The single task model (Fig. 1) has 467,713 trainable parameters, hence the stacked STL (Fig. 2) operating on the All, Men, and Women portions of the data has 3× 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) 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. 4) 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

Model type	# Params	Δ	Time(s)	Δ
Stacked STL (3 models)	1,403,139	-	7,200	-
CSMTL [30]	1,403,166	+0%	2,600	-64%
TradMTL	615,683	-56%	2,200	-69%
CondMTL (Ours)	615,683	-56%	2,050	-72%

Table 4: 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.

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 [48].

We report the training runtime in Table 4 *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.

6.2 Performance Measures

We show the performance comparison of the models on the Wilds-Civil Comments [21] test dataset. The Accuracy numbers in **Appendix B: Table 7** indicate that all of the models roughly perform the same in terms of overall accuracy *w.r.t.* the Stacked STL (∼ 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.

In order to identify the discrepancies between the models, we compare the Recall, F1 and Precision scores in **Table 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 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

		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 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.

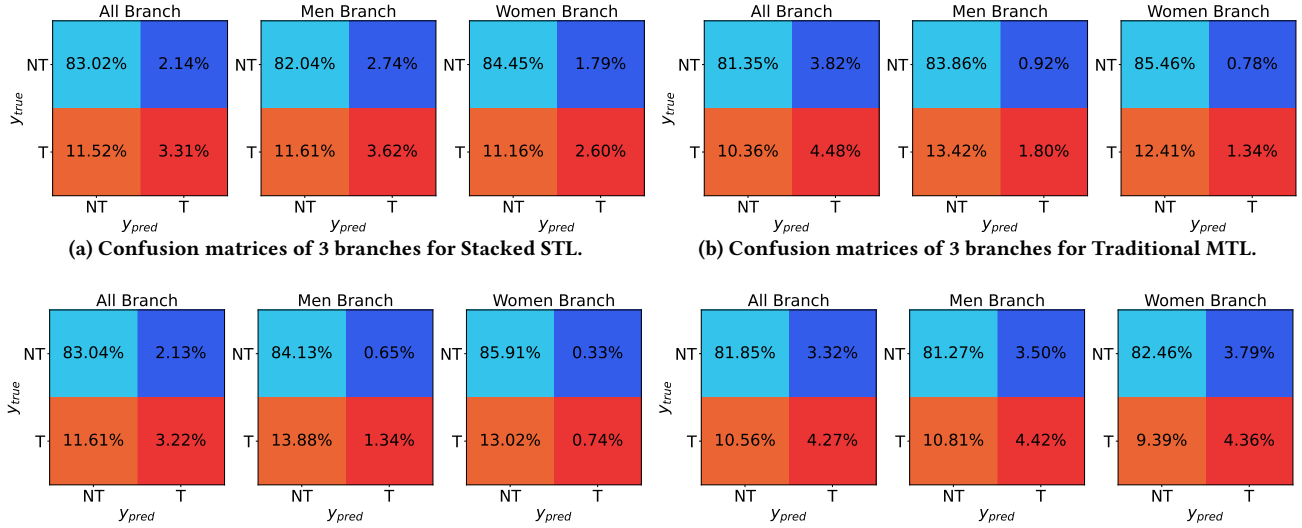
outperforming Stacked STL (24%, 24%). The superior performance of CondMTL in terms of recall, likely driven by its more accurate understanding of toxicity at a group-specific level, is crucial in the context of automated toxicity detection, where we would like to ensure that toxic posts are not mislabeled 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 [52].

In terms of precision, CSMTL performs the best (67%, 69%) compared to TradMTL (48%, 47%) and CondMTL (56%, 54%). CSMTL's higher precision number 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 [14]. 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 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.

Comparing the confusion matrices of the three branches of the MTL models in **Fig. 5** reveals that CondMTL performs better in



(a) Confusion matrices of 3 branches for Stacked STL.

(b) Confusion matrices of 3 branches for Traditional MTL.

(c) Confusion matrices of 3 branches for Cross Stitch MTL.

(d) Confusion matrices of 3 branches for Conditional MTL.

Figure 5: 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. The performance of the models are comparable in the *All* branch, since they train over the full dataset. CondMTL performs significantly better in the demographic specific *men* and *women* branches, due to training over group relevant examples only. This can be observed by the higher numbers in the red boxes of these branches for CondMTL compared to the other MTL variants. The toxic class is the smaller class, hence a good toxicity detection model should be able to correctly identify as many of them as possible. A good model should also have fewer misses (toxic posts identified as non-toxic), indicated by the orange boxes, where also CondMTL performs best.

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 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.

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.

6.3 Analysis of Conditional MTL

We make two propositions 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 in **Appendix D**, both on a regression and classification task.

Proposition 1. *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) 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. 1), thereby avoiding label contamination.

$$\begin{aligned} err_{all} &= wBCE([y_{true}]_{\mathcal{D}}, [y_{pred}]_{\mathcal{D}}) \\ err_{men} &= wBCE([y_{true}]_{\mathcal{D}_1}, [y_{pred}]_{\mathcal{D}_1}) \\ err_{women} &= wBCE([y_{true}]_{\mathcal{D}_2}, [y_{pred}]_{\mathcal{D}_2}) \end{aligned} \tag{1}$$

Proposition 2. *Cross Stitch MTL [30] allows contamination of weights across shared task layers.*

The CS unit (Fig. 4) 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. 2 are *w.r.t.* Fig. 4 where we have three CS matrices. These three CS matrices show clear deviation from symmetry in the off-diagonal elements.

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

7 DISCUSSION AND FUTURE WORK

Effect of Label Contamination. As a result of the misleading labeling schema discussed in Section 4.1, 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 [45] that can optimize a network for a fairness measure. Correspondingly, networks can also be optimized for equal accuracy across groups [23] or equalized odds [35]. 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 Sap et al. [43] 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 [40]. Using the proposed CondMTL to model the problem in relation to other stakeholders' demographics is a natural extension of the proposed work.

Extension to other problem domains. The proposed CondMTL can be easily applied to tackle group specific labeling in domains such as media bias detection and fact-checking, where sparsity in dataset labels could similarly lead to label contamination.

8 CONCLUSION

In this work, we frame the challenge of demographic-specific toxicity detection as a multi-task learning problem. We also note that the traditional MTL labeling schema causes label contamination under this problem setting, since posts targeted towards one group are labeled in a way that is misleading to the loss of branches specialized in other groups. To bypass this issue and address differential subgroup validity in the context of toxic language, we propose both: a) an updated labeling schema to avoid label contamination; and b) a conditional MTL framework with group-specific loss function based on a selective binary cross entropy formulation. The proposed architecture is shown to have significantly fewer trainable parameters and runtime than the stacked single task baseline, making it more suitable for deployment. Finally, we experimentally demonstrate that our framework achieves higher group recall values for the smaller toxic class over other baselines, and for the minority demographic group in particular, at no cost in overall accuracy compared to other MTL models.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable feedback. This research was supported in part by Wipro, an Amazon Research Award, and by *Good Systems* [2], a UT Austin Grand Challenge to develop responsible AI technologies. Our opinions reflect our own views only.

[2] http://goodsystems.utexas.edu/

REFERENCES

[1] Aymé Arango, Jorge Pérez, and Barbara Poblete. 2019. 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*. 45–54.

[2] Md Rabiul Awal, Rui Cao, Roy Ka-Wei Lee, and Sandra Mitrović. 2021. Angrybert: Joint learning target and emotion for hate speech detection. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 701–713.

[3] Jonathan Baxter. 1997. A Bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning* 28, 1 (1997), 7–39.

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

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

[6] François Chollet. 2015. keras. https://github.com/fchollet/keras.

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

[8] Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796* (2020).

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

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

[11] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. 2019. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3205–3214.

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

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

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

[15] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1923–1933.

[16] Xiaolei Huang and Michael Paul. 2019. Neural user factor adaptation for text classification: Learning to generalize across author demographics. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (* SEM 2019)*. 136–146.

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

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

[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

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

[21] 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. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*. PMLR, 5637–5664.

[22] Iasonas Kokkinos. 2017. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6129–6138.

[23] Venelin Kovatchev, Soumyajit Gupta, and Matthew Lease. 2022. Fairly Accurate: Learning Optimal Accuracy vs. Fairness Tradeoffs for Hate Speech Detection. *arXiv preprint arXiv:2204.07661* (2022).

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

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

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

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

[28] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. 2017. Learning multiple tasks with multilinear relationship networks. *Advances in neural information processing systems* 30 (2017).

[29] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. 2017. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5334–5343.

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

[31] Skye Morgan, Tharindu Ranasinghe, and Marcos Zampieri. 2021. WLV-RIT at GermEval 2021: Multitask Learning with Transformers to Detect Toxic, Engaging, and Fact-Claiming Comments. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments*. Association for Computational Linguistics, Duesseldorf, Germany, 32–38.

[32] Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing Gender Bias in Abusive Language Detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2799–2804. https://doi.org/10.18653/v1/D18-1302

[33] 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. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[34] Flor Miriam Plaza-del Arco, Sercan Halat, Sebastian Padó, and Roman Klinger. 2021. Multi-task learning with sentiment, emotion, and target detection to recognize hate speech and offensive language. *arXiv preprint arXiv:2109.10255* (2021).

[35] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. 2021. FairBatch: Batch Selection for Model Fairness. In *9th International Conference on Learning Representations*. The International Conference on Learning Representations.

[36] Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, Janet Pierrehumbert, et al. 2021. 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)*. Association for Computational Linguistics, 41–58.

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

[38] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[39] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2019. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4822–4829.

[40] Pratik S Sachdeva, Renata Barreto, Claudia von Vacano, and Chris J Kennedy. 2022. Assessing Annotator Identity Sensitivity via Item Response Theory: A Case Study in a Hate Speech Corpus. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 1585–1603.

[41] Niloofar Safi Samghabadi, Parth Patwa, Srinivas Pykl, Prerana Mukherjee, Amitava Das, and Thamar Solorio. 2020. Aggression and misogyny detection using BERT: A multi-task approach. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. 126–131.

[42] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

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

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

[45] Aili Shen, Xudong Han, Trevor Cohn, Timothy Baldwin, and Lea Frermann. 2022. 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*. Association for Computational Linguistics, Seattle, United States, 4073–4084. https://doi.org/10.18653/v1/2022.naacl-main.299

[46] Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 231–235.

[47] Timo Spinde, Jan-David Krieger, Terry Ruas, Jelena Mitrović, Franz Götz-Hahn, Akiko Aizawa, and Bela Gipp. 2022. Exploiting transformer-based multitask learning for the detection of media bias in news articles. In *International Conference on Information*. Springer, 225–235.

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

[49] Ameya Vaidya, Feng Mai, and Yue Ning. 2020. 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*, Vol. 14. 683–693.

[50] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2021. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence* (2021).

[51] Slavena Vasileva, Pepa Atanasova, Lluís Màrquez, Alberto Barrón-Cedeño, and Preslav Nakov. 2019. It Takes Nine to Smell a Rat: Neural Multi-Task Learning for Check-Worthiness Prediction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. 1229–1239.

[52] Emily A. Vogels. 2021. The State of Online Harassment. *Pew Res. Center, Washington, DC, USA, Tech. Rep* (2021).

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

[54] Yongxin Yang and Timothy Hospedales. 2017. Deep Multi-task Representation Learning: A Tensor Factorisation Approach. In *5th International Conference on Learning Representations*.

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

A EXPERIMENTAL DETAILS

A.1 Setup

Experiments use a Nvidia 2060 RTX Super 8GB GPU, Intel Core i7-9700F 3.0GHz 8-core CPU and 16GB DDR4 memory. We use the Keras [6] library on a Tensorflow 2.8 backend with Python 3.7 to train the networks in this paper. For optimization, we use AdaMax [19] with parameters (*lr*=1e-5) and 1000 steps per epoch. For each configuration, we did five independent runs to report mean and variance across different baselines and our method.

A.2 Training Loss Trajectory

We show the loss trajectory over the training data in **Fig. 6**. One can observe that the use of AdaMax results in the classification loss *i.e.,* weighted Binary Cross Entropy (WBCE) dropping rapidly at first and then stabilizing at around 10 epochs. Since the All branch deals with the full dataset, it achieves the highest error value. Due to the conditional nature of CondMTL, which only considers group relevant examples for each branch, we see improved loss over both the groups (men and women). More specifically, both groups start roughly at the same loss, with the men group achieving lower error values after stabilizing. This empirically highlights that CondMTL is not biased towards the majority group (women), rather giving importance to the minority group (men) as well.

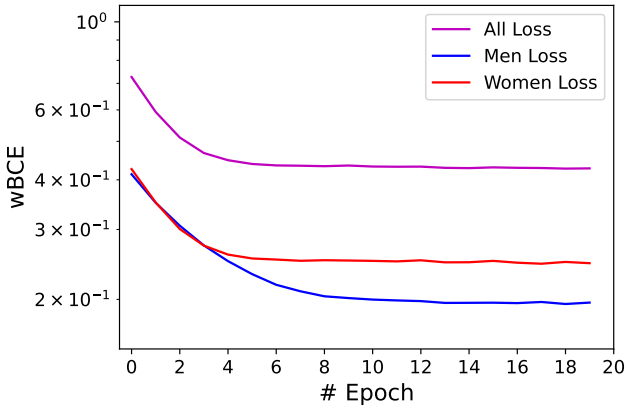


Figure 6: Loss trajectory over training data for CondMTL for 20 epochs. Note that due to the conditional nature of the formulation, the error in the men (minority group) goes down as well.

A.3 Class Balancing Strategy

As common in most toxic language tasks, the Wilds dataset has a class label imbalance, with most (85%) of the posts belonging to the non-toxic class. To account for this, we perform a class balancing strategy to assign more weight to the toxic examples during model training. We use a weighted version of Binary Cross Entropy (BCE) measure that re-weights the error for the different classes proportional to their inverse frequency in the data [24].

We use SKLearn's [33] *compute_class_weights='balanced'* flag for extracting weights $(w_{\text{toxic}}, w_{\text{non-toxic}})$ for toxic and non-toxic classes for each branch, given as[3]:

$$n\_samples/(n\_classes \times bincount(y)) \tag{3}$$

[3]https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

B MEAN VARIANCE NUMBERS

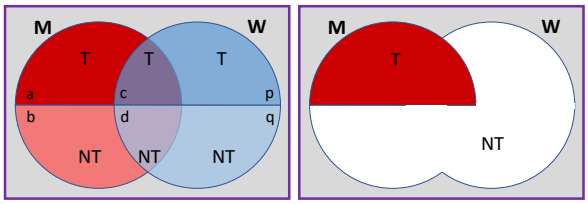
We report mean and standard deviation results of five runs over the model by using wBCE loss function. For simplicity, we show the overall accuracy numbers in Table 7. We observe that all of the models roughly perform the same *w.r.t.* the Stacked STL (~ 86%) and with negligible variance over runs. Note that all the five runs were independent *i.e.,* their network weights were randomly initialized at start of each training run.

Loss type	All	Men	Women
Stacked Single Task	86.3 ± 0.1	85.6 ± 0.2	87.0 ± 0.1
Cross Stitch Multi Task [30]	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 7: Mean and Standard Deviation measures of models across five runs. All the models perform the same *w.r.t.* Overall Accuracy.

C LABEL CONTAMINATION

To illustrate *label contamination*, let us consider a toxicity labeling task with binary labels (NT and T) and binary groups (M and F). Given the set of all possible posts, indicated by the grey rectangle in **Fig. 7 (a)**, the set of posts belonging to the groups M and F are indicated by their respective circles, red ($\mathcal{D}_1$) and blue ($\mathcal{D}_2$) shades respectively. The combined dataset $\mathcal{D}$ is a union of the two circles ($\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$) with some post targeted towards individual groups, and some targeting both groups (intersection). Within each group, we have a set of toxic (T) *vs.* non-toxic posts, indicated by dark and light shades respectively. The individual portions of the venn diagram are marked with lower case alphabet for clarity, as *e.g.,* region (*a*) indicates toxic post targeted towards men only, while region (*c*) indicates toxic posts targeted at both men and women.



(a) True Labels **(b) TradMTL Label for Men**

Figure 7: Illustration of label contamination when following TradMTL labeling schema due to false consideration of group-irrelevant posts as non-toxic (NT). (*a*) shows the true labels for each group, being toxic/non-toxic. However, following the TradMTL scheme for group (M) results in both toxic/non-toxic posts explicitly from group (W) (regions *p* and *q*) to falsely be grouped with the non-toxic labels for group (M) (ideally just regions *b* and *d*).

Table 8 shows the ideal and cross contaminated labels for posts *w.r.t.* toxic *vs.* no-toxic labels pertaining to each group. Due to label contamination in TradMTL, the non-toxic parts $NT|M$ and $NT|F$ have additional group-irrelevant labels associated with them which causes the classifier to learn incorrect boundaries.

Group	T | M	T | F	NT | M	NT | F
True Label	$a \cup c$	$c \cup p$	$b \cup d$	$d \cup q$
TradMTL Label	$a \cup c$	$c \cup p$	$b \cup d \cup p \cup q$	$d \cup q \cup a \cup b$

Table 8: TradMTL schema assumes group-irrelevant posts, leading to mislabeling in each non-toxic group, marked in red.

D BENCHMARKS

We verify our stated propositions for CondMTL and CSMTL through simple and verifiable benchmarking cases on a regression and classification task. We consider the integer group $\mathbb{Z}$ of numbers for the tasks {1,2,…,10}, with two groups (A and B), where group [A] contains the numbers {1,2,…,5} and group [B] contains the numbers {6,7,…,10}. To generate the actual data and mimic variability and imbalance, we replicate each number in the set in increments of 100 *i.e.,* $\mathcal{D} = [1:100, 2:200, \ldots, 10:1000]$ with each group data as $\mathcal{D}_A = [1:100, 2:200, \ldots, 5:500]$ and $\mathcal{D}_B = [6:600, 7:700, \ldots, 10:1000]$. The data notation $d = [p:q]$ indicates that there are q instances of element p in the dataset.

D.1 Regression Task

For the regression task, we check each MTL model's performance on a multiplication task. We define the following three tasks: T1) given data X produce $Y = 4X$ over the dataset $\mathcal{D}$; T2) given data X produce $Y = 2X$ over the dataset $\mathcal{D}_A$; and T3) given data X produce $Y = 6X$ over the dataset $\mathcal{D}_B$. The architecture contains a shared layer of 4 dense neurons and two depths for the task specific neurons containing 2 and 1 neurons respectively. Since scaling a number is a linear operation, we keep the dense neurons in the network with linear activations, and mean squared error loss.

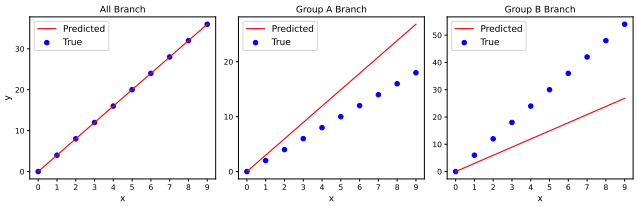
Fig. 8 shows the results obtained by the MTL models, where due to label contamination, both the CSMTL and TradMTL suffer in predictions for Groups A and B. Since CondMTL makes branch [A] learn over examples of $\mathcal{D}_A$ explicitly, it learns the representation of $Y = 2X$ and is able to predict $2X$ both in and out of domain of $\mathcal{D}_A$. Same logic holds for branch [B] which can predict $6X$ both in and out of domain of $\mathcal{D}_B$. Also looking at the entries of the CS matrix, one can observe that both identity and symmetry does not hold.

$$CS = \begin{bmatrix} 1.20 & 0.41 & -0.34 \\ 0.47 & 1.21 & -0.11 \\ -0.43 & -0.46 & 1.29 \end{bmatrix}$$

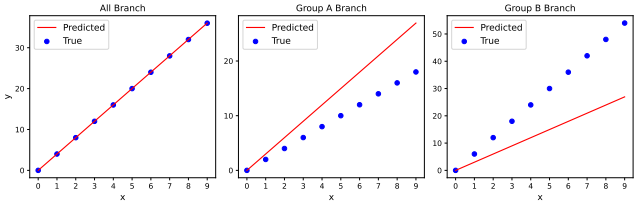
In another regression variant, we intentionally set task T2 as: given data X produce $Y = 0$ over the dataset $\mathcal{D}_A$, which essentially implies predicting a constant value for all input data. This updated task T2 is independent of both tasks T1 and T3, where the Cross Stitch unit should not communicate anything through T2. However, after training, we still observe that the off-diagonal CS matrix entries corresponding to Task 2 are still non-zero leading to contamination of weights across other task layers.

D.2 Classification Task

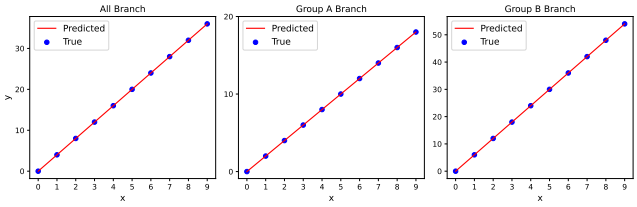
For the classification task, we keep the same architecture, but replace linear activation with *tanh* and for the final layer a *sigmoid* activation. We define the following three tasks: T1) given data X classify $Y = 1$ if $4 \leq X \leq 7$ else 0 over the dataset $\mathcal{D}$; T2) given data X classify $Y = 1$ if $X \geq 4$ else 0 over the dataset $\mathcal{D}_A$; and T3) given data X classify $Y = 1$ if $X \geq 7$ else 0 over the dataset $\mathcal{D}_B$. Similar faulty performances occur in this setting as well in both CSMTL and TradMTL due to label contamination, while CondMTL learns the correct group specific representation.



(a) Predictions for CSMTL



(b) Predictions for TradMTL



(c) Predictions for CondMTL

Figure 8: Predictions of the three MTL models over the benchmark regression tasks, where T1) $y = 4x$, T2) $y = 2x$ and T3) $y = 6x$. Since the all branch looks at the full data $\mathcal{D}$, it performs accurately over all the models. However, for the Group [A] and [B] branches, the CSMTL and TradMTL learns from a mix of labels of $\mathcal{D}_A$ and $\mathcal{D}_B$, which leads to its predictions overshooting and undershooting respectively.