Towards Unbiased and Accurate Deferral to Multiple Experts

Vijay Keswani
Yale University
vijay.keswani@yale.edu

Matthew Lease
University of Texas at Austin
Amazon AWS AI
ml@utexas.edu

Krishnaram Kenthapadi
Amazon AWS AI
kenthk@amazon.com

ABSTRACT

Machine learning models are often implemented in cohort with humans in the pipeline, with the model having an option to defer to a domain expert in cases where it has low confidence in its inference. Our goal is to design mechanisms for ensuring accuracy and fairness in such prediction systems that combine machine learning model inferences and domain expert predictions. Prior work on "deferral systems" in classification settings has focused on the setting of a pipeline with a single expert and aimed to accommodate the inaccuracies and biases of this expert to simultaneously learn an inference model and a deferral system. Our work extends this framework to settings where multiple experts are available, with each expert having their own domain of expertise and biases. We propose a framework that simultaneously learns a classifier and a deferral system, with the deferral system choosing to defer to one or more human experts in cases of input where the classifier has low confidence. We test our framework on a synthetic dataset and a content moderation dataset with biased synthetic experts, and show that it significantly improves the accuracy and fairness of the final predictions, compared to the baselines. We also collect crowdsourced labels for the content moderation task to construct a real-world dataset for the evaluation of hybrid machine-human frameworks and show that our proposed framework outperforms baselines on this real-world dataset as well.

CCS CONCEPTS

• **Human-centered computing** → *HCI theory, concepts and models*; • **Computing methodologies** → *Machine learning*.

KEYWORDS

Deferral models; Fairness; Hybrid human-machine frameworks

ACM Reference Format:

Vijay Keswani, Matthew Lease, and Krishnaram Kenthapadi. 2021. Towards Unbiased and Accurate Deferral to Multiple Experts. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society (AIES '21), May 19–21, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3461702.3462516

AIES '21, May 19–21, 2021, Virtual Event,USA

ACM ISBN 978-1-4503-8473-5/21/05…$15.00
https://doi.org/10.1145/3461702.3462516

1 INTRODUCTION

Real-world applications of machine learning (ML) models often involve the model working together with human experts [19, 33]. For example, a model that predicts the likelihood of a disease given patient information can choose to defer the decision to a doctor who can make a relatively more accurate diagnosis [43, 67]. Similarly, risk assessment tools work together with judges and domain experts to provide a baseline recidivism risk estimate [22, 29]. Other examples of such hybrid decision-making settings include financial analysis tools [78] and content moderation tools for abusive speech detection [60] and fake news identification [70].

Human-in-the-loop frameworks are often employed in settings where automated models cannot be trusted to have high quality inferences for all kinds of inputs. Beyond the incentive of improved overall accuracy, having human experts in the pipeline also ensures timely audits of the predictions [72] and helps fill gaps in the training of the automated models [48, 61]. A case in point is the study done by Chouldechova et al. [12] which showed that erroneous risk assessments by a child maltreatment hotline screening tool were frequently flagged as being incorrect by the human reviewers, implying that automated tools may not always cover the entire feature space that the domain experts use to make the decision.

However, the interaction between an ML model and a human expert is inherently more complicated than an entirely-automated pipeline. Prior studies on settings where human-in-the-loop frameworks have been implemented provide evidence of such complexities [2, 16, 27, 56]. One serious complication is the possibility of aggravated biases against protected groups, defined by attributes such as gender and race. With increasing utilization of ML in *human classification* tasks, the problem of biases against protected groups in automated predictions has received a lot of interest. This has led to a deep exploration of social biases in popular models/datasets and ways to algorithmically mitigate them [4, 53]. Nevertheless, a number of such biased models and datasets are still in use [58]. In a pipeline that involves an interaction between a possibly-biased ML model and a human, the biases of the human can aggravate the biases of the model [64]. For example, in a study by Green and Chen [29], participants were given the demographic attributes and prior criminal record of various defendants, along with the model-predicted risk of recidivism associated with each defendant, and asked to predict the risk. They found that the participants associated a higher risk with black defendants, compared to the model prediction. In this case, the possible biases of the human in the pipeline seem to exacerbate the bias of the model prediction. Similar ethical concerns regarding the interplay between the biases of model and humans have been highlighted in other papers [11, 68].

Motivated by the challenges discussed above, we focus on mechanisms for ensuring accuracy and fairness in hybrid machine-human pipelines. We consider the setting where a classification model is trained to either make a decision or defer the decision to human experts. Most machine-human pipelines employed in real-world applications have multiple human experts available to share the load and to cover different kinds of input samples [12, 30]. Therefore, the hybrid decision-making framework will have an additional task of appropriately choosing one or more experts when deferring. Each expert may also have their own area of expertise as well as possible biases against certain protected groups, characterized by their prior predictions on some samples. Correspondingly, the training of a machine learning model in such a composite pipeline has to take into account the domain expertise of the humans, and delegate the prediction task in an input-specific manner. Hence, our goal is to train a classifier and a deferral system such that the final predictions of the composite system are accurate and unbiased.

Our Contributions. We study the multiple-experts deferral setting for classification problems, and present a formal *joint learning framework* that aims to simultaneously learn a classifier and a *deferrer*. The job of the deferrer is to select one or more experts (including the classifier) to make the final decision (§2.1). As part of the framework, we propose loss functions that capture the costs associated with any given classifier and deferrer. We theoretically show that, given prior predictions from the human experts and true class labels for the training samples, the proposed loss functions can be optimized using gradient-descent algorithm to obtain an effective classifier and deferrer. Our framework further supports the settings where (a) number of experts that can be consulted for each input is limited, (b) each expert has an individual cost of consultation, and/or (c) expert predictions are available for only a subset of training samples (§2.2). To ensure that the final predictions are unbiased with respect to a given protected attribute, we propose two fair variants of the framework (*joint balanced* and *joint minimax-fair*) that aim to improve error rates across all protected groups. Our framework can handle both multi-class labels and non-binary protected attributes.

We empirically demonstrate the efficacy of our framework and its variants on multiple datasets: a synthetic dataset constructed to highlight the importance of simultaneously learning a classifier and a deferrer (§3.1), an offensive language dataset [17] with synthetically-generated experts (§3.2), and a real-world dataset constructed to specifically evaluate deferral frameworks with multiple available experts (§4). The real-world dataset consists of a large number of crowdsourced labels for the offensive language dataset, and is also a contribution of this paper. Unlike most crowdsourced datasets where the goal is simply to obtain accurate annotations, this dataset explicitly contains a dictionary of crowdworker (anonymized) to predicted labels, ensuring that the decision-making ability of each crowdworker can be inferred and consequently used to evaluate the performance of a hybrid framework like ours. We plan to publish this dataset as this will provide a strong empirical benchmark to foster future work. For all datasets, our framework significantly improves the accuracy of the final predictions (compared to just using a classifier and other baselines, such as task allocation algorithms of Li and Liu [45] and Qiu et al. [65] from crowdsourcing literature), and for the offensive language datasets, the fair variants of the framework also reduce disparity across the dialect groups.

Related Work. Given the difficulty of constructing and analyzing a human-in-the-loop framework, prior work has looked at human-in-the-loop settings from various viewpoints. One direction of research has explored the idea of the classifier having a "reject"/"pass" option for contentious input samples [13–15, 25, 37, 46, 49]. While such an option is usually provided to ensure that low confidence decisions can be deferred to human experts, the penalty of abstaining from making a decision in these models is fixed, and therefore, they do not take into account whether the expert at the end of the pipeline has the relevant knowledge to make the decision or not.

On the other hand, papers that take the biases and/or accuracies of the human experts into consideration are inherently more robust, but also more difficult to train and analyze. Prior theoretical models for learning to defer [18, 51, 55, 66, 75] have constructed explicit loss functions/optimization methods to model the combined inaccuracies and biases of the classifier and the human expert. Unlike the classifiers with reject option, they use a non-static loss function for the human expert and ensure that the penalty of deferring to a human expert is input-specific. However, [3, 18, 51, 55, 75] work with a single expert, assuming that the expert in the pipeline will be fixed and remain the same for future classification. Such an assumption is inhibitory in the settings where multiple experts are available [12], as different human experts can have different prediction behaviours [31]. Raghu et al. [66] model an optimization problem for the hybrid setting as well, but they learn a classifier and a deferrer separately, which (as shown by [55] and discussed in §3) cannot handle a large variety of input settings since the classifier does not adapt to the experts. In comparison, our method learns a classifier and a deferrer simultaneously, and can handle multiple experts.

Empirical studies in this direction [12, 19, 29, 41, 41, 79] often inherently use multiple experts since the results are based on crowdsourced data, but do not aim to propose a learning model for the pipeline. They, however, do highlight the importance of taking the domain knowledge of experts into account to improve the accuracy and fairness of the entire pipeline.

Another field that studies the problem of *task allocation* among different humans is *crowdsourcing*. Crowdsourcing for data collection is a popular approach to label or curate different kinds of datasets [44]. Since crowdworkers employed for such annotation tasks come from diverse backgrounds, prior work in crowdsourcing has looked at the related issue of efficient distribution of input amongst the available workers [38, 45, 57, 59, 61, 65, 73, 74, 76]. The main difference between this line of work and our setting is the presence of the automated classifier. In our setting, the classifier is expected to handle the primary load of prediction tasks and the role of human experts is to provide assistance for input samples where the classifier cannot achieve reasonable confidence. Crowdsourcing models, however, do not usually involve construction of any prediction model. One can alternately pre-train the classifier and treat it as another crowdworker to use task-allocation algorithms from crowdsourcing literature to distribute the samples among the

experts. The main issue with this approach is that training the classifier and deferrer separately can lead to an ineffective prediction pipeline. In our empirical analysis (§3), we assess the performance of two task-allocation algorithms from crowdsourcing literature [45, 65], and demonstrate the necessity of simultaneous training. See Appendix B for detailed discussion on these crowdsourcing methods.

2 MODEL

Each sample in the domain contains a class label, denoted by $Y \in \mathcal{Y}$, n-dimensional feature vector (default attributes) of the sample used to predict the class label, denoted by $X \in \mathcal{X}$, and additional information about the sample that is available only to the experts, denoted by $W \in \mathcal{W}$. W can represent different human factors that often assist in decision-making, such as training or background of the expert for the given task. Let Δ_Y denote the vertices of the simplex corresponding to the unique class labels in $\mathcal{Y}$ and let $\text{conv}(\Delta_Y)$ denote the simplex and its interior. Every sample also has a protected attribute $Z \in \mathcal{Z}$ associated with it (e.g., gender or race); Z can be part of default attributes X or additional attributes W, depending on the context.

Our framework consists of a classifier and a deferrer. The classifier $F : \mathcal{X} \to \text{conv}(\Delta_Y)$, given the default attributes of an input sample, returns a probability distribution over the labels of $\mathcal{Y}$. Let $L_{\text{clf}}(F; X, Y)$ denote the convex loss associated with the prediction of classifier F at point (X, Y). For $\ell > 0$, we will call L_{clf} an ℓ-Lipschitz smooth function if for all classifiers F,

$$\nabla_F^2 \left(\mathbb{E}_{X,Y} L_{\text{clf}}(F; X, Y) \right) \preccurlyeq \ell \cdot \mathbf{I}.$$

Intuitively, Lipschitz-smoothness characterizes how fast the gradient of L_{clf} changes around any point in the parameter space of the classifier; this characterization crucially helps determine the step-size required for the gradient-descent optimization of the loss function and will be useful for convergence rate bounds in our setting as well.

The framework also has access to $m-1$ human experts $E_1, \ldots, E_{m-1} : \mathcal{X} \times \mathcal{W} \to \Delta_Y$ who can assist with the decision-making. The output of the expert will be a vector with 1 for the index of the predicted class and 0 for all other indices (one-hot encoding). The experts are assumed to have access to the additional information (from domain $\mathcal{W}$) that can be used to make the predictions more accurately; however, deferring to an expert will come at an additional cost which we will quantify later. We also assume that there is an *identity expert* which just returns the decision made by the classifier F; therefore, in total we have m experts ($E_m(X, W) = F(X)$) (see Figure 2 in Appendix). For any given input X, the following notation will denote all the decisions,

$$Y_E(X, W) := [E_1(X, W)^\top, \ldots, E_{m-1}(X, W)^\top, F(X)^\top].$$

The goal of the deferral system $D : \mathcal{X} \to \{0, 1\}^m$, given the input, is to defer to one or more experts (including the classifier) who are likely to make accurate decision for the given input. Given any input, D will choose a committee of experts and the final output of the framework will be based on the entries of the following matrix-vector product: $Y_E(X, W) \cdot D(X)$ (the specific aggregation method used is specified in the §2.1). If the committee chosen contains only the *identity expert*, then the output of the framework is the output of the classifier F; otherwise, the output of the model is the aggregated decision of the chosen committee.

Remark 2.1. *The difference between a human-in-the-loop setting and setting with composition of multiple prediction models [6, 10, 23] is the access to additional information W. W represents the decision-making assistance available to the experts that is not available to the prediction model either due to computational limits on the prediction model or due to lack of availability of this data for training. This assumption crucially implies that, in most cases, we cannot construct a suitably-accurate model to simulate the predictions of the experts since the importance assigned to the additional information W is unknown. In the absence of W, one can only try to identify the input samples for which the expert is expected to be more accurate than the trained classifier; identifying such input samples using X is exactly the job of the deferrer in our framework. This distinction separates our problem setting from one where expert labels are used to bootstrap a classifier [61].*

2.1 Simultaneous Learning Classifier & Deferrer

We first present our framework for the case of binary class label and later discuss the extension to multi-class setting.

Binary class label, i.e, $\mathcal{Y} = \{0, 1\}$. Suppose the classifier F is fixed and, given the m experts, we need to provide a mechanism for training the deferral system (we will generalize this notion for simultaneous training shortly). For any given input X, the deferrer output $D(X)$ is expected to be a vector in the discrete domain $\{0, 1\}^m$. For the sake of smooth optimization, we will relax the domain of the output of D to include the interior of the hypercube $[0, 1]^m$, i.e., $D(X)$ will quantify the weight associated with each expert, for the given input X. Since we consider the binary class label setting, we can simplify our notation further for this section. Let $Y_{E,1}(X, W)$ denote the second row of the $2 \times m$ matrix $Y_E(X, W)$; this simplification does not lead to any loss of representational power since the sum of first and second row is the vector $\mathbf{1}$. Along similar lines as logistic regression, using $D(X)$ one can then directly calculate the output prediction (probabilistic) as follows: $\hat{Y}_D := \sigma(D(X)^\top Y_{E,1}(X, W))$, where $\sigma(x) := e^x/(e^x + e^{1-x})$. We can then train the deferrer to optimize the standard log-loss risk function:

$$\min_D -\mathbb{E}_{X,Y} \left[Y \log\left(\hat{Y}_D\right) + (1 - Y) \log\left(1 - \hat{Y}_D\right) \right].$$

The expectation is over the underlying distribution; the empirical risk can be computed as mean of losses over any given dataset samples (i.e., expectation over empirical distribution). For any input sample, the output prediction of the framework is 1 if $\sigma(D(X)^\top Y_{E,1}(X, W)) > 0.5$ else 0.

While the above methodology trained F and D separately, we can combine the training of the two components as well. To train F and D simultaneously, we introduce hyper-parameters α_1, α_2, and merge the loss functions for the classifier F and deferrer D linearly

using these hyperparameters.

$$L(F,D) = \alpha_1 \mathbb{E}_{X,Y}\left[L_{\text{clf}}(F;X,Y)\right] - \alpha_2 \mathbb{E}_{X,Y}\left[Y \log\left(\hat{Y}_D\right) + (1-Y)\log\left(1-\hat{Y}_D\right)\right].$$

The choice of hyperparameters is context-dependent and is discussed later. The goal of the framework is then to find the (classifier, deferrer) pair that optimizes $\min_{F,D} L(F,D)$. We will refer to this model as the *joint framework*. The joint learning framework extends the standard logistic regression method, and hence, exhibits some desirable properties.

Proposition 2.2. *$L(F,D)$ is convex in F and D, given a convex loss function L_{clf}.*

The convexity of the function enables us to use standard gradient-descent optimization approaches [7] to optimize the loss function. In particular, we will use the projected-gradient descent algorithm, with updates of the following form:

$$F_{t+1} = F_t - \eta \cdot \left.\frac{\partial L}{\partial F}\right|_{F=F_t},$$

$$D_{t+1} = \text{proj}_{\{0,1\}^m}\left(D_t - \eta \cdot \left.\frac{\partial L}{\partial D}\right|_{D=D_t}\right),$$

where $\eta > 0$ is the learning rate and $\text{proj}_{\{0,1\}^m}(\cdot)$ operator projects a point to its closest point in the hypercube $\{0,1\}^m$. Further, we can show that the gradient of the loss function assigns relatively larger weight to more accurate experts.

Theorem 2.3 (Deferrer gradient updates). *Suppose that α_1, α_2 are independent of the parameters of D. Let $Y_E \in \{0,1\}^m$ denote the decisions of the experts and classifier for any given input, and let Y denote the class label for this input. Then, for any $i \in \{1,\ldots,m\}$,*

$$-\frac{\partial L}{\partial D}^{(i)} \propto \begin{cases} e^{1-D^\top Y_{E,1}}, & \text{if } Y=1, Y = Y_{E,1}^{(i)}, \\ -e^{D^\top Y_{E,1}}, & \text{if } Y=0, Y \neq Y_{E,1}^{(i)}, \\ 0, & \text{otherwise.} \end{cases}$$

Here $u^{(i)}$ denotes the i-th element of vector u.

The above theorem states that gradient descent moves in a direction that rewards more accurate experts. Conditional on $Y=1$, the difference between the weight updates of a correct and an incorrect expert is proportional to $e^{1-D^\top Y_{E,1}}$. Similarly, conditional on $Y=0$, the difference between the weight updates of a correct and an incorrect expert is proportional to $e^{D^\top Y_{E,1}}$. We next provide convergence bounds for the projected gradient descent algorithm in our setting when L_{clf} is Lipschitz-smooth and α_1, α_2 are constants.

Theorem 2.4 (Convergence bound). *Suppose L_{clf} is ℓ-Lipschitz smooth and α_1, α_2 are constants. Let $(F^\star, D^\star) := \arg\min_{F,D} L(F,D)$. Given starting point F_0, such that $\|F_0 - F^\star\| \leq \delta$, step size $\eta = c(\ell+m)^{-1}$, for an appropriate constant $c > 0$, and $\varepsilon > 0$, the projected-gradient descent algorithm, after $T = O\left(\frac{(\ell+m)(\delta^2+m)}{\varepsilon}\right)$ iterations, returns a point F°, D°, such that*

$$L(F^\circ, D^\circ) \leq L(F^\star, D^\star) + \varepsilon.$$

Note that for m=1 (just the classifier), we recover the standard gradient descent convergence bound for ℓ-Lipschitz smooth loss function L_{clf}, i.e., $O(\ell\delta^2/\varepsilon)$ iterations [7]. For m>1, additionally finding the optimum deferrer results in an extra $(m(\delta^2+\ell)+m^2)/\varepsilon$ additive term. With standard classifiers and loss functions, we can use the above theorem to get non-trivial convergence rate bounds. For example, if F is a logistic regression model and L_{clf} is the log-loss function, Lipschitz-smoothness parameter ℓ is the maximum eigenvalue of the feature covariance matrix. Our theoretical results show that, given prior predictions from the experts and true class labels for a training set, loss function L can be used to train a classifier and an effective deferrer using gradient descent. Appendix A contains the proofs of all theoretical results.

Multi-class label. The above framework can be extended to multi-class settings as well. In this case, the matrix-vector product $Y_E(X,W) \cdot D(X)$ is a $|\mathcal{Y}|$-dimensional vector. Similar to the binary case, we extract the probability of every class label and represent it using $\hat{Y}_D$, where the j-th coordinate of $\hat{Y}_D$ represents the probability of class label being j, i.e,

$$\hat{Y}_D^{(j)} := \frac{e^{D(X)^\top Y_{E,j}(X,W)}}{\sum_{j'=1}^{|\mathcal{Y}|} e^{D(X)^\top Y_{E,j'}(X,W)}}.$$

The loss function $L(F,D)$ in this case can be written as

$$\alpha_1 \mathbb{E}_{X,Y}\left[L_{\text{clf}}(F;X,Y)\right] - \alpha_2 \mathbb{E}_{X,Y}\left[\sum_{j=1}^{|\mathcal{Y}|} \mathbb{1}[Y=j] \log \hat{Y}_D^{(j)}\right].$$

The final output of the framework, for any given input, is $\arg\max \hat{Y}_D$. The above loss function retains the desired properties from the binary setting; it is convex with respect to the classifier and deferrer, and the indicator formulation ensures that each gradient step still rewards the experts that are correct for any given training input. Additional costs considered in cost sensitive learning [80], e.g., different penalties for different incorrect predictions, can also be incorporated in our framework by simply replacing the indicator function $\mathbb{1}[Y=j]$ with the penalty function [55]. For the sake of simplicity, we omit those details from this version of the paper.

Choice of hyperparameters. α_1 and α_2 can either be kept constant or chosen in a context dependent manner. First, note that, since $\hat{Y}_D$ includes the classifier decision as well (scaled by the weight assigned to the classifier), keeping $\alpha_1 = 0$ would also ensure that the classifier and deferrer are trained simultaneously. However, due to the associated weight, classifier training with $\alpha_1 = 0$ can be slow and, since the initial classifier parameters are untrained, the classifier predictions in the initial training steps can be almost random. This will lead to the deferrer assigning low weight to the classifier. Correspondingly, depending on the complexity of the prediction task, it may be necessary to give the classifier a head-start as well. One way is to use time-dependent α_1, α_2. set $\alpha_1 = 1$ and $\alpha_2 = 1 - t^{-c}$, where $t \in \mathbb{Z}_+$ is the training iteration number and $c > 0$ is a constant. This choice ensures that in the initial iterations F is trained primarily and in the later iterations F and D are trained simultaneously.

There is a natural tradeoff associated with this head-start approach as well. The simultaneous training of F and D is crucial because the goal is to defer to experts for input where the classifier cannot make an accurate decision without the additional information. Therefore, a large head-start for the classifier can lead to a sub-optimal framework if the classifier tries to improve its accuracy over the entire domain.[1] Another choice of hyperparameters that can address this domain-partition setting is the following: set α_1=1 and α_2=$\mathbb{1}[\arg\max F(X) \neq Y]$ so that the deferrer is trained on training samples for which the classifier is incorrect.

2.2 Variants of the Joint Framework

We propose several variants of the joint learning framework that are inspired by the real-world problems that a human-in-the-loop model can encounter.

Fair Learning. The above joint framework aims to use the ability of the experts to ensure that the final predictions are more accurate than just the classifier. However, a possible pitfall of this approach can be that it can exacerbate the bias of the classifier, with respect to the protected attribute Z. Prior work has shown that misrepresentative training data [8, 42] or inappropriate choice of model [58], along with the biases of the human experts [29, 69] can lead to disparate performance across protected attribute types. An example of such disparity in our setting would be when, in an attempt to decrease the error rate of the prediction, the joint framework assigns larger weights to the biased experts, leading to an increase in disparity of predictions with respect to the protected attribute. We provide two approaches to handle the possible biases in our framework and ensure that the final predictions are fair.

Balanced Error Rate. One way to address the bias in final predictions is to give equal importance to all protected groups in our loss function. For protected attribute type z, let

$$L^z(F, D) := \alpha_1 \mathbb{E}_{X,Y|Z=z}\left[L_{\text{clf}}(F; X, Y)\right] - \alpha_2 \mathbb{E}_{X,Y|Z=z}\left[Y \log\left(\hat{Y}_D\right) + (1-Y)\log\left(1-\hat{Y}_D\right)\right].$$

Then the goal of this fair framework is to find the optimal solution for $\min_{F,D} \sum_{z\in\mathcal{Z}} L^z(F, D)$. The above method is also equivalent to assigning group-specific weights to the samples [26, 39]. We will refer to this framework as the *joint balanced framework*.

Minimax Pareto Fairness. Martinez et al. [52]'s proposed Pareto fairness aims to reduce disparity by minimizing the worst error rate across all groups. In other words, minimax Pareto fairness proposes solving the following optimization problem: $\min_{F,D} \max_{z\in\mathcal{Z}} L^z(F, D)$.

We will employ this fairness mechanism as well and refer to this framework as the *joint minimax-fair framework*. To understand the intuition behind this framework, we theoretically show that, in case of a binary protected attribute, the solution to the minimax Pareto fair program reduces the disparity between the risks across the protected attribute types.

[1]The synthetic experiment in §3.1 and the examples in Mozannar and Sontag [55] (for a single expert setting) highlight the necessity of simultaneously learning the classifier and deferrer.

Theorem 2.5 (Disparity of minimax-fair solution). *Suppose $\mathcal{Z} = \{0, 1\}$. Let $F^\star, D^\star := \arg\min_{F,D} \max_{z\in\mathcal{Z}} L^z(F, D)$ denote the joint minimax-fair framework optimal solution and let $F^\circ, D^\circ := \arg\min_{F,D} L(F, D)$ denote the joint framework optimal solution. Then*

$$\left|L^0(F^\star, D^\star) - L^1(F^\star, D^\star)\right| \leq \left|L^0(F^\circ, D^\circ) - L^1(F^\circ, D^\circ)\right|.$$

The proof is presented in Appendix A. Note that minimax Pareto fairness is a generalization of fairness by balancing error rate across the protected groups, but is also more difficult and costly to achieve. Furthermore, minimax Pareto fairness can handle non-binary protected attributes as well; we refer the reader to Martinez et al. [52] for further discussion on the properties of the minimax-fair solution. For our simulations, we will use the algorithm proposed by [20] to achieve minimax Pareto fairness (implementation available at https://github.com/amazon-research/minimax-fair).

Depending on the application, other fairness methods can also be incorporated in the framework. For example, if the fairness goal is to ensure demographic parity or equalized odds, then fairness constraints [9, 23], regularizers [40], or post-processing methods [34, 63] can alternately be employed.

Sparse Committee Selection. The joint framework could assign non-zero weight to all experts. In a real-world application, requiring predictions from all of the experts can be extremely costly. To address this, we propose a sparse variant to choose a limited number of experts per input.

The number of experts consulted for any given input can be limited by using the weights from $D(X)$ to construct a small committee. Suppose we are given that the committee size can be at most k. Then, for any input X, we construct a probability distribution over the experts with probability assigned to each expert being proportional to its weight in $D(X)$, and sample k experts i.i.d. from this distribution. The final output can be obtained by replacing $D^\top Y_E$ in $\hat{Y}_D$ by the mean prediction of the committee formed by this subset (scaled by the sum of weights in D). We refer to this framework as the *joint sparse framework*, when using the simple log-loss objective function, or *joint balanced/minimax-fair sparse framework*, when using either balanced or minimax-fair log-loss objective function. We can show that the expected error disparity between joint normal and joint sparse solutions indeed depends on the properties of the distribution induced by $D(X)$.

Theorem 2.6 (Price of sparsity). *Suppose $\mathcal{Y} = \{0, 1\}$ and let D denote the deferrer output and $\hat{Y}_D$ denote the prediction of the joint framework for a given input. Given $k \in [m]$, let random variable $\tilde{Y}_{D,k}$ denote the prediction of the joint **sparse** framework for this input. The expected difference of loss across the two predictions can be bounded as follows:*

$$\mathbb{E}\left|\log \hat{Y}_D - \log \tilde{Y}_{D,k}\right| < s_D \|D\|_1 + \max\left(2\|D\|_1, 1\right),$$

where s_D denotes the mean absolute deviation [28] of the distribution induced by D.

s_D characterizes the dispersion of the distribution induced by D and if D has low dispersion, then the expected difference of loss from choosing a committee from distribution induced by D is low.

The proof is presented in Appendix A. One could also, alternately, select the experts with the k-largest weights for each input [36].

Dropout. Given the possible disparities in the accuracies of the experts at the end of the pipeline, training a joint learning framework with diverse experts can suffer from the generalization pitfalls seen commonly in optimization literature [54]. If one expert is relatively more accurate than other experts the framework can learn to assign a relatively larger weight to this expert for every input compared to other experts. This is, however, quite undesirable as it assigns a disproportionate load to just one (or a small subset) of experts.

To tackle this issue, we introduce a random *dropout* procedure during training: an expert's prediction is randomly dropped with a probability of p and the expert's weight is not trained on the input sample for which it is dropped. This simple procedure helps reduce dependence on any single expert and ensures a relatively balanced load distribution.

Additional Regularization. As mentioned earlier, the experts can have individual costs associated with their consultation. Let $C_{E_1,\ldots,E_{m-1}} : \mathcal{X} \to \mathbb{R}^{m-1}$ refer to the vector of input specific cost of each expert consultation. Assuming that the costs of the experts are independent of one another, we can take these costs into account in our framework by adding $\lambda \cdot C_{E_1,\ldots,E_{m-1}}(X)^\top D(X)_{-1}$ as a regularizer to the loss function, where $D(X)_{-1}$ denotes the first $(m-1)$ elements of the vector $D(X)$ and $\lambda > 0$ is a hyperparameter.

3 SYNTHETIC SIMULATIONS

We first test the efficacy of the joint learning framework and its variants on synthetic settings. We use a synthetic and a real-world dataset for these simulations, and synthetically generate expert predictions for each input sample. For all datasets, L_{clf} will be the log-loss function and classifier F will be the standard logistic function.

3.1 Synthetic Dataset

Dataset and Experts. Each sample in the dataset contains two features, sampled from a two-dimensional normal distribution, and a binary class label (positive or negative). There are two available experts; their behaviour is described below.

Let $\mu \sim \text{Unif}(0,1)^2$ denote a randomly sampled mean vector and let $\Sigma \in \mathbb{R}^{2\times 2}$ denote a covariance matrix that is a diagonal matrix with diagonal entries sampled from $\text{Unif}(0,1)$. The data has 3 clusters, represented by colors *orange*, *blue*, and *green*. The *orange* cluster has two further sub-clusters: the first sub-cluster is sampled from the distribution $\mathcal{N}(\mu, \Sigma)$ and is assigned class label 1, while the second sub-cluster is sampled from the distribution $\mathcal{N}(\mu+3, \Sigma)$ and is assigned label 0. Since the sub-clusters are well-separated, this *orange* cluster can be accurately classified using the two dimensions.

The *blue* cluster is sampled from the distribution $\mathcal{N}(\mu+6, \Sigma)$, and each sample is assigned a class label 1 with probability 0.5. Expert 1 is assumed to be accurate over the *blue* cluster, i.e., if a sample belongs to the *blue* cluster, expert 1 returns the correct label for that sample; otherwise it returns a random label. Similarly, the *green* cluster is sampled from the distribution $\mathcal{N}(\mu+9, \Sigma)$, each sample is assigned a class label 1 with probability 0.5 and Expert 2 is assumed to be accurate over the *green* cluster and random for other clusters.

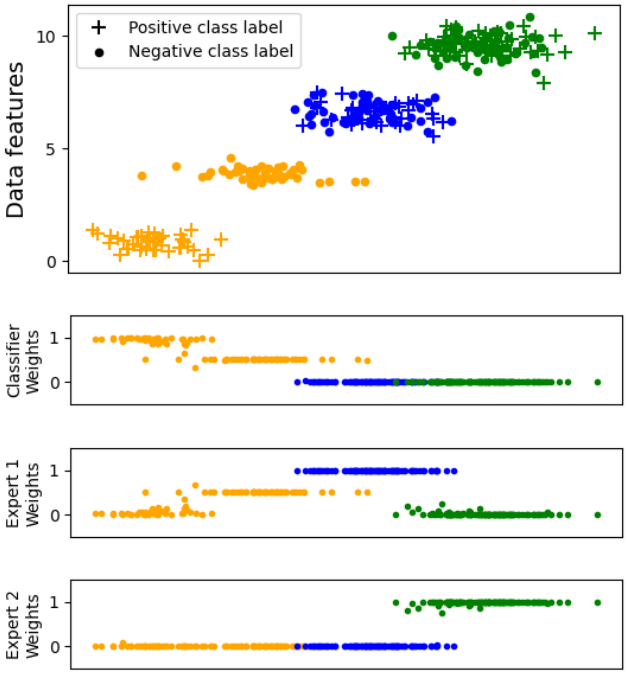

Figure 1: (§3.1 simulations) The first plot shows the data-points in the synthetic dataset. The next three plots show the weights assigned to classifier, expert 1 and expert 2 respectively for different clusters by the joint learning framework.

We construct a dataset with 1000 samples using the above process, with an almost equal proportion of samples in each cluster; the samples are randomly divided into train and test partitions (80-20 split). The distribution of the data-points is graphically presented in **Figure 1**. Suppose the hypothesis class of classifiers is limited to linear classifiers. The ideal solution (in the absence of any expert costs) is for the classifier to accurately classify elements of the *orange* cluster, and defer the samples from *blue* cluster to expert 1 and the samples from *green* cluster to expert 2. If the linear classifier is learnt before training the deferrer, then it will try to reduce error across all clusters, and resulting framework will not be accurate over any cluster, since clusters *blue* and *green* cannot be linearly separated. By studying the performance for this synthetic dataset we can determine if the joint learning framework accurately deciphers the underlying data-structure.

We also report the performance of two crowdsourcing algorithms: (a) *LL* algorithm [45] which tackles the worker selection problem, given the reliability and variance of all the workers, and (b) *Crowd-Select* [65], which aims to model the behaviour of the workers to appropriately allocate a subset of workers to each task. For both crowdsourcing algorithms, the classifier is pre-trained using the

train partition, and treated as just another worker. The details of these algorithms are provided in Appendix B.

Implementation Details. We use projected gradient descent, with 3000 iterations, learning rate $\eta = 0.05$, and $\alpha_1 = 0, \alpha_2 = 1$. As discussed before, $\alpha_1 = 0$ can also train the classifier and deferrer simultaneously.

Results. A baseline SVM classifier trained over the entire dataset has accuracy around 0.67 (accurate for one cluster and random over the other two). In comparison, the joint learning framework has perfect (1.0) accuracy. If the sparse variant of the joint learning framework is used with k=1 (defer to single expert), the accuracy drops to 0.91. To better understand the performance of the framework, Figure 1 presents the weights (normalized) assigned to the different experts (and classifier) for the test partition (bottom three plots).

Starting with the *green* cluster, the lowest plot shows that expert 2 is assigned the highest weight for samples in this cluster, implying that the prediction for this cluster is always correctly deferred to expert 2. Similarly, the prediction for the *blue* cluster is always correctly deferred to expert 1. For most of the samples in the *orange* cluster, the weight assigned to the classifier is larger than the weights assigned to the two experts. For some samples in this cluster, however, a non-trivial weight is also assigned to expert 1, which is why the accuracy for the sparse variant is lower than the accuracy of the non-sparse variant. This can be prevented using non-zero expert costs, which we employ in the next simulation.

The baseline *LL* algorithm achieves an accuracy of 67% on this dataset; this is because it associates a single measure of aggregated reliability with each worker, which in this case is unsuitable since each worker has their specific domain of expertise. The *CrowdSelect* algorithm achieves the best accuracy of around 83%; in this case, the error models for each expert and the classifier are constructed individually. Due to this, the algorithm is unable to perfectly stratify the input space amongst the experts (and classifier).

Discussion. The purpose of this simulation was to show that the deferrer can choose experts in an input-specific manner. The results show that the deferrer can indeed decipher the underlying structure of the dataset, and accordingly choose the expert(s) to defer to for each input (addressing the drawback of *LL*). The important aspect of the problem to notice here is that the cluster identity is the additional information available only to the experts. The cluster identity is crucial for the experts as it reflects their domain of expertise and helps them make the correct prediction if the sample lies in their domain. On the other hand, the cluster identity is useful to the deferrer only to defer correctly; even if the cluster is part of the input, the framework cannot use it to make a correct prediction, but can use it to defer to the correct expert. In other words, the framework can use the available information to identify samples that need to be deferred to an expert (addressing the drawback of *CrowdSelect*). This sub-problem of directly identifying contentious input samples is also related to prior work by Raghu et al. [67].

3.2 Offensive Language Dataset

Dataset. Our base dataset consists of around 25k Twitter posts curated by Davidson et al. [17]; all posts are annotated with a label that corresponds to whether they contain hate speech, offensive language, or neither. We set class label to 1 if the post contains hate speech or offensive language, and 0 otherwise. Using the dialect identification model of Blodgett et al. [5], we also label the dialect of the posts: African-American English (AAE) or not. Around 36% of the posts in the dataset labeled as AAE. We treat dialect as the protected attribute in this case.

Experts. The experts are constructed to be biased against one of the dialects. We generate m synthetic experts, with $\lfloor 3m/4 \rfloor$ experts biased against AAE dialect and $\lceil m/4 \rceil$ experts biased against non-AAE dialect. To simulate the first $\lfloor 3m/4 \rfloor$ experts, for each expert $i \in \{1, \ldots, \lfloor 3m/4 \rfloor\}$, we sample two quantities: $p_i \sim \text{Unif}(0.6, 1)$ and $q_i \sim \text{Unif}(0.6, p_i)$. For expert i, p_i will be the its accuracy for the non-AAE group and q_i will its accuracy for the AAE group. To make a decision, if the input belongs to the non-AAE group then this expert outputs the correct label with probability p_i and if the input belongs to the AAE group then this expert outputs the correct label with probability q_i. By design, the first $\lfloor 3m/4 \rfloor$ experts can have a certain level of bias against the AAE group since $q_i < p_i$ for all $i \in \{1, \ldots, \lfloor 3m/4 \rfloor\}$. The same process, with flipped p_i and q_i, is repeated for the remaining $\lceil m/4 \rceil$ experts, so that they are biased against the non-AAE group.

Baselines. There are three simple baselines that can be easily implemented: (1) using the classifier only, (2) randomly selected committee - a committee of size $\lceil m/4 \rceil$ is randomly selected (in this case, the predictions are expected to be biased against the AAE dialect since most of the experts are biased against the AAE dialect - see §C), and (3) random fair committee - i.e., if the post is in AAE dialect, the committee randomly selects from experts with higher accuracy for AAE group, and if the post is in non-AAE dialect, the committee randomly selects from experts with higher accuracy for non-AAE group. This committee selection should ensure relatively balanced accuracy across the dialects, and can therefore be used to judge the fairness of the joint learning framework. We also implement and report the performance of *LL* and *CrowdSelect* algorithms for this dataset.

Implementation Details. The dataset is split into train and test partitions (80-20 split). For both classifier and deferrer, we use a simple two-layer neural network, that takes as input a 100-dimensional vector corresponding to a given Twitter post (obtained using pre-trained GloVe embeddings [62]). The experts are given a cost of 1 each, i.e., $C_{E_1,\ldots,E_{m-1}} = \mathbf{1}$ and $\lambda = 0.05$ (the regularizer used is $\lambda \cdot \mathbb{E}[C_{E_1,\ldots,E_{m-1}}(X)^\top D(X)_{-1}]$). Inspired by prior work on adaptive learning rate [21], exponent c of parameter α is set at 0.5 and dropout rate at 0.2. We present the results for $m = 20$ in this section and discuss the performance for different m, λ, and dropout rate in Appendix C. We use stochastic gradient descent for training with learning rate $\eta = 0.1$ and for 100 iterations with batch size of 200 per iteration. For the sparse variants with $m = 20$, we sample $k = 5$ experts from the output distribution. The process is repeated 100

Method		Overall Accuracy	Non-AAE Accuracy	AAE Accuracy
Baselines	Classifier only	.89 (.00)	.86 (.00)	.96 (.00)
	Randomly selected committee	.84 (.07)	.83 (.10)	.85 (.01)
	Randomly selected fair committee	.88 (.06)	.86 (.11)	.93 (.03)
	LL	.96 (.03)	.97 (.03)	.95 (.04)
	CrowdSelect	.91 (.04)	.89 (.06)	.93 (.04)
Joint learning frameworks and fair variants	Joint framework	.92 (.02)	.89 (.03)	.97 (.00)
	Joint balanced framework	.94 (.01)	.92 (.02)	.98 (.00)
	Joint minimax-fair framework	.98 (.01)	.98 (.01)	.97 (.01)
Sparse variants of joint learning framework	Joint sparse framework	.92 (.01)	.90 (.02)	.96 (.01)
	Joint balanced and sparse framework	.92 (.01)	.89 (.01)	.97 (.00)
	Joint minimax-fair and sparse framework	.98 (.01)	.97 (.01)	.98 (.00)

Table 1: Overall and dialect-specific mean accuracies (standard error in brackets) for simulations in §3.2.

times, with a new set of experts sampled every time, and we report the mean and standard error of the overall and dialect-specific accuracies.

Results. The results for the joint learning framework and its variants, along with the baselines are presented in **Table 1**. The joint learning framework has a larger overall and group-specific average accuracy than the classifier. The best group-specific and overall accuracy is achieved by the joint minimax-fair framework (and its sparse variant), showing that it is indeed desirable to enforce minimax-fairness in this setting as it leads to an overall improved performance across all groups. The sparse variations of all joint frameworks, as expected, still have better performance than the classifier and random-selection baselines, and are quite similar to the non-sparse variants. Joint fair (balanced and minimax-fair) frameworks also have similar or lower accuracy disparity across the groups than random fair committee baseline. This shows that the learnt deferrer is also able to differentiate between biased and unbiased experts to an extent. Due to the non-zero λ parameter used, on average, the classifier is assigned around 5% of the deferrer weight per input sample. This implies that, when creating sparse committees with $k = 5$, the classifier is consulted for around 25% of the input samples. This fraction can be further increased by appropriately increasing λ.

Further, due to our use of dropout, more accurate experts are not assigned disproportionately high weights, exhibiting the effectiveness of load balancing using dropout. This is demonstrated in **Figure 3** in Appendix, which presents variation of the weights assigned by the joint framework to the experts vs the accuracies of the experts for a single repetition.

The *LL* algorithm is able to achieve very high overall accuracy ($\geq$ 95% for both groups) for this setting. However, our joint minimax-fair sparse framework has two advantages over *LL* algorithm. First, it achieves relatively better accuracy for both dialect groups. Second, *LL* pre-selects the most accurate experts to whom all the inputs are deferred. This is problematic and inefficient since *LL* only uses k out of m experts; in comparison, our algorithm distributes the input samples amongst all experts to reduce the load on the most accurate experts (see Figure 3 in Appendix). *CrowdSelect*, on the other hand, achieves lower overall and group-specific accuracies than joint minimax-fair frameworks.

4 SIMULATIONS USING REAL-WORLD DATA FOR THE OFFENSIVE LANGUAGE DATASET

The simulations in the previous sections highlighted the effectiveness of the joint learning framework in improving the accuracy and fairness of the final prediction. In this section, we present the results on a similar real-world dataset of Twitter posts, annotated using Mechanical Turk (MTurk).

Dataset. We use a dataset of 1471 Twitter posts for the MTurk survey. This is a subset of the larger dataset by Davidson et al. [17]. Importantly, this dataset is jointly balanced across the class categories used in Davidson et al. [17] and the two dialect groups (as predicted using Blodgett et al. [5]). Once again, the labels from Davidson et al. [17] are treated as the *gold labels* for this dataset.

MTurk Experiment Design. The MTurk survey presented to each participant started with an optional demographic survey. This was followed by 50 questions; each question contained a Twitter post from the dataset and asked the participant to choose one of the following options: 'Post contains threats or insults to a certain group', 'Post contains threats or insults to an individual', 'Post contains other kinds of threats or insults, such as to an organization or event', 'Post contains profanity', 'Post does not contain threats, insults, or profanity'. The options presented to the user are along the lines of the taxonomy of offensive speech suggested by Zampieri et al. [77]. The first four options correspond to offensive language in the Twitter post, while the last option corresponds to the post being non-offensive. As in the synthetic simulations, the participants are also provided the predicted dialect label of the post. The participants were paid a sum of $4 for completing the survey (at an hourly rate of $16).

MTurk Experiment Results. Overall, 170 MTurk workers participated in the survey and each post in the dataset was labeled by around 10 different annotators. Since each participant only labels a fraction of the dataset, we will treat this setting as one where there are missing expert predictions during the training of the joint

Method	Overall Accuracy	Non-AAE Accuracy	AAE Accuracy
Classifier only	.78 (.02)	.76 (.05)	.80 (.04)
Joint framework	.85 (.03)	.87 (.04)	.83 (.03)
Joint balanced framework	.84 (.03)	.87 (.03)	.81 (.04)
Joint minimax framework	.85 (.02)	.87 (.02)	.83 (.02)

Table 2: Results of the joint learning framework and fair variants on the MTurk dataset.

learning framework. The inter-rater agreement, as measured using Krippendorff's α measure, is 0.27. As per heuristic interpretation [32], this level of interrater agreement is considered quite low for a standard dataset annotation task. However, it is suitable for our purpose since our framework aims to address situations where there is considerable disparity in the performances of different humans in the pipeline, and the goal of the joint learning framework is to choose the annotators that are expected to be accurate for the given input.

The overall accuracy of the aggregated responses (i.e., taking a majority of all responses for every post and comparing to the *gold label*) is around 87%, which is close to the accuracy of the automated classifier in §3.2 (84% for AAE posts and 91% for non-AAE posts). The high accuracy shows that using crowdsourced annotations in this setting is quite effective and the hypothetical *aggregated crowd annotator* can indeed be considered an *expert* for this content moderation task. However, the individual accuracies of the experts is arguably more interesting and relevant to our setting.

The average individual accuracy of a participant is 77% (±13%). The minimum individual accuracy is ≈ 38% while the maximum individual accuracy is 98%. The wide range of accuracies evidences large variation in annotator expertise for this task. The individual accuracies for posts from different dialects also presents a similar picture. The average individual accuracy of a participant for the AAE dialect posts is 76% (±15%) and average individual accuracy of a participant for the non-AAE dialect posts is 78% (±14%).

While mean individual accuracies for the two dialects are quite similar, most annotators do display a disparity in their accuracy across the two groups. 92 of the 170 participants had a higher accuracy when labeling posts written in a non-AAE dialect. The average difference between the accuracy for non-AAE dialect posts and AAE dialect posts for this group of participants was 8.5% (±6.6%). 75 participants had a higher accuracy when labeling posts written in the AAE dialect. The average difference between the accuracy for AAE dialect posts and non-AAE dialect posts was 7.1% (±5.5%). Three remaining participants were equally accurate for both groups. The disparate accuracies here are quite similar to those in the early synthetic simulations. We next analyze the performance of joint learning framework on this dataset.

Joint Learning Framework Results on MTurk Dataset. We perform five-fold cross validation on the collected dataset. For each fold, we train our joint learning framework (with $\eta = 0.3$) on the train split and evaluate it on the test split. Since expert decisions are available only for a subset of the dataset, we do not use dropout or expert costs. Results are shown in **Table 2**. As before, the overall accuracy of the joint learning frameworks is higher than the accuracy of the classifier alone. Amongst the fair variants, even though the accuracy for both dialect groups is larger when using the balanced or minimax loss function (compared to the classifier alone), it does not lead to significantly different group-specific accuracies vs. simple joint learning framework. The performance of sparse variants is presented in Appendix D. Since a relatively small number of prior predictions is available for each expert, the task of differentiating between experts here is tougher. Hence, sparse variants perform similar or better than the classifier when committee size k is around 60 or greater.

Discussion. The wide range in accuracy observed across annotators confirms the expectation that different humans-in-the-loop will naturally bring varying levels and domains of expertise. Their accuracy will be affected by not only the training they receive, but also by their background. For example, native speakers of a given dialect are naturally expected to be better annotators for language examples from that dialect. However, despite the difficulty of the task and the disparity in group accuracies, our joint learning framework is still able to identify the combination of experts that are suitable for any given input and, correspondingly, increase the accuracy and fairness of the final prediction.

5 DISCUSSION, LIMITATIONS, AND FUTURE WORK

Our proposed framework addresses settings that involve active human-machine collaboration. Having shown its efficacy for synthetic and real-world datasets, we next highlight certain limitations and fruitful directions for future work.

Fairness of the Framework. It is crucial that the framework is fair with respect to the protected attribute. We proposed two methods for ensuring that the predictions are unbiased: by trying to achieve a balanced error rate for all groups, or by trying to minimize the maximum group-specific error rate (minimax Pareto fairness). Both fairness mechanisms can handle multi-class protected attributes, which helps generalize our framework to settings beyond simple binary protected attributes (e.g., multiple racial categories). An additional advantage of using these fairness definitions is that the protected group labels are not required for test or future samples, addressing the issue of their possible unavailability due to policy or privacy restrictions [24].

As mentioned in §2.2, other fairness mechanisms can also be incorporated into our framework. For most applications, the choice of fairness mechanism and constraint is often a context-dependent question. An uninformed choice of these variables can possibly lead to a degradation of both accuracy and fairness [47] and, therefore, it is important to take the impact of any fairness constraint on the user population into account before its implementation. Similarly, in our setting, it is important to first decide whether the goal of fairness is minimizing the worst group error, demographic parity, etc., and then choose the mechanism to implement it.

Real-world Benchmark Dataset. We created an MTurk dataset for evaluating human-in-the-loop prediction frameworks with multiple experts for detecting hate speech. The goal of constructing this dataset was to facilitate the learning and evaluation of hybrid frameworks, since having a large number of annotations for each input better enables a learning procedure to differentiate between annotators with different abilities. Existing datasets have often released only aggregate labels, such as by majority voting, which supports ML model training but does not allow modeling individual annotators. To be able to release such data, we have replaced annotator platform IDs with automatically generated pseudonyms.

Our new dataset has important limitations. First, in order to obtain a large number of annotations for each Twitter post, we kept the dataset size relatively small. Furthermore, since the dataset is a subset of the dataset constructed by Davidson et al. [17], it cannot be considered representative of the larger population of Twitter posts/users and the performance demonstrated in our simulations may not translate to larger Twitter datasets. The number of human annotators (170) in our survey is also larger than desired, even though each annotator labels 50-100 posts. Our framework aims to learn the domain of expertise of the human experts using only the prior decisions of the experts. However, it is not completely clear how many prior decisions are needed to accurately determine the domain of expertise of every annotator. The gap between the performance using synthetic experts (§3) and real-world experts (§4) partially shows that it might be necessary to get more predictions for each expert.

Poursabzi-Sangdeh et al. [64], in a position paper on human-in-the-loop frameworks in facial recognition, argue the necessity of real-world empirical studies of such frameworks to justify their widespread use. They also list the technical challenges associated with such empirical studies. The real-world dataset we provide attempts to initiate a real-world empirical study of human-in-the-loop frameworks for content moderation but, at the same time, faces similar challenges as highlighted by Poursabzi-Sangdeh et al., i.e., issues with data availability and generalizability of participants/context.

MTurk Experiment Generalizability. Similar to any other study done using MTurk participants, questions can be raised about the generalizability of the results to a larger population. While MTurk participants do seem suitable for detecting offensive language in Twitter posts (as seen from the performance of the *aggregated crowdworker* in §4), they may not accurately represent how a lay person would respond to a similar survey or how a domain expert would judge the same posts. The performance of domain experts (people with more experience in screening offensive language) will most likely be better than the accuracy of an average crowd annotator. Correspondingly, our framework with better trained content moderation experts can be expected to have similar or better performance. Nevertheless, as pointed out in prior work [1, 64], experimental design and choice of participants will play a much bigger role in simulating human-in-the-loop frameworks in settings where human experts cannot be imitated by volunteers.

Replaceable Experts. An extension of our model that can be further explored is addition/removal of experts. If a new expert is added to the pipeline and the domain of expertise of this expert is different than the domain of the replaced/existing experts, then the framework might need to be retrained to appropriately include the new expert. This overhead of retraining can, however, be avoided. For instance, one could train the framework using a *basis of experts*, i.e., divide the feature space into interpretable sub-domains and map the experts to these sub-domains. Then if we train the framework using sample decisions of experts with disjoint sub-domains of expertise, we can ensure that the entire feature space is covered either by the classifier or the deferrer (in a similar manner as §3.1), and any new expert could be mapped to the corresponding sub-domain. Approaches from prior work [50, 71] can be potentially used to learn these sub-domains and extend our joint learning framework for such settings.

Improved Implementation. Like other complex frameworks involving many decision making components, our framework can also suffer from issues that arise from real-world implementations. For instance, dropout reduces overdependence on any particular expert, but does not consider the load on any small subset of experts. Alternate load distribution techniques (e.g., Nguyen et al. [57]) can be explored further, at the risk of inducing larger committee sizes. Another extension that can be pursued is to keep the committee size small but variable; this can help with load distribution as well as better committee selection.

6 CONCLUSION

We proposed a joint learning model to simultaneously train a classifier and a deferrer in the multiple-experts setting. The code and dataset are available at https://github.com/vijaykeswani/Deferral-To-Multiple-Experts. Our framework can help increase the applicability of automated models in settings where human experts are an indispensable part of the pipeline. At the same time, by addressing the domains and biases of the model and the humans, we ensure that its utilization is thoughtful and context-aware.

REFERENCES

[1] Alfredo Alba, Anni Coden, Anna Lisa Gentile, Daniel Gruhl, Petar Ristoski, and Steve Welch. 2017. Multi-lingual concept extraction with linked data and human-in-the-loop. In *Proceedings of the Knowledge Capture Conference*. 1–8.

[2] Eugenio Alberdi, Lorenzo Strigini, Andrey A Povyakalo, and Peter Ayton. 2009. Why are people's decisions sometimes worse with computer support?. In *International Conference on Computer Safety, Reliability, and Security*. Springer, 18–31.

[3] Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S Weld. 2020. Optimizing AI for Teamwork. *arXiv preprint arXiv:2004.13102* (2020).

[4] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. *Fairness and Machine Learning*. fairmlbook.org. http://www.fairmlbook.org

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

[6] Amanda Bower, Sarah N Kitchen, Laura Niss, Martin J Strauss, Alexander Vargas, and Suresh Venkatasubramanian. 2017. Fair pipelines. *arXiv preprint arXiv:1707.00391* (2017).

[7] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[8] Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Conference on Fairness, Accountability and Transparency*. 77–91.

[9] L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. 2019. Classification with Fairness Constraints: A Meta-Algorithm with Provable Guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 319–328.

[10] Lingjiao Chen, Matei Zaharia, and James Zou. 2020. FrugalML: How to use ML Prediction APIs more accurately and cheaply. In *NeurIPS*.

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

[12] Alexandra Chouldechova, Diana Benavides-Prado, Oleksandr Fialko, and Rhema Vaithianathan. 2018. A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. In *Conference on Fairness, Accountability and Transparency*. 134–148.

[13] Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Scott Yang. 2018. Online learning with abstention. In *International conference on machine learning*. PMLR, 1059–1067.

[14] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. 2016. Boosting with abstention. *Advances in Neural Information Processing Systems* 29 (2016), 1660–1668.

[15] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. 2016. Learning with rejection. In *International Conference on Algorithmic Learning Theory*. Springer, 67–82.

[16] Mary Cummings. 2004. Automation bias in intelligent time critical decision support systems. In *AIAA 1st Intelligent Systems Technical Conference*. 6313.

[17] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

[18] Abir De, Paramita Koley, Niloy Ganguly, and Manuel Gomez-Rodriguez. 2020. Regression under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 2611–2620.

[19] Maria De-Arteaga, Riccardo Fogliato, and Alexandra Chouldechova. 2020. A Case for Humans-in-the-Loop: Decisions in the Presence of Erroneous Algorithmic Scores. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[20] Emily Diana, Wesley Gill, Michael Kearns, Krishnaram Kenthapadi, and Aaron Roth. 2021. Minimax Group Fairness: Algorithms and Experiments. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*.

[21] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research* 12, 7 (2011).

[22] Grant Duwe and Michael Rocque. 2017. Effects of Automating Recidivism Risk Assessment on Reliability, Predictive Validity, and Return on Investment (ROI). *Criminology & Public Policy* 16, 1 (2017), 235–269.

[23] Cynthia Dwork and Christina Ilvento. 2019. Fairness Under Composition. In *Innovations in Theoretical Computer Science Conference (ITCS)*.

[24] Lilian Edwards and Michael Veale. 2017. Slave to the algorithm: Why a right to an explanation is probably not the remedy you are looking for. *Duke L. & Tech. Rev.* 16 (2017), 18.

[25] Ran El-Yaniv et al. 2010. On the Foundations of Noise-free Selective Classification. *Journal of Machine Learning Research* 11, 5 (2010).

[26] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. 2019. A comparative study of fairness-enhancing interventions in Machine Learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 329–338.

[27] Kate Goddard, Abdul Roudsari, and Jeremy C Wyatt. 2012. Automation bias: a systematic review of frequency, effect mediators, and mitigators. *Journal of the American Medical Informatics Association* 19, 1 (2012), 121–127.

[28] Stephen Gorard. 2005. Revisiting a 90-year-old debate: The Advantages of the Mean Deviation. *British Journal of Educational Studies* 53, 4 (2005), 417–430.

[29] Ben Green and Yiling Chen. 2019. Disparate interactions: An algorithm-in-the-loop analysis of fairness in risk assessments. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 90–99.

[30] Tor Grønsund and Margunn Aanestad. 2020. Augmenting the Algorithm: Emerging human-in-the-loop work configurations. *The Journal of Strategic Information Systems* 29, 2 (2020), 101614.

[31] Melody Guan, Varun Gulshan, Andrew Dai, and Geoffrey Hinton. 2018. Who Said What: Modeling Individual Labelers Improves Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[32] Kilem L Gwet. 2011. On The Krippendorff's Alpha Coefficient. (2011). https://agreestat.com/papers/onkrippendorffalpha_rev10052015.pdf

[33] Aaron Halfaker and R Stuart Geiger. 2019. ORES: Lowering Barriers with Participatory Machine Learning in Wikipedia. *arXiv preprint arXiv:1909.05189* (2019).

[34] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*. 3315–3323.

[35] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. 2013. *Convex Analysis and Minimization Algorithms I: Fundamentals*. Vol. 305. Springer science & business media.

[36] Hyun Joon Jung and Matthew Lease. 2013. Crowdsourced Task Routing via Matrix Factorization. *arXiv preprint arXiv:1310.5142* (2013).

[37] Hyun Joon Jung, Yubin Park, and Matthew Lease. 2014. Predicting Next Label Quality: A Time-Series Model of Crowdwork. *HCOMP* 14 (2014), 1–9.

[38] Ece Kamar, Ashish Kapoor, and Eric Horvitz. 2015. Identifying and Accounting for Task-dependent Bias in Crowdsourcing. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 3.

[39] Faisal Kamiran and Toon Calders. 2009. Classifying without Discriminating. In *2009 2nd International Conference on Computer, Control and Communication*. IEEE, 1–6.

[40] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 643–650.

[41] Michael Katell, Meg Young, Dharma Dailey, Bernease Herman, Vivian Guetler, Aaron Tam, Corinne Bintz, Daniella Raz, and PM Krafft. 2020. Toward situated interventions for algorithmic equity: lessons from the field. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 45–55.

[42] Matthew Kay, Cynthia Matuszek, and Sean A Munson. 2015. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3819–3828.

[43] Peter Kieseberg, Edgar Weippl, and Andreas Holzinger. 2016. Trust for the doctor-in-the-loop. *ERCIM news* 104, 1 (2016), 32–33.

[44] Matthew Lease. 2011. On quality control and machine learning in crowdsourcing. *Human Computation* 11, 11 (2011).

[45] Hongwei Li and Qiang Liu. 2015. Cheaper and Better: Selecting Good Workers for Crowdsourcing. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 3.

[46] Lihong Li, Michael L Littman, Thomas J Walsh, and Alexander L Strehl. 2011. Knows what it knows: a framework for self-aware learning. *Machine learning* 82, 3 (2011), 399–443.

[47] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. 2018. Delayed Impact of Fair Machine Learning. In *International Conference on Machine Learning*. 3150–3158.

[48] Zimo Liu, Jingya Wang, Shaogang Gong, Huchuan Lu, and Dacheng Tao. 2019. Deep reinforcement active learning for human-in-the-loop person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*. 6122–6131.

[49] Ziyin Liu, Zhikang Wang, Paul Pu Liang, Russ R Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. 2019. Deep gamblers: Learning to abstain with portfolio theory. In *Advances in Neural Information Processing Systems*. 10623–10633.

[50] Pedro Lopez-Garcia, Antonio D Masegosa, Eneko Osaba, Enrique Onieva, and Asier Perallos. 2019. Ensemble classification for imbalanced data based on feature space partitioning and hybrid metaheuristics. *Applied Intelligence* 49, 8 (2019), 2807–2822.

[51] David Madras, Toni Pitassi, and Richard Zemel. 2018. Predict Responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*. 6147–6157.

[52] Natalia Martinez, Martin Bertran, and Guillermo Sapiro. 2020. Minimax Pareto fairness: A multi objective perspective. In *International Conference on Machine Learning*. PMLR, 6755–6764.

[53] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635* (2019).

[54] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of Machine Learning*. MIT press.

[55] Hussein Mozannar and David Sontag. 2020. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*. PMLR, 7076–7087.

[56] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. 2018. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1802.00682* (2018).

[57] An Thanh Nguyen, Byron C Wallace, and Matthew Lease. 2015. Combining crowd and expert labels using decision theoretic active learning. In *Third AAAI conference on human computation and crowdsourcing*.

[58] Safiya Umoja Noble. 2018. *Algorithms of oppression: How search engines reinforce racism*. NYU Press.

[59] Besmira Nushi, Adish Singla, Anja Gruenheid, Erfan Zamanian, Andreas Krause, and Donald Kossmann. 2015. Crowd Access Path Optimization: Diversity Matters. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 3.

[60] Alexandra Olteanu, Kartik Talamadupula, and Kush R Varshney. 2017. The limits of abstract evaluation metrics: The case of hate speech detection. In *Proceedings of the 2017 ACM on Web Science Conference*. 405–406.

[61] Genevieve Patterson, Grant Van Horn, Serge Belongie, Pietro Perona, and James Hays. 2013. Bootstrapping Fine-Grained Classifiers: Active Learning with a Crowd in the Loop. In *NeurIPS Workshop on Crowdsourcing: Theory, Algorithms and Applications*.

[62] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[63] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. 2017. On fairness and calibration. In *Advances in Neural Information Processing Systems*. 5680–5689.

[64] Forough Poursabzi-Sangdeh, Samira Samadi, Jennifer Wortman Vaughan, and Hanna Wallach. 2020. A Human in the Loop is Not Enough: The Need for Human-Subject Experiments in Facial Recognition. In *CHI Workshop on Human-Centered Approaches to Fair and Responsible AI*.

[65] Chenxi Qiu, Anna C Squicciarini, Barbara Carminati, James Caverlee, and Dev Rishi Khare. 2016. Crowdselect: Increasing Accuracy of Crowdsourcing Tasks through Behavior Prediction and User Selection. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 539–548.

[66] Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. 2019. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220* (2019).

[67] Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Bobby Kleinberg, Sendhil Mullainathan, and Jon Kleinberg. 2019. Direct uncertainty prediction for medical second opinions. In *International Conference on Machine Learning*. 5281–5290.

[68] Inioluwa Deborah Raji, Timnit Gebru, Margaret Mitchell, Joy Buolamwini, Joonseok Lee, and Emily Denton. 2020. Saving face: Investigating the ethical concerns of facial recognition auditing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 145–151.

[69] 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.

[70] Eliza Strickland. 2018. https://spectrum.ieee.org/computing/software/aihuman-partnerships-tackle-fake-news

[71] Carolin Strobl, James Malley, and Gerhard Tutz. 2009. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods* 14, 4 (2009), 323.

[72] Andrew Sutton, Reza Samavi, Thomas E Doyle, and David Koff. 2018. Digitized trust in human-in-the-loop health research. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 1–10.

[73] Jinzheng Tu, Guoxian Yu, Carlotta Domeniconi, Jun Wang, Guoqiang Xiao, and Maozu Guo. 2020. Multi-label Crowd Consensus via Joint Matrix Factorization. *Knowledge and Information Systems* 62, 4 (2020), 1341–1369.

[74] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. 2014. Community-based Bayesian Aggregation Models for Crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*. 155–164.

[75] Bryan Wilder, Eric Horvitz, and Ece Kamar. 2020. Learning to Complement Humans. (2020).

[76] Yan Yan, Romer Rosales, Glenn Fung, and Jennifer G Dy. 2011. Active learning from crowds. In *International Conference of Machine Learning*.

[77] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 1415–1420.

[78] Dirk A Zetzsche, Douglas W Arner, Ross P Buckley, and Brian Tang. 2020. Artificial Intelligence in Finance: Putting the Human in the Loop. (2020).

[79] Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 295–305.

[80] Zhi-Hua Zhou and Xu-Ying Liu. 2010. On multi-class cost-sensitive learning. *Computational Intelligence* 26, 3 (2010), 232–257.