Jury Learning: Integrating Dissenting Voices into Machine Learning Models

Mitchell L. Gordon
Stanford University
Stanford, USA
mgord@cs.stanford.edu

Michelle S. Lam
Stanford University
Stanford, USA
mlam4@stanford.edu

Joon Sung Park
Stanford University
Stanford, USA
joonspk@stanford.edu

Kayur Patel
Apple Inc.
Seattle, USA
kayur@apple.com

Jeffrey T. Hancock
Stanford University
Stanford, USA
hancockj@stanford.edu

Tatsunori Hashimoto
Stanford University
Stanford, USA
tatsu@cs.stanford.edu

Michael S. Bernstein
Stanford University
Stanford, USA
msb@cs.stanford.edu

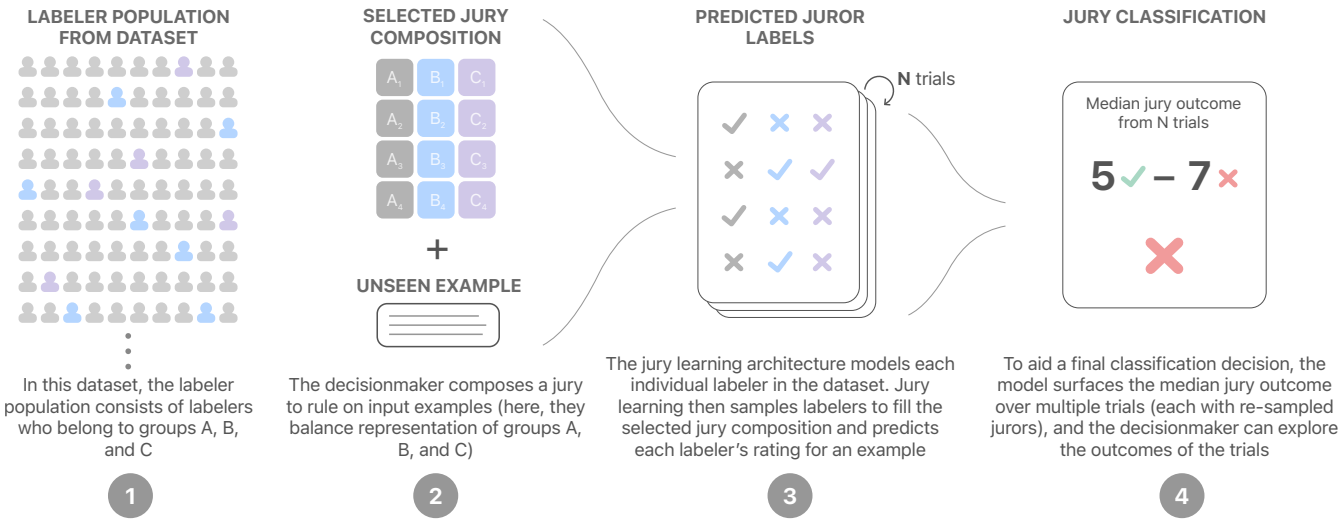

Figure 1: An overview of jury learning. (1) Given a dataset annotated by labelers from different groups, (2) the machine learning practitioner can compose a jury to rule on an unseen input example by allocating seats to labelers from the dataset with specified characteristics. (3) Then, the jury learning architecture models each individual labeler in the dataset, and performs *N* trials in which it samples labelers as jurors to populate the specified jury composition and predicts each juror's decision for the example. (4) The system then outputs a median-of-means jury outcome alongside jury outcome exploration visualizations that the decisionmaker can use to reach a classification decision.

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3502004

ABSTRACT

Whose labels should a machine learning (ML) algorithm learn to emulate? For ML tasks ranging from online comment toxicity to misinformation detection to medical diagnosis, different groups in society may have irreconcilable disagreements about ground truth labels. Supervised ML today resolves these label disagreements *implicitly* using majority vote, which overrides minority groups' labels. We introduce *jury learning*, a supervised ML approach that resolves these disagreements *explicitly* through the metaphor of a jury: defining which people or groups, in what proportion, determine the classifier's prediction. For example, a jury learning model

for online toxicity might centrally feature women and Black jurors, who are commonly targets of online harassment. To enable jury learning, we contribute a deep learning architecture that models every annotator in a dataset, samples from annotators' models to populate the jury, then runs inference to classify. Our architecture enables juries that dynamically adapt their composition, explore counterfactuals, and visualize dissent. A field evaluation finds that practitioners construct diverse juries that alter 14% of classification outcomes.

ACM Reference Format:
Mitchell L. Gordon, Michelle S. Lam, Joon Sung Park, Kayur Patel, Jeffrey T. 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 (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3491102.3502004

1 INTRODUCTION

Whose voices—whose labels—should a machine learning algorithm learn to emulate? In supervised machine learning today, the answers to these questions are often left *implicit* in the data collection and training procedure. In a typical procedure, the practitioner pays multiple annotators to label each example [38], then aggregates those labels via majority vote [50, 77] into a single ground truth label [61, 80]. The algorithm then trains on this aggregated ground truth, learning to predict ground truth labels that represent the largest group's point of view.

While this majoritarian [55] procedure succeeded for many early machine learning tasks, it now runs aground on tasks where there is substantial disagreement on what the correct label ought to be [37]. Tasks with substantial disagreement are common in user-facing contexts, including classification of online comment toxicity [35, 83], news misinformation [5, 91], and medical diagnosis [72]. In these tasks, up to *one third* of expert annotators disagree with each other when labeling an average example. Properly accounting for labels from non-majority groups in a comment toxicity task, for example, reduces classifier performance from 0.95 ROC AUC—nearly solved—to a much less persuasive 0.73 ROC AUC [37]. This less persuasive number is indicative of the fact that it is impossible to create a classifier that makes every user happy—we have to make a choice.

Today's supervised learning approach, however, does not afford the technical or interactive tools necessary to resolve annotator disagreements through an *explicit*, carefully considered choice. One response is to train the model to output a distribution across annotators rather than across classes [56, 64, 65, 85, 90]—e.g., "40% of annotators will say this comment is toxic, and 60% will not." However, for an HCI researcher or practitioner who is designing a classifier that must make decisions in the face of disagreement, the quantity of interest is rarely just a question of how many people disagree, but one of *who* disagrees and why [92]. Reflective practices around dataset generation [34] can help specify whose voices a classifier should be designed to emulate during the dataset collection stage. However, once a dataset has been collected and the resulting model trained, today's supervised learning pipeline does not afford the ability to reason over disagreement and then change a classifier's voices as tasks change or culture shifts. In most cases we lack even awareness of the need to do so: practitioners are typically unaware of whether stakeholders for a particular deployment or inference will disagree with a classifier's decisions, because they haven't modeled every annotator's or group's opinions. There remains a gap in providing algorithmic and interactive mechanisms that resolve the who, why, and decision rules of machine learning under societal disagreement.

In this paper we introduce *jury learning*, a supervised learning architecture that closes this gap through the metaphor of a jury. Jury learning models every individual annotator in the dataset, enabling the practitioner to declaratively define which people or groups from the training dataset, in what proportion, should determine the classifier's prediction. The jury learning model architecture then predicts each juror's label and outputs the joint jury prediction to classify unseen examples. Rather than a typical machine learning classifier outputting a label of, for example, toxic or not toxic, a jury learning classifier might output a prediction such as, "For this jury of six men and six women, which is split evenly between White, Hispanic, AAPI, and Black jurors, 58% of the jury are predicted to agree that comment is toxic." Through jury learning, practitioners can define jury compositions that reflect stakeholders for the task, for example that the toxicity classifier should centrally feature women and Black jurors because they are commonly targets of online harassment [58, 66]. The jury can articulate specific individuals, or any group-based annotation in the dataset (e.g., gender identity, political affiliation, racial identity).

To make a prediction on a new input, jury learning samples jurors from the practitioner's articulated jury composition, predicts each juror's response to the new input, then aggregates those responses into a final prediction. Our jury learning exploratory interface (Figure 2) visualizes how each juror voted, enabling sensemaking about the nature of disagreement on an input or set of inputs. This approach reconsiders the annotators who label training datasets not as inputs to an aggregation function but as a population of potential jurors. To ensure that no groups are represented as singular and monolithic in their opinions, jury learning does not model groups but instead individual jurors. This model architecture enables visualizations that highlight where each sampled juror falls relative to the distribution of all annotators in that group.

We contribute algorithms and visualizations that enable jury learning, then demonstrate them on a popular user-facing task of toxicity detection. The core technical challenge: how do we achieve jury-based prediction from a dataset of similar size and scope as those already in use today, and without abandoning the architectures that make modern machine learning models highly performant? We introduce a model architecture that combines state of the art natural language processing pipelines with techniques drawn from recent advances in deep learning based recommender systems [62, 86]. This joint model architecture trains the algorithm to predict how every individual person in the training data would label previously unseen examples, much like a movie recommender system might model how a user would react to a movie—but with the added challenge that every inference is on an example previously unseen by anyone in the training data. Our architecture enables this prediction task, and in addition enables visualization of the uncertainty underlying each decision—how many juries with the

same composition would have ruled differently?—while highlighting differences between groups' predicted labels. It also facilitates highly expressive jury-based algorithms, for example those that conditionally adapt the jury composition based on the relevant stakeholders for the input (e.g., populating with religious groups when the questionable comment is about religion, and political groups when the questionable comment concerns politics). In addition, by adapting techniques from quadratic programming, we demonstrate that developers can understand how jury composition impacts classifier behavior through counterfactual juries: automatically identifying the smallest change to the jury composition that would reverse a decision.

In an evaluation, we test whether jury learning changes which groups influence classifications of a machine learning algorithm for toxicity. Moderators of online communities (N=18) were asked to author juries for a comment toxicity classification task. We find that the resulting juries contain 2.9 times the representation of non-White jurors and 31.5 times the representation of non-binary jurors compared to those created implicitly by a large toxicity dataset [49]. This increased diversity in the jury composition changed the algorithm's classifications on 14% of items, reflecting the fact that jury learning captured those individual jurors' views far better than a baseline, state of the art aggregated model (with an MAE of 0.62 versus 1.05). We further find that our model architecture is *more* accurate at predicting aggregate test set labels (MAE=0.27) than today's state of the art classifiers (MAE=0.41). This finding, which highlights the inherent instability of ground truth in the standard aggregate labeling approach, means that our model architecture both enables highly performant jury learning verdicts and also offers performance gains in the traditional aggregated task. Both of these are achieved by modeling each individual annotator whose opinions make up an aggregate label or jury verdict.

Taken together, this work contributes algorithms and interfaces for a machine learning architecture that makes explicit the selection of whose voice, with what weight, determines each prediction. We argue for this approach normatively, demonstrate its predictive accuracy, and produce evidence that practitioners' jury learning classifiers result in material changes in classifier behavior.

2 RELATED WORK

In this section, we motivate jury learning through an integration of research in human-computer interaction—especially social computing—along with work in machine learning and AI fairness.

2.1 Engaging stakeholders in algorithm design

Our work draws on a critique of strict and unexamined *majoritarianism* in governance [55], which tends to exclude the viewpoints of minority groups [20]. To protect minority rights, governance structures in practice typically include mechanisms that help avoid a tyranny of the majority (e.g., a bicameral structure [69]). How should machine learning practitioners respond to this challenge? Jury learning presents one possible response, in which we introduce new levers that enable explicit control of how majorities are formed. In doing so, jury learning raises awareness of each potential majorities' consequences and encourages intentionality in their selection. We argue that in the hands of a well-intentioned actor, jury learning represents meaningful progress towards the problems that strict, unexamined majoritarianism can bring in machine learning. Doing so also opens opportunities for participatory and democractic approaches to jury selection.

Researchers in human-computer interaction and artificial intelligence have long articulated the need for algorithms to balance multiple stakeholders' needs, motivations, and interests, and to help achieve important collective goals [2, 12, 18, 25, 60, 78, 88, 92]. One such thread, stemming from ethics in AI, focuses on ensuring fairness of outcomes. It demonstrates how machine learning training algorithms can enforce mathematical notions of individual [29] and group [3, 10, 39] fairness in classification tasks such as recidivism prediction. We build on advances in algorithmic fairness that help manage disparate *outcomes* [59], by contributing a technique that instead helps manage disparate *beliefs*: whose labels we should be learning when there are irreconcilable disagreements among groups in society. For instance: in today's fairness approaches, the developer may normatively decide what a fair outcome looks like: e.g., comments submitted by Black users should be removed just as frequently as comments submitted by White users. Our work focuses on an orthogonal aspect of fairness: while disparate outcomes might focus on how many of these comments should be removed from different groups of users, we ask *whose voices* should be involved in the decision of whether a comment should be removed.

Closer to our aims, a second thread of work proposes design guidelines and frameworks to help system designers ensure they are creating algorithms that reflect their stakeholders' values [68, 88, 92]. These design processes argue for explicit inclusion of appropriate stakeholders in the design and evaluation of the algorithm. For instance, in WeBuildAI [53], stakeholders design their own models representing their beliefs, and then a larger algorithm uses each of these models as a single vote when making a decision for the group. We agree that stakeholders' voices should be directly modeled in algorithmic systems. We contribute a jury-based metaphor, along with a model architecture and algorithms designed to empower practitioners to explicitly resolve disagreements between stakeholders while retaining the performance of today's machine learning pipeline.

In creating our approach, we draw on recent work that adopts a civics and governance metaphor for socio-technical design. Contested platform decisions can be made by juries of platform members, which can increase the perceived legitimacy of the decisions [32, 48]. Platforms such as Facebook have recently engaged such models for setting decision-making precedent, as in their Oversight Board [47]. The PolicyKit toolkit demonstrates how such participatory processes can be encoded directly into the software that powers these platforms [89]. Our work extends these metaphors to demonstrate their power in fully algorithmic environments as well, where they offer legitimacy and interpretability benefits.

2.2 Disagreement, datasets, and machine learning

Across tasks such as identifying toxic comments [35, 83], bot accounts [84], and misinformation [91], researchers and platforms [45]

increasingly turn to machine learning to aid their efforts [36]. Specifically, these models are often trained using a supervised learning pipeline where we:

(1) Collect a large dataset of individual beliefs, either generated through crowdsourcing services that ask several labelers to annotate each item according to policy and then aggregate the result into a single ground truth label (e.g., [21]), or similarly by asking and then aggregating experts (e.g., [82]).
(2) Use those ground truth labels to train a model that produces either a discrete binary prediction or a continuous probabilistic prediction for any given example.

For instance, in a Kaggle competition that received over 3,000 submissions, researchers were challenged to discover the best-performing architecture in a toxicity detection task [44]. Facebook makes the vast majority of moderation decisions through classifiers [11], and YouTube does similarly [14].

Classifiers typically speak with one voice, an aggregated pseudo-human that reflects the majority voice in the dataset they have been trained on [34, 70, 73]. This majority-voice outcome can arise for two reasons: (1) majority vote aggregation of the raw crowdsourced annotations overrides minority viewpoints in generating ground truth, or (2) even if training data points are disaggregated, the training algorithm minimizes its loss function by predicting accurately for the opinions held by the largest group of people in the dataset. Unfortunately, while this majority-voice approach to classification has been highly successful in many artificial intelligence (AI) tasks such as image classification [21], the results for many tasks in social computing and HCI remain problematic.

One potential explanation for these problems may be that the voice a model has learned is not the right voice for every deployment, or even every inference within a deployment. To see how this might be true, we can examine annotator disagreement rates in today's datasets: for instance, in a toxicity task, over one third of annotators on average disagree with any toxic classification, even after accounting for label noise [37]; in a misinformation classification task, three professional fact checkers were unanimous on only half of URLs [5]. Across countries, content that was perceived as more or less harmful varies significantly [43]. Such disagreement indicates that there may be multiple competing voices, potentially representing different groups of people or sets of values. Indeed, a toxicity model tuned with a simple positive or negative offset (i.e. baseline) for each annotator achieves far more accurate per-annotator results than a standard classifier [49].

We build on research that aims to accurately capture the distribution of annotator opinions [17, 19, 26–28, 46]. Given a dataset with individual annotator labels, machine learning researchers have begun training models to output a distribution of labels rather than a single class label, using loss functions such as cross-entropy compared to the distribution of annotators' labels [56, 64, 65, 85, 90]. While training models with cross-entropy loss acknowledges the existence of disagreement, it does not tell us *who* disagrees or why, so we cannot readily act on it. An alternative approach, annotator-level modeling, has been shown to yield benefits to uncertainty estimation and majority vote prediction [19]. In this work, we introduce an annotator-level modeling architecture in the service of the decision rules underlying jury learning. As support for our approach, a Wizard of Oz study found that moving beyond raw distributions and towards AI-provided arguments for competing options resulted in users reviewing more contentious cases themselves [73].

Dataset documentation [34] and value-sensitive data collection practices [92] can help specify whose voices a classifier should be designed to emulate. We build on these approaches in two ways. First, we provide an algorithm that makes clear when these voices disagree and provides tools to reflect on and re-weight whose voices are embedded in the model. From this perspective, our work innovates on this literature by directly modeling this information, allowing the machine learning practitioner to understand the nature of the disagreement and make explicit the representation that should resolve it. Second, our work addresses a practical reality of machine learning: while we cannot possibly have a universal set of voices that are appropriate for all models in a given task such as toxicity detection, existing approaches assume practitioners have the resources and motivation to collect new large-scale datasets every time the relevant stakeholders change. In reality, even in the rare cases in which practitioners have the required resources to collect new datasets, they are often unaware of the need to do so: we cannot know whether stakeholders will disagree with a classifier's decisions unless we've modeled every annotator's or group's opinions, leaving many practitioners unaware of the extent to which they are ignoring the opinions of certain annotators or groups. It is therefore not sufficient to have a procedure that requires that requires prior knowledge of the optimal annotator population at the outset. We contribute an approach that can model each relevant individual or group from a dataset similar in size and scope as those already collected today, so that practitioners can reason over and specify which of these individuals or groups their model should and should not reflect, iteratively and reflexively.

A large body of work in both HCI and machine learning discusses how improved dataset collection practices may result in more performant and ethical classifiers. Often, dataset authors instead strive for a goal of impartiality, so that data is supposedly "unbiased" [75]. To achieve such a goal, crowdsourcing researchers have proposed a number of methods that aim to resolve annotator disagreement either by making task designs clearer or relying on annotators to resolve disagreement among themselves [13, 15, 26, 57, 74]. However, for tasks such as those common in social computing contexts, much of the disagreement is likely irreducible [37, 46, 67], stemming from the socially contested nature of questions such as "What does, and doesn't, cross the line into harassment?". The above methods may help resolve some disagreement in these datasets, but until such an unlikely time as there is ever to be a global consensus on questions such as what constitutes harassment, classifiers must make decisions that represent some users' voices more than others'. Jury learning offers one approach to this decisionmaking.

An alternative approach is to retrain a model's single voice to represent a desired group [4, 9, 33]. If this decision can be made effectively up front, and the practitioner has the substantial budget and resources required to collect their own large-scale dataset, then a single data collection and training pipeline can succeed. Jury learning contributes an approach that allows real-time exploration and tuning of this population without requiring practitioners to

collect new and far larger datasets, and makes stronger guarantees about whose voice is being represented in each specific inference.

2.3 Interactive Machine Learning

Our work draws on a recent thread of research integrating human-centered methods into machine learning systems. Interactive machine learning seeks methods to integrate human insight into the creation of ML models (e.g., [7, 31]). One general thrust of such research is to aid the user in providing accurate and useful labels, so that the resulting model is performant [15]. Another line of work has sought to characterize best practices for designers and organizations developing such classifiers [6, 8]. Our work extends this literature, focusing on ameliorating issues that developers and product teams face in reasoning about their models and performance [63].

A third line of works demonstrates that end users struggle to understand and reason about the resulting classifiers. Many are unaware of their existence [30], and many others hold informal folk theories as to how they operate [23]. In response, HCI researchers have engaged in algorithmic audits to help hold algorithmic designers accountable and make their decisions legible to end users [71]. Our work extends this literature, positioning classifiers as a reflection of many different voices, enabling control over that composition of voices, and enabling both practitioners and users to easily understand which voices are contributing to their models.

3 JURY LEARNING

Machine learning often aims to emulate people's labels. Faced with annotator disagreement representing multiple competing voices, which people should we be emulating—whose training labels should a classifier use to make its decisions? We take the position that it is the machine learning practitioner's responsibility to make explicit normative decisions about whose voices their classifiers are reflecting in any given inference. In this section, we describe how we designed jury learning and our motivation in making each of these design decisions.

3.1 Design goals

We begin by considering today's approach. Many models return the class label or labels with the highest likelihood (e.g., label = 'toxic', confidence .9). Some models instead predict the distribution of opinions over all annotators: for instance, that 60% of annotators will label a comment as toxic, 30% will label a comment as non-toxic, and 10% will label as unsure. How is a practitioner to act on this information? If their goal is always to satisfy the largest number of annotators, the answer is easy. However, there are many scenarios in which that is not—or should not—be the goal. The practitioner may want to consider different voices (representing different values, experiences, or expertise) depending upon the situation. Consider that a member of the LGBTQ+ community may be more informed about transphobic comments than the population at large. When a comment targets LGBTQ+ issues, or if a community is centered on supporting LGBTQ+ members, a practitioner may wish to more heavily weigh the opinion of these annotators. Or consider that when doctors labeling MRI data disagree about a patient's diagnosis, the practitioner may wish to more heavily weigh opinions from doctors with a particular background or training. Or, it may be the case that the practitioner isn't initially sure who to side with, and so would like to reason over the different decisions that different annotators or groups of annotators would make.

It is, in theory, possible to achieve some of these goals using today's standard supervised learning pipeline. For instance, a practitioner deploying a classifier to the LGBTQ+ community could collect their own dataset, ensuring that a sufficient portion of annotators identify as LGBTQ+ so that disagreements are resolved by more heavily weighing opinions from LGBTQ+ annotators. In practice, however, such an approach fails to meet our goals. Datasets are expensive and difficult to collect, so practitioners often rely on existing datasets they did not collect, meaning they do not control how disagreements are resolved, and worse: *do not even know that voices they care about are dissenting*. Without such knowledge, practitioners cannot reason over the different decisions that different annotators or groups of annotators would make. We require a different approach from today's standard supervised learning pipeline.

3.2 Approach and interaction

Jury learning is a supervised learning approach that asks practitioners to specify whose voices their classifiers reflect, and in what proportion. To achieve this, jury learning models every individual annotator in a dataset, so that their model may serve as a potential juror. Practitioners then articulate a set of jurors that should be sampled from the groups or individuals in the annotators. That jury's labels determine the classifier's behavior.

For our purposes, we refer to a *jury* as a bounded set of individuals whose opinions aggregate into a decision. These individuals are randomly sampled from the population of labelers based on the jury composition that the machine learning practitioner has articulated (e.g., six conservative jurors and six liberal jurors). Jury learning then algorithmically predicts how each of these twelve selected jurors would label the input, and then aggregates those responses into a decision. For many of our examples, we refer to a twelve person jury, which is the default jury size in the American legal system. However, the jury can be any size, if there are enough annotators in each group in the dataset to populate it. If the task is regression rather than classification (e.g., a toxicity score rather than a binary toxic-or-not decision), the outcome is an average of jurors' predicted scores.

Jury learning enables the creation of many possible classifiers from a single dataset of labels, with the added requirement that the dataset contain information about each annotator for any group or voice that the practitioner wishes to include. For instance, the toxicity dataset we use as our example application domain [49] includes education, past work experience or qualifications, racial identity, gender identity, political affiliation, age, disability status. Practitioners specify jurors either individually, or using any group membership criteria available in the dataset. If a practitioner selects a juror using group-based data, we demonstrate where that juror fits within the full distribution of all annotators within that group, ensuring that no group is represented as monolithic. Practitioners can interactively explore different jury compositions, gaining an understanding of the consequences of each composition that they

try: how are specific annotators or groups of annotators differing in their labels?

Figure 2 displays the jury learning interface for our example application domain of toxicity detection. In Figure 2(A), practitioners specify a jury composition by assigning a *juror sheet* to each of twelve empty *juror slots*. A juror sheet defines the characteristics of the annotator who will fill the juror slot. A simple juror sheet may specify only one characteristic, such as a juror identifying as Black, while a more complex juror sheet may articulate an intersectional identity, such as a juror identifying as a Black LGBTQ+ woman. The possible characteristics are dictated by the provided dataset: the set of identities must be broad enough to reflect the relevant stakeholders [76]. If a characteristic is better captured through open ended text boxes than categories [76], the practitioner could explore individual people in the dataset and select a subset for inclusion. The jury composition can be defined interactively via a web interface. The interface also allows the machine learning practitioner to explore different jury compositions and how each might react to different inputs.

The machine learning practitioner can then apply their jury to an input or set of inputs. Given an input to predict, jury learning makes a prediction for that input for every annotator. It then takes a step not possible when convening real-world juries, but possible with jury learning: it convenes many parallel iterations of the jury, by repeatedly resampling a large number of juries that match the jury specification. Each jury may contain different jurors (annotators from the dataset), and the model will predict different responses to the input for each juror based on that juror's training data in the dataset. The interface disallows selecting any groups with an insufficient number of jurors in the dataset to complete the resampling procedure without replacement, directing practitioners to collect more data for the particular group.

The system then responds with a jury verdict for the input: the single, final decision of the median jury on that input, shown in Figure 2(B). To identify a verdict, the system samples a set of individual jurors filling the jury specification, predicts each juror's decision, and then determines the aggregate verdict taken as a majority vote (for classification) or mean (for regression). The default decision is calculated as the median jury decision from the set of sampled juries matching the jury specification. This median-of-means estimator [51]—the median of the mean juror responses across juries—produces an estimate that is robust both to variance within groups and to potential juror-level modeling errors by the AI. In particular, this approach is resistant to the model being wrong about any small number of jurors, though less effective for systematic errors that may impact most or all jurors.

The approach also results in a direct measure of uncertainty: how often the outcome changed across the jury samples. For example, the system might communicate that 85% of juries matching the specification resulted in a "toxic" label, and 15% of juries resulted in a "not toxic" label. Or, for a regression task, it might communicate a histogram distribution of jury decisions, as shown via the histogram in Figure 2(B).

Because the system returns a specific jury, the system can visualize each juror in context of the group from which they were sampled (Figure 2(C)). This contextual information helps the machine learning practitioner better understand the behavior of the jurors chosen for their jury. Specifically, for each juror, we make available all of their annotations and all associated background information that is present in the dataset. This visualization also helps make clear that different members of a group may vote differently, and that despite this individual variation, the overall jury outcome may be stable. In addition, our approach grounds the jurors as individual people with specific characteristics and enables other explainability-related information, such as highlighting how the juror labeled similar inputs in the training data or providing their specific modeling error rate over all of their test examples 2(D)).

The system is interactive to encourage better sensemaking, but it also provides a code layer for automated systems. The jury definition can be passed as a Python dictionary object, as in Figure 3. The response likewise is returned as a Python dictionary object, as in Figure 4.

3.3 Example scenario

Saanvi has created an online news-sharing social network, and wants to create a classifier to detect any instances of personal attacks on the platform. She finds a popular, publicly available large-scale dataset, trains a model using the traditional supervised learning pipeline, and deploys it to her community. The classifier takes as input the text of a comment, and returns a "toxic" or "not toxic" label. Unfortunately, Saanvi soon begins to notice that both she and many members of her community often disagree with the decisions this classifier makes. Saanvi suspects that perhaps her classifier isn't making decisions in ways that reflect the voices in her community.

Saanvi's dataset contains characteristics about each annotator, so she switches from the traditional classifier to a classifier created through jury learning. First, Saanvi explores different jury configurations to confirm any group-based differences that she expects to see, inputting comments and exploring how the jurors in each group respond. She confirms that men are more likely to rate borderline comments as not toxic, but notes that there are many women on her platform. By exploring, she also observes that seniors find more comments to be toxic, and 18–35 year olds find fewer comments to be toxic. Saanvi begins by constructing a jury that she believes better represents the members and values of her community. She deploys a private test of it, and notices a significant improvement: the classifier's decisions start making a lot more sense to her and her community members. Saanvi then begins a participatory process to bring in stakeholders from her community, allow them to test different jury configurations, and agree upon a jury to use on their platform.

Saanvi and the other stakeholders observe that their intuitions of the proper jury composition change based on which groups might be targeted in that post: that when a news article is about women's issues, they want more women on the jury; when a news article concerns LGBTQ+ rights, they want more jurors identifying as LGBTQ+; when an article is about a Black woman, they want more Black women on the jury. So, they agree to dynamically allocate four seats on the jury to the appropriate group based on the news category that the post is shared in (e.g., four women for news articles shared in the womens' rights category).

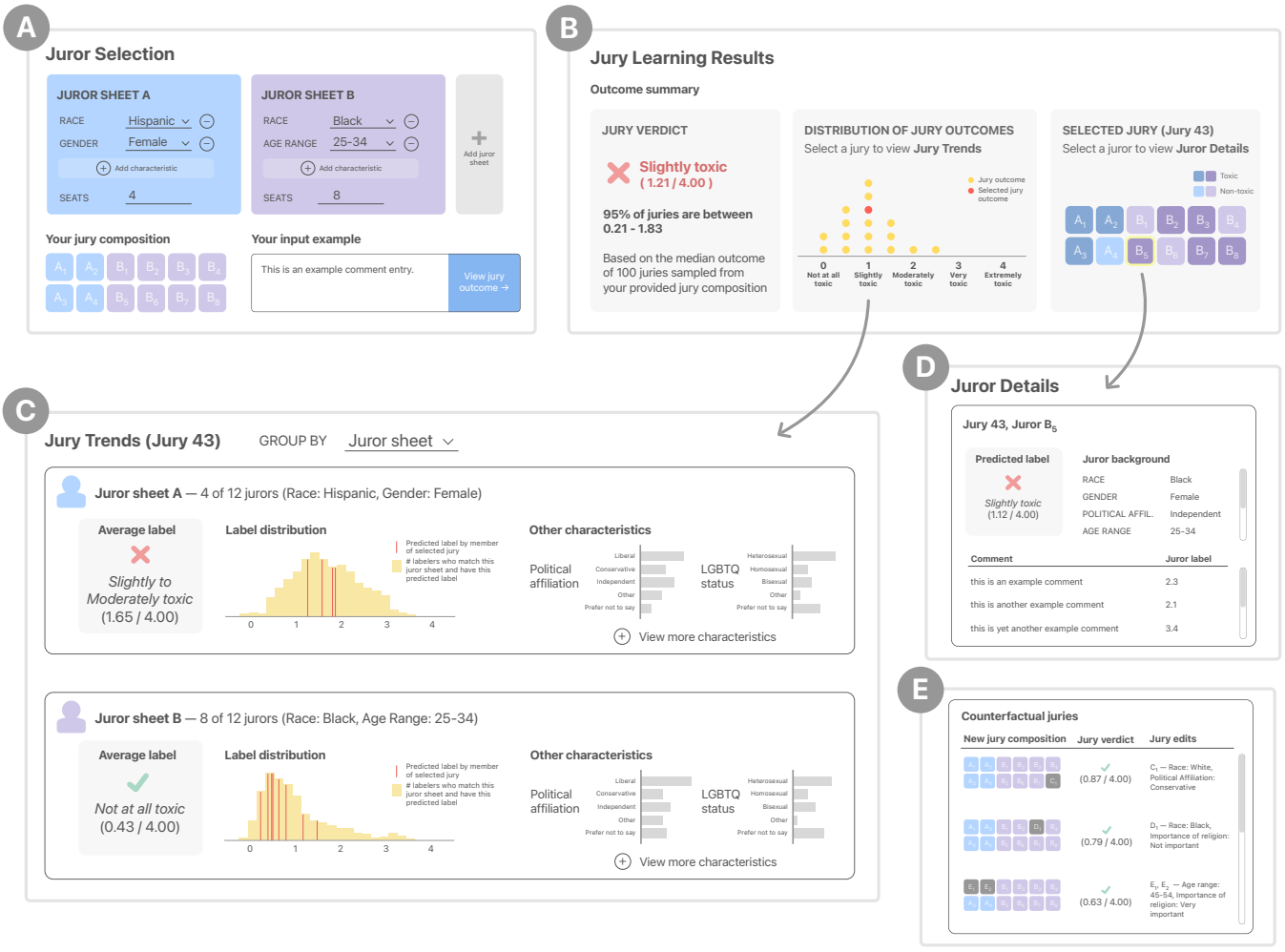

Figure 2: System Overview. (A) In the Jury Selection portion of the system, the user can create juror sheets to populate their jury composition and can provide one or more input examples to evaluate. (B) Then, the system outputs the Jury Learning Results section where they can view a summary of the jury verdict based on a median-of-means estimator of jury outcomes. Here, they can view the full distribution of jury outcomes, select individual juries to view trends, and inspect individual jurors on a jury. (C) When a user selects a jury, the Jury Trends section is updated. There, they can group by different fields like the juror sheet, decision label, or other demographic attributes to understand patterns in the labels from this jury and contextualize them with respect to the larger population. (D) When a user selects a particular juror, the Juror Details view opens, and they can inspect the predicted label for the juror, the background of this juror, and the juror's annotations. (E) Users can also inspect counterfactual juries that would result in the opposite verdict.

Saanvi exports the model and puts it into private testing on her server, where its predictions are not yet shown to users. She and the group of stakeholders continue to monitor its behavior. Eventually, as they build trust in the algorithm, they begin to use it to prioritize comments for human moderators on the platform.

4 TECHNICAL APPROACH

Jury learning requires that we predict how each individual annotator would label an unseen example. A jury outcome is then an aggregation of the jurors' (annotators') individual classifications.

Enabling the broadest set of applications also requires an approach that can make such predictions from a dataset of similar size and structure to those already in use when training supervised standard classifiers: a labeled dataset with a few annotators labeling each item and each annotator labeling a few items, such as those commonly acquired from crowdsourcing services. The only additional assumption we make is that any characteristic used to select jurors (e.g., gender identity) must exist for each juror. This is achievable by adding a small survey when an annotator begins labeling examples. In what follows, we describe our model architecture for jury learning. While we focus our description on

```
jury = [
    {
        'jurors': 4,
        'gender_identity': 'female',
    },
    {
        'jurors': 4,
        'gender_identity': 'nonbinary',
    },
    {
        'jurors': 4,
        'gender_identity': 'male',
    }
]
```

Figure 3: The jury definition can be passed as a Python dictionary object.

```
result = {
    'verdict': 'toxic',
    'votes': {
        'toxic': .67, # 8 of 12 jurors voted 'toxic'
        'nontoxic' .33, # 4 of 12 jurors voted 'nontoxic'
    },
    'jurors': [
        {
            'juror_id': 1023,
            'gender_identity': 'female',
            'racial_identity': 'White',
            'political_affiliation': 'liberal',
            'vote': 'toxic',
        },
        {
            'juror_id': 2342,
            'gender_identity': 'female',
            'racial_identity': 'South Asian',
            'political_affiliation': 'conservative',
            'vote': 'nontoxic',
        },
        ...
    ],
    'population': {
        'toxic': .85, # 85% of sampled juries voted 'toxic'
        'nontoxic': .15, # 15% of sampled juries voted 'nontoxic'
    }
}
```

Figure 4: The response likewise is returned as a Python dictionary object.

natural language processing tasks (specifically, toxicity detection), the high-level architecture is general and can apply to any inputs that allow content embeddings (e.g., images via Resnet [81], screens via Screen2Vec [54], or text via BERT [24]).

We base our model architecture on the insight that, in trying to predict how each annotator would label an unseen example, we share part our goal with the aim of today's recommender systems. Like recommender systems, we must not only perform well over a range of inputs, but also over a range of individuals. Like recommender systems, we expect that different opinions between annotators can often be partly explained by explicit categorical information about the groups that each annotator belongs to or identify with, but are also partly unique to a particular annotator or explained by unobserved latent factors [40]. In other words, much like how Netflix might develop a model to predict individual users' opinions on films, our jury-based model will predict individual labelers' perspectives on new inputs.

Unlike Netflix, however, all of the inputs to our model will be unseen examples (or, in recommender systems language, all examples suffer from the *cold start problem*), meaning that they have never been seen by any annotators in our training set. This is a standard assumption in any classification task, but not a typical assumption of most recommender systems, which often rely heavily on an item's existing annotations to inform what other users will think of it. We require an approach that relies entirely on an input's featurization: by taking an input and embedding it, we can predict an annotator's label by comparing this input to similar examples they have already annotated. This means that today's hybrid deep learning recommender systems for natural language input, which typically train their own item embeddings [87], are insufficient. We propose a model architecture that jointly trains a content model for classification tasks (such as from BERT) alongside a deep recommender system. By combining deep recommender systems' ability to model individuals' opinions with modern pre-trained deep learning models' classification task performance, our architecture takes full advantage of the strengths of each.

For our recommender system architecture, we select a Deep & Cross Network (DCN) [86]. DCNs were designed for web-scale collaborative filtering applications in which data are mostly categorical, leading to a large and sparse feature space. While DCNs were created for classic recommender system tasks, our insight is that a modified DCN architecture is strong fit for jury learning. A typical DCN involves three sets of embeddings: content, annotator, and group. The content embedding enables prediction on previously unseen items by mapping those items into a shared space. The group embeddings make use of the data from all annotators who belong to each group, helping overcome sparsity in the dataset. The annotator embedding ensures that the model learns when each annotator differs from the groups they belong to. The DCN learns to combine these embeddings to predict each individual annotator's reaction to an example: the embeddings are concatenated into an input layer, then fed into a cross network containing multiple cross layers that model explicit feature interactions, and then combined with a deep network that models implicit feature interactions [86]. We modify the DCN architecture to jointly train (or more precisely in the case of a pre-trained models, jointly fine tune) a pre-trained BERT-based model, using its pooler output as the content embeddings. Figure 5 displays a high-level view of our end to end model architecture.

4.1 Implementation for toxicity detection

Having described our high-level approach and architecture, which can be applied to a wide range of tasks, we now turn to the specific task we use to demonstrate jury learning in this paper: toxicity detection.

4.1.1 Dataset description. We train our model using a publicly available balanced dataset [49] in which 107,620 social media comments were labeled by five annotators each, from a pool of 17,280 unique annotators. This dataset was collected to understand how user expectations for what constitutes toxic content differ across demographics, beliefs, and personal experiences. Each annotator labeled a minimum of 20 comments, with a small fraction labeling more than 20. Each annotator contained categorical information noting their self-identified gender, race, education, political affiliation, age, whether they're a parent, and whether they consider religion an important part of their lives. Annotators were asked to rate each social media comment's toxicity on a scale from 0 to 4,

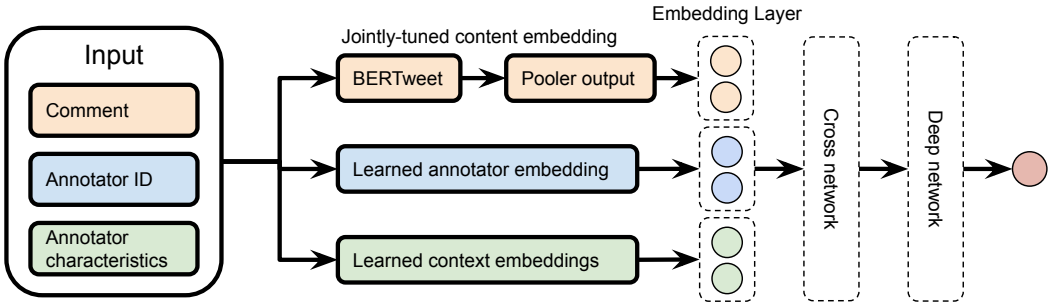

Figure 5: We introduce a model architecture that jointly fine tunes the practitioner's existing content-based classifier alongside a Deep & Cross Network recommender system.

with 0 being non-toxic, 1 being slightly toxic, and 4 extremely toxic. If we binarize this task, with a rating of < 1 indicating non-toxic and $>= 1$ indicating toxic, we find that annotators in this dataset disagree with each other 35.9% of the time.

4.1.2 Training. We use TensorFlow Recommenders (TFRS) as the basis of our implementation. TFRS natively supports DCNs. We use Huggingface's Tensorflow API to instantiate BERTweet (a large-scale language model pre-trained on English Tweets, released by NVIDIA [62]) as the pre-trained content embeddings within our recommender system. We adapt the model to the task by performing an initial fine-tuning step on a large-scale toxicity dataset released by Jigsaw [44].

Initially, we co-train all the model's components together: we fine tune the pre-trained large language model, and we train from randomly initialized values for the annotator embedding, group embeddings, and the DCN. However, while BERT-based models have been shown to quickly overfit after fine tuning for a few epochs, our newly initialized components can benefit from a longer training procedure. We therefore co-train the entire model for two epochs, freeze the large language model, and continue training the remainder of the model for 8 epochs. Further epochs did not noticeably improve the model's performance. We used the Adam optimizer and Mean Squared Error as our loss function.

We trained our model on one machine with one NVIDIA Titan XP GPU. The majority of the Titan XP's memory is taken up by BERTweet, so most of the DCN itself is stored in the machine's memory during training. We chose standard hyperparameters used when fine tuning BERT-based models: we used learning rate of $2e-5$, a batch size of 16, and a maximum length of 128 tokens. We set our DCN-specific hyperparameters as follows: we set a constant embedding dimension of 32, a three-layer cross network of size 768, three dense layers of size 768, and a output dense layer of size 1. We selected these sizes and the number of training epochs after performing a small grid search.

5 EXTENSIONS

The architecture of jury learning directly affords new decision-making and interpretability techniques that are not available with traditional algorithms. Here we overview two such techniques that we have implemented.

5.1 Conditional juries

We might desire different forms of expertise depending on the decision at hand. For example, CHI's peer review process identifies jurors (reviewers) who differ for each paper under review, based on the content of the paper. Likewise, civil society organizations convene different groups of stakeholders depending on who their decisions might impact. In the context of AIs, for example, classifying misogynistic comments may call for a jury with a larger representation of women, whereas classifying racist comments may call for a jury with a larger representation of minoritized racial groups.

While the default jury learning algorithm focuses on a simple metaphor of a stable jury composition that is used for all decisions, jury composition can be *conditional* on the item being classified. A simple code conditional might adapt the jury composition:

```
# Define a default six of the twelve jury members, allowing the other six to vary based on the context
default_jurors = [ ... ]

# select the other six jurors based on context
conditional_jurors = []
if '#metoo' in tweet:
    conditional_jurors = [
        {
            'jurors': 6,
            'gender_identity': 'female',
        }
    ]
elif '#blm' in tweet:
    conditional_jurors = [
        {
            'jurors': 6,
            'racial_identity': 'Black',
        }
    ]
elif ... # additional conditions and conditional jurors

# combine the default six jurors with the six jurors who have been selected for this context
jury = default_jurors + conditional_jurors
```

Alternatively, approaches such as clustering or topic modeling might be appropriate:

```
jury = [ ... ] # default jury

# embedding_distance calculates the comment's cosine similarity to comments that contain a given term
if embedding_distance('#blm', tweet) < .05:
    jury = ... # jury composition for Black Lives Matter topics
elif embedding_distance('#metoo', tweet) < .05:
    jury = ... # jury composition for MeToo topics
elif embedding_distance('vaccination', tweet) < .05:
```

```
    jury = ... # jury composition for vaccination topics
elif ... # additional conditional juries for specific topics that the community
↪  cares about
```

5.2 Counterfactual juries

When a jury decides a given comment to be non-toxic, it naturally gives rise to the question: what jury composition, if any, *would* find the comment to be toxic? How different would the jury need to have been to flip the outcome? Jury learning enables this interaction to search for a *counterfactual jury*, by automatically identifying the minimal change to the jury composition that would result in a different outcome than the current jury (Figure 2(E)).

Within the jury learning framework, we frame the search for the counterfactual jury as an optimization problem: flip the classification by making the smallest edit possible to the current jury composition. Formally, we can define this as a quadratic program, solvable via convex optimization. Consider that we have K different annotators or groups of annotators, and we have a prediction s_k associated with each. We set the size of our jury, n_{jurors}, to 12, meaning that we must assign a value in $\{0 \dots K\}$ to each of the 12 juror slots. To represent a jury composition, we define a jury allocation vector p of length K. Each index of p refers to an annotator or group in K, and the value at each index refers to the number of jurors from the corresponding annotator or group. The jury allocation vector should therefore should sum to n_{jurors}. The classification decision we consider is a threshold on a jury's average prediction, which we define as $v_p = \frac{\sum_k p_k s_k}{n_{jurors}}$. The final classification is based on whether $v_p > 1$. The problem of identifying a counterfactual jury is now equivalent to a quadratic program. If the current decision is in the negative class $v_p \leq 1$, then the counterfactual jury that flips this decision is defined as the solution to the following optimization problem

$$\min_{p^* \in \mathbb{Z}^+} \sum_k (p_k - p_k^*)^2$$

$$s.t.$$

$$\sum_k p_k^* = n_{jurors} \quad \text{and} \quad v_{p^*} > 1 \quad \text{and} \quad p_k^* \geq 0.$$

This optimization problem can then be solved by off-the-shelf optimization solvers.

Counterfactual juries can serve as a useful interpretability lens, aiding the community in understanding how dependent the classification outcome was on the jury composition.

6 MODEL EVALUATION

Taking our example application of toxicity detection, we evaluate the performance of our proposed model architecture at two levels:

(1) How accurate are individual juror predictions?
(2) How accurate are the final predictions produced by a jury?

The most important question to test with jury learning is whether the learning algorithm correctly estimates what jurors' opinions are on previously unseen data. Recommender systems make predictions across different individuals by identifying commonalities among annotators and borrowing information. Without an approach that sometimes borrows information, building a jury learning system would require acquiring a large dataset from each group, including each intersectional identity group, which is often infeasible. However, any machine learning approach that borrows information also brings a risk: it is possible to borrow *too much* information, particularly when we have less data from a specific group or annotator. So, our evaluation seeks to test whether the approach is correctly estimate each juror's labels.

6.1 Individual juror performance

6.1.1 Performance versus a standard classifier. We first demonstrate that jury learning is substantially more accurate in predicting individual annotator responses when compared to a baseline state-of-the-art, annotator-agnostic classifier.

To create a state of the art baseline model, we fine tune BERTweet on the toxicity dataset using the same standard hyperparameters we used to fine tune BERTweet within the jury learning algorithm. As the toxicity dataset provides a regression task in (0, 4), we report the Mean Absolute Error (MAE), comparing each individual annotator's predicted response to their observed response. We design the test set for this evaluation so that all of the comments were never seen by the annotators in the training set. This more challenging prediction task reflects the expected usage of our model, as discussed earlier. Our test set contains 5,000 comments and 24,545 annotations.

We find that our model achieves an MAE of 0.61, and the baseline model achieves an MAE of 0.90. This large improvement is not necessarily surprising: our architecture is the only one that makes use of information about individual annotators. This result demonstrates that our model was able to learn a substantial amount of useful information about each annotator or their groups; if jury learning had learned nothing about either an individual or groups, then its predictions would simply match those of a standard state of the art classifier trained on aggregated labels, which makes one prediction per example.

6.1.2 Performance versus a group-based classifier. The above performance gains could either have come from learning about individual annotators, the groups they belong to, or both. Our goal with jury learning is to ensure that models are not solely reliant on group membership; we would also like our model learn about how individual annotators diverge within their groups. We therefore now ask: how performant is our model at predicting individual annotators' responses to an example, compared to an ablation of our model that only knows about group membership? If our model performs better using both annotator and group information than solely group information, it has learned specific information about annotators.

To create a group-specific classifier, we train a model using our proposed architecture with one change: we remove annotator IDs as a feature, meaning that our model can only rely on group-based and content-based features. We find that this model achieves an MAE of 0.81. This score is an improvement over the baseline aggregated classifier's 0.90, indicating that our model learned useful information from group-based features. However, our full individual+group model's MAE of 0.61 is a substantial improvement over both, indicating that our full architecture is reliant on both group and individual annotator features.

	Full test set	Asian	Black	Hispanic	White	Male	Female
Number unique annotators	11262	817	1774	424	9087	6077	6985
MAE: Baseline aggregated model	0.90	0.83	1.12	0.87	0.87	0.94	0.86
MAE: Jury learning model	0.61	0.62	0.65	0.57	0.60	0.61	0.60

	Liberal	Independent	Conservative	Asian+Female+Liberal	Hispanic+Male+Conservative
Number unique annotators	5388	3764	3687	206	54
MAE: Baseline aggregated model	0.86	0.86	1.01	0.84	0.96
MAE: Jury learning model	0.60	0.58	0.65	0.62	0.64

Table 1: Performance against individual annotator's test labels for three models: today's standard state-of-the-art aggregate approach (which is annotator-agnostic, and makes one prediction per example), a group-specific version of our proposed architecture, and the full version of our proposed architecture. The standard aggregated model's performance varies substantially between groups. For instance, it achieves an MAE of 0.83 for Asian annotators and 1.12 for Black annotators, a performance decrease of 35.0%. By comparison, we find that our model does show differences between groups, but with far smaller magnitudes. It achieves an MAE of 0.62 for Asian annotators and 0.65 for Black annotators, a performance decrease of 4.9%.

6.1.3 Is our model more performant for some groups than others? A recommender-like prediction system may implicitly group 'similar' individuals together (due to its low-rank inductive bias), leading to some unique individual and intersectional perspectives being erased. Such issues could give practitioners false confidence that they are accounting for intersectional opinions, decrease public confidence (as individuals can verify predictions are incorrect for them), and lead to decision systems that are worse than the status quo. In particular, this issue could arise for smaller groups where our model may need to borrow more information. Addressing this issue requires first understanding its extent. We therefore now ask: is performance consistent across groups of varying sizes?

This section is not an exhaustive study of intersectional identities in our dataset, which would be infeasible to report. Rather, we focus on three of the most salient group-based categories in our dataset (race, gender, and political affiliation), shown in Table 1. As illustrative examples, we also report results for two intersectional identities.

We first note that the baseline aggregated model's performance varies substantially between groups. For instance, it achieves an MAE of 0.83 for Asian annotators and a far worse 1.12 for Black annotators, a performance decrease of 35.0%. By comparison, we find that while our model does show differences between groups, but it does so with far smaller magnitudes. It achieves an MAE of 0.62 for Asian annotators and 0.65 for Black annotators.

6.2 Jury-level performance

Having shown that our architecture can model individual annotators, we now turn to jury level predictions. Ultimately, these are the most important predictions that our model makes. We ask: how performant is our model at predicting a jury's verdict?

To evaluate jury-level predictions, we'd like to compare the predicted final value produced by a jury against an observed final value produced by the same jury. Ideally, we would use comments in our test set that have been labeled by at least 12 annotators, and treat those 12 annotators as a de-facto jury.

While our dataset does not contain comments labeled by twelve annotators, it does contain a subset that were labeled by ten annotators. We rely on this small subset to get a close approximation (though likely a slightly pessimistic estimate) of the MAE of a 12-member jury. We define the *observed verdict* as the mean observed annotation over all 10 annotators, who serve as the de-facto jury. We define the *predicted verdict* as the mean of our model's individual predictions for those same ten annotators. Over 550 10-annotator juries, we find that our model produces a jury-level MAE of 0.27.

We have shown in the previous sections that jury learning is very effective when the annotators of interest are different from the distribution of annotators in the original dataset (e.g. intersectional identities). However, we show a surprising result: jury learning is more effective than the current aggregate prediction approach *even when the annotator distribution is the same as that of the dataset.* We find that the above baseline model produces an MAE of 0.41 over aggregate test labels, notably worse than our model's 0.27. These gains are due to the fact that these examples are annotated by a small group of 10 annotators where the identity of each annotator has a strong influence on the observed verdict, and jury learning can make predictions that account for the identity of these jurors.

7 USER EVALUATION

Having demonstrated the technical efficacy of our jury learning architecture in making annotator-level and jury-level inferences, we then sought to evaluate jury learning in the hands of real-world stakeholders in the content moderation setting. Our study aimed to answer the following questions:

- Q1: What jury compositions do participants select? How diverse are the selected jury compositions with respect to the implicit jury compositions embedded in the original dataset?
- Q2: Do participant-specified juries result in different prediction outcomes than those produced by a standard classifier?

To answer these questions, we targeted our study towards two audiences in the context of our focal task of toxicity classification: content moderators and platform users. Given their expertise in

making policy decisions that are tailored to the needs of particular online communities, content moderators are the population most likely to directly utilize our system.

In the supplementary materials, we replicate this study with everyday platform users who are not involved in content moderation and who might not currently feel that they have a voice in this decision-making, and we also report on survey instruments measuring the perceived legitimacy (willingness to grant deference and authority) of jury learning compared to traditional algorithms.

7.1 Study design

We conducted an online study that consisted of a Qualtrics survey with two main components: (1) a jury composition section where participants were asked to design a jury for an online community and answered several short-answer follow-up questions, and (2) a moderation algorithm legitimacy section where they answered questions to assess their perceptions of the legitimacy of a current moderation algorithm and the proposed jury algorithm. To ground the survey in a concrete scenario, we framed all of the questions in terms of a hypothetical online social media platform called YourPlatform that is planning to use algorithmic approaches as a major component of its content moderation strategy. At the start of the survey, we provided a detailed explanation of a *current algorithm* (a standard machine learning classifier trained on human labels using majority vote label aggregation) and a *jury algorithm* (an instantiation of our jury learning approach) and explained that YourPlatform was considering using one of these methods.

For the jury composition task, we displayed one of 5 possible comment sets (generated by random samples from our comment toxicity dataset [49] stratified by toxicity severity and labeler disagreement) to exemplify the type of content they would need to moderate on YourPlatform. Participants were then shown a simplified jury composition input form that allowed them to allocate 12-person jury slots using three demographic attributes: (1) *gender* (Female, Male, Non-binary, Other), (2) *race* (Black or African American, White, Asian, Hispanic, American Indian or Alaska Native, Native Hawaiian or Pacific Islander, Other) and (3) *political affiliation* (Conservative, Liberal, Independent, Other). While our approach can accommodate as many categorical values as are associated with labelers, we selected this limited set of axes because they are common demographic attributes that capture a fair amount of variation among users and that were relevant to the topics of the comment sets. Further details on our study procedure and the full survey contents are found in our supplementary materials.

7.2 Participant recruitment

For our content moderator study, we recruited participants who serve as moderators for Discord or Reddit. A member of our research team recruited Discord moderators via a server where many Discord moderators gather to discuss issues around moderation and recruited Reddit moderators of major subreddits via individual solicitation. Due to their domain expertise and relative scarcity, we offered content moderators $40.00 to complete our 30 to 45-minute survey. In total, 18 content moderators participated in our study. These participants moderate on a variety of platforms (17 on Discord, 5 on Reddit, and 2 on Twitch; some participants moderate across multiple platforms and communities). Based on self-reported demographics, we had 9 participants of age 18-24 and 9 participants of age 25-34; we had 4 women, 9 men, 4 non-binary individuals, and 1 participant who did not disclose their gender; we had 12 White, 2 Asian, and 3 multi-racial participants (1 participant did not disclose their racial identity).

7.3 Analysis approach

To analyze our results, for each available demographic attribute value, we compared its representation in participant juries against its corresponding *current algorithm implicit jury* representation. The *current algorithm implicit jury* represents the proportion of each demographic group in the original dataset. For each demographic attribute, we calculated the proportion of labelers for each comment who possessed that attribute and computed the average of these per-item proportions across the dataset. These proportions were normalized among the subset of demographic attributes that we selected for this study. The current algorithm implicit jury determined through this process—the annotators in the training data for the current algorithm—is 74% White (see red lines on Figure 6).

In addition, both survey sections had open-response questions. The goal of our analysis here was to summarize high-level themes that emerged from these responses, so a member of the research team read through all responses multiple times to generate a set of themes using qualitative open coding [16], then coded comments according to these themes.

As a post-study analysis step, we took participants' jury compositions and performed inference with our jury learning algorithm to compare the jury-based outcome with that of a standard ML algorithm.

7.4 Results: Jury composition diversity (Q1)

First, we examined the jury compositions designed by our participants. We had a total of eighteen moderators who completed our survey, of which we were able to analyze sixteen.[1] All possible values for all three attributes were utilized in the study, and participants constructed diverse juries with a mean of 5.7 unique race values (SD=0.85), 3.1 unique gender values (SD=0.56), and 3.4 unique political affiliation values (SD=0.61). This diversity involved the explicit inclusion of non-majority identities (here, defined as values other than White for race, Male or Female for gender, and Liberal or Conservative for political affiliation): on average, participants created juries with 10.31 individuals (SD=1.26) who had one or more non-majority attributes; participants created juries with on average 3.88 individuals (SD=1.76) who had two or more non-majority attributes (e.g., Black and Non-binary).

We then compared the diversity of the moderator-designed juries relative to the diversity of the current algorithm implicit jury we defined earlier. As summarized in Figure 6, we observed that for all three demographic attributes, participants juries achieved greater diversity than the current algorithm implicit jury. We performed

[1]We exclude two of the moderators' jury composition results: while these participants demonstrated an accurate understanding of the current algorithm and jury algorithm (and thus have valid moderation legitimacy responses), they utilized the "Other" fields to mean "any" or "null," but this field was defined to map to jurors who explicitly self-identified with "Other" for these attributes. Since these responses are not directly comparable, they have been excluded from the quantitative jury composition analysis.

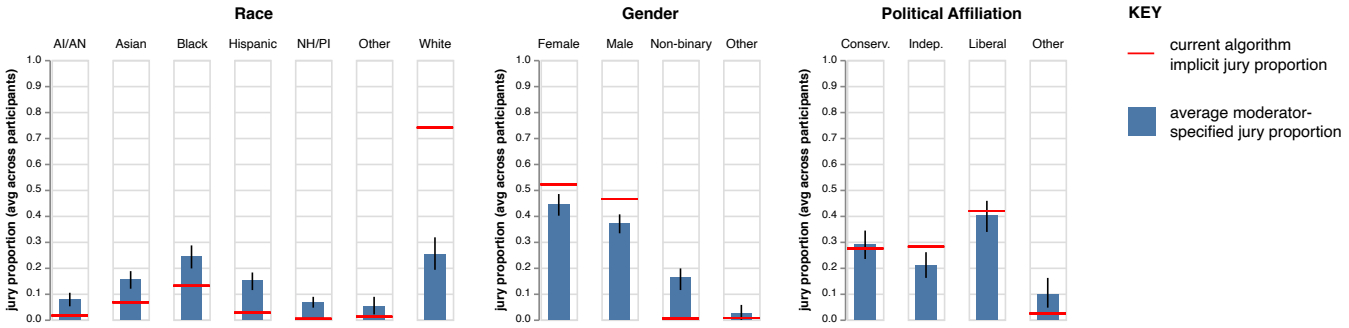

Figure 6: Jury composition results ($N = 16$). While there are sizeable disparities in group representation in the current algorithm implicit jury (denoted with red lines), the moderator-specified juries generally achieve greater diversity (raising representation for groups with the lowest red lines and lessening the gap in representation among groups).

Attribute	Value	t-statistic	p-value
Race	White**	-14.79	< 0.001
Race	Asian**	4.82	< 0.001
Race	Black or African American**	4.84	< 0.001
Race	Hispanic**	6.50	< 0.001
Race	American Indian or Alaska Native**	4.35	< 0.001
Race	Native Hawaiian or Pacific Islander**	5.60	< 0.001
Race	Other*	2.16	< 0.05
Gender	Male**	-5.18	< 0.001
Gender	Female**	-3.51	< 0.01
Gender	Other	1.13	*n.s.*
Gender	Non-binary**	7.06	< 0.001
Political affiliation	Liberal	-0.53	*n.s.*
Political affiliation	Independent*	-2.69	< 0.05
Political affiliation	Other*	2.43	< 0.05
Political affiliation	Conservative	0.59	*n.s.*

Table 2: Jury composition one-sample t-test results ($N = 16$). Values denoted with double-asterisks (**) are significant with $p < 0.01$; values denoted with a single asterisk (*) are significant with $p < 0.05$. Most notably, we observe strongly significant increases in the representation of non-White jurors, strongly significant increases in the representation of Non-binary jurors and corresponding strongly significant decreases in the representation of Male and Female jurors.

one-sample t-tests comparing the representation of demographic attribute values between the current algorithm implicit jury and the participant jury and report the results in Table 2. For racial identity, we observed strongly significantly decreases in the representation of White jurors ($p < 0.001$) and strongly significant increases ($p < 0.001$) in representation for all non-White race attribute values except for the "Other" category, where we still saw a significant increase in representation ($p < 0.05$); participants' juries contained 2.9 times the representation of non-White jurors than the current algorithm implicit jury. For gender identity, we observed a strongly significant reduction in both male and female jurors and a strongly significant increase in the representation of non-binary jurors ($p < 0.001$) with 31.5 times the representation of non-binary jurors compared to the current algorithm implicit jury. Finally, for political affiliation, we observed a significant decrease ($p < 0.05$) in the representation of Independents (who were oversampled in the original dataset) and a significant increase in the representation of other political affiliations.

Our qualitative coding shed light on the reasons underlying participants' jury composition decisions. As summarized in Table 3, a vast majority of participants aimed to prioritize diversity and equal representation of juror attributes, and the majority took special care to increase representation of groups who were targeted in the provided comment set. When asked to envision how outcomes of the jury algorithm might differ from those of the current algorithm, many participants felt that it would better capture the views of minority groups and would increase the number of comments rated as toxic. Finally, when explaining which groups had more or less voice in their jury composition, many users stated that they based their decision on whether certain groups had relevant experience with the comment topic or whether certain groups had been historically marginalized or underrepresented.

7.5 Results: Jury prediction outcomes (Q2)

7.5.1 How many classification outcomes flip between toxic and non-toxic? Having established that participants composed a diverse selection of juries, we now ask: do these participant-specified juries result in different prediction outcomes than those produced by a standard classifier? As our standard baseline classifier, we use the

Jury composition approach	Count	Anticipated outcomes	Count	Justifications for increasing/decreasing voice	Count
Diversity, equal representation, fairness	13	Better capturing views of minority groups	8	Extent to which <group> has relevant experience or knowledge about the issues at hand	7
Prioritizing groups targeted in sample comments	10	Increase in number of comments rated as toxic	6	Extent to which <group> is marginalized or has experienced historical harms	6
Increasing representation of minority groups	4			Extent to which <group> has been targeted by the sample comments	4
Decreasing representation of groups that may cause harm to minority groups	1			Extent to which <group> is expected to view as toxic the content that other groups would find toxic	3

Table 3: In a field study, we asked participants ($N = 16$) open response questions about their approach to composing juries, the outcomes they anticipated, and justifications for their jury composition decisions. We manually coded their responses to identify themes. The count column indicates the number of participants who mentioned each theme. A majority of participants aimed to prioritize diversity and equal representation of juror attributes, and the majority took special care to increase representation of groups who were targeted in the provided comment set.

same state of the art BERTweet-based classifier defined earlier in the Model Evaluation section.

We first aim to establish that jury learning effectively models the individual jurors selected by participants. We therefore perform a disaggregated analysis in which we randomly sample jurors for each of the diverse, participant-provided jury composition 100 times. We then compute an MAE over all the comments labeled by all selected jurors. We find that jury learning decreases the average error of these diverse participant-provided juror's opinions by 41% when compared to the predictions from our baseline aggregated model, from an MAE of 1.05 to 0.62.

We then focus on the final predictions produced by jury learning, computed through a median-of-means estimator over 100 resampled juries. We compare these predictions to the predictions from our baseline classifier. To determine whether a jury's prediction caused a toxicity decision to change, we binarize the final regression values found from our median-of-means estimator such that a value < 1, indicates non-toxic, and ≥ 1, corresponding to a value "slightly toxic" or greater in the annotation scheme, indicates toxic. We remove a small number of juror sheets (mean: 4%) because because the participant requested more jurors from an intersectional identity than available in the original dataset.

Over the 16 moderator-provided juries, we find that a mean of *13.6%* of decisions flip, with a standard deviation of 4.1% across moderators. This result suggests that a meaningful number of classifications can change between an off-the-shelf classifier and a jury learning classifier customized for the community.

7.5.2 Do diverse juries flip divisive comments? Having established that the diverse juries provided by participants cause toxicity predictions to flip, we now investigate *which* comments are flipping. Specifically, we ask whether the comments that flip tend to be more divisive among annotators than the comments that do not flip. To make this determination, we compute an annotator disagreement rate for each comment in the test set. We find the annotator disagreement rate by randomly sampling pairs of annotations for the same comment, and computing the percentage of the time that these pairs disagree with each other. Across all comments that participants' proposed juries cause to flip, the annotator disagreement rate is 46.4%. A two-proportion z-test shows this to be a significant increase over the 37.2% disagreement rate for comments that these juries did not flip ($z = 2.89, p < .01$). This result indicates that jury learning has the biggest impact on comments that are the most divisive.

8 DISCUSSION

In this section, we reflect on the contributions, limitations and future opportunities of our approach. We reflect on how designers and product teams might use it in practice. Finally, we reflect on the ethical considerations of our approach.

8.1 Implications for design

How do we build artificial intelligence systems that reflect our values? Values are often diverse and heterogeneous across individuals. While the raw datasets that most ML systems rely on are made up of individuals, today's approaches to building machine learning classifiers typically abstract the individuals out of the pipeline. They view differences among annotators as label noise, rather than as genuine differences of opinion that practitioners need to understand and account for. Jury learning is an attempt to re-think the machine learning pipeline so that practitioners make explicit value judgements about the voices that their classifiers should reflect. We believe, and our evaluation suggests, that practitioners and researchers who make these decisions explicitly will include greater diversity than typical models today. Our approach centers individuals at each stage of the pipeline rather than abstracting or aggregating them as in today's ML approaches.

8.1.1 A new lens for ML interpretability. Today, approaches for machine learning interpretability typically base their explanations around properties of the item in question, aiming to communicate how an item's features or content led the model to make its decision. Our approach affords a new, complementary lens to machine learning interpretability, in which we aim to explain a model's prediction as a function of the properties of its annotators.

Consider an activist whose social media posts are removed by an AI. They might rightfully wonder if their posts were moderated because the annotators that trained the moderation model had different political views. Such information is currently completely hidden, making it difficult for this activist to trust the outcomes of automated moderation systems. In contrast, jury learning enables new interpretable methods for users to interrogate which groups' opinions are being listened to, which groups' opinions are not, and for what kinds of inputs. Does it weigh men's voices more

than women's in its training data? Does this amplify bias for some topics? Jury learning could empower end users to call for greater representation. More broadly, jury learning offers a new way for users and decision makers to communicate and debate normative decisions about whose perspectives should be included.

8.2 Ethical considerations

Compared to today's implicit procedure for selecting a classifier's voice, our explicit approach introduces its own ethical issues and trade-offs.

8.2.1 Making fair and transparent decisions. How do we eradicate harmful biases in machine learning? Existing approaches in the machine learning fairness literature largely take the training data as a given, and then enforce statistical constraints that can introduce notions of fairness on the resulting model's output (e.g., that a model's decisions must be equitable across genders). In other words, the existing fairness literature starts from the assumption that the underlying statistical correlations in the world are flawed, and that they must be corrected through post-hoc adjustments of decisions that were learned from a flawed world. However, these solutions are ultimately band-aids to a problematic input pipeline. A useful distinction is to consider different forms of justice. We can think of jury learning as a form of *procedural justice*. We do not claim to guarantee the fairness of outcomes, but instead we make claims around the correctness of the process.

Our work instead takes the position that it is sometimes more desirable or tractable to select specific people whose voices should be emulated. This position comes with its own set of challenges. While jury learning empowers and normatively encourages practitioners to think carefully about whose voices their models represent, it does not inherently enforce notions of fairness. Jury learning can be used to beneficially select the most important voices to a practitioner, or to equitably represent a diverse set of groups. Jury learning can also be used to unintentionally or deliberately make biased decisions that may cause harm. A practitioner could purposely exclude a relevant group's voice, or could unintentionally include a harmful voice. If, for instance, a practitioner unintentionally or intentionally selects racist jurors, then the resulting model will be racist.

However, unlike fairness approaches that focus on outcomes, the jury learning approach can make use of tools from the human-computer interaction and social sciences literature that provide established and effective levers to recruit, train, and socialize people such that a practitioner can overcome these challenges and achieve the jury composition that they want. We argue that, if the options are to make decisions by enforcing post-hoc constraints on the decisions learned from large and somewhat random datasets, or the jury learning approach of explicitly selecting people who make decisions, it is often better to go with the latter. In doing so, we can entrust decision-making to the most relevant, qualified people for any task or situation.

Beyond the juror selection considerations above, we also advocate for transparent juries. Even if jury learning leads to increases in diversity, jury learning is unlikely to dramatically re-order the existing power structures within sociotechnical systems. Rather, the aim of jury learning is to ensure that decision-making regarding issues of power, in particular whose voices are represented in classification tasks, is made explicit and transparent. We therefore propose that any organization deploying a jury-based classifier make their jury composition transparent to relevant stakeholders. In doing so, jury learning enables a new set of conversations between practitioners and stakeholders about precisely whose voices a classifier is emulating, the implications of emulating those voices, and the ability to explore and implement different sets of voices. Such conversations could be considered akin to a *Batson challenge*, a process in the US legal system in which stakeholders to a case can argue against the removal of particular jurors on impermissible ground. To that end, we also suggest that practitioners make their instantiation of our jury learning interactive interface publicly available as a sandbox so that anyone can understand how different juries might make different decisions.

8.2.2 Addressing the ecological fallacy. Our aim with jury learning is to help practitioners recognize and integrate annotator disagreement in the classifier pipeline. To achieve this, we ask practitioners to create a jury that specifies the individuals or groups their classifiers should emulate. One approach to creating such a classifier might have been to simply model each group as a singular representative voice, akin to personas in traditional HCI methods. However, such an approach would promote an ecological fallacy because it does not demonstrate the extent to which annotators within a group disagree with each other. Our approach instead models individual annotators, enabling tools that inform practitioners about disagreement within groups. The amount of this disagreement depends upon the extent to which the group identities selected by the practitioner can explain disagreement between annotators.

Another risk arises from the requirement that many machine learning tasks produce a single decision. To make this decision, we must take a position that resolves any disagreement: specifically, we use a median-of-means approach that takes the median jury after randomly sampling 100 juries that match the practitioner's jury composition, ignoring ones that might be outliers. Thus, our system still presents an opportunity promote the ecological fallacy. To ensure that practitioners are aware of this risk, our interactive interface clearly communicates that each jury composition can have many different instantiations, and that a jury's verdict may change depending upon which jurors happened to be selected. Further, we promptly display visualizations that contextualize each individual juror within their larger group, demonstrating where they fall within the distribution of other annotators that may have been chosen in their stead. Finally, as mentioned in our system description, the interface disallows selecting any groups with an insufficient number of annotators in the dataset to complete the resampling procedure without replacement, directing practitioners to collect more data for the particular group.

8.2.3 Accurate representation. As with any machine learning system, our approach is only as good as the labels provided to it, and only as good as the model's ability to learn from these labels. If a dataset does not accurately represent the views of its annotators, or does not accurately convey each annotator's group memberships, then our model will emulate those inaccuracies. Users of our system must therefore follow best practices when collecting their datasets. For instance, the dataset we used to demonstrate jury learning relies on self-identifications, which brings its own tradeoffs when

compared to an approach that attributes identity characteristics to participants.

Further, no current model architecture can perfectly emulate the annotators it was trained on. The high stakes nature of social computing settings means that there can be substantial harm from misrepresenting minority perspectives. Good crowdsourcing practices should therefore be paired with participatory methods for auditing the models produced by jury learning, and any performance metrics should be split out by group to ensure that the model's performance is equitable across groups. Future work should also develop new techniques based on robust machine learning to ensure that models are trained to explicitly optimize for performance across all subpopulations rather than on average [41].

8.2.4 Abdication of responsibility. One risk of the jury learning approach is that it may provide a mechanism for platforms to both avoid taking broad policy stances and also evade blame for content moderation decisions. This stems from two aspects of its present design that remain open-ended: (1) the choice of the decisionmaker who wields the jury learning tool to make content moderation decisions, and (2) the meta-policy by which the jury learning outputs are incorporated into an end-to-end content moderation system (answering questions like: *what circumstances do and don't warrant the creation of a new jury? How do we weight the jury outcomes against other algorithmic tools' outcomes in a standard, principled way? How should we balance the jury outcome against the opinion of a content moderator? How do we select what comments should be sent to a jury?*). Ultimately, the organizations deploying classifiers are responsible for the decisions their classifier makes, and should still be held accountable for them.

8.2.5 Annotator privacy. Faithful and accurate representation of jurors potentially requires information collection about the private views and attributes of jurors. Factors such as sexual orientation are highly private, but can be a key part of creating a jury with diverse perspectives. Data recovery and record linkage attacks mean that such information could potentially be leaked to an adversary. Balancing the rights of jurors to privacy with the accountability and transparency benefits of leveraging juror demographics is a challenging open question. Future work in jury learning should investigate methods to disclose potential privacy harms to annotators. For instance, disclosure may require that, when collecting new datasets, we make clear to labelers the possibility that these attributes may be recoverable. Future work should also draw on approaches for differential privacy in AI [1] to help ensure us that individuals or rare demographic attributes are not rediscoverable.

8.3 Limitations and future work

As with any machine learning approach, there are several limitations and future directions worth discussing:

8.3.1 Domains. In this paper, we demonstrated jury learning using a single application domain: toxicity detection. However, our approach is designed to work for any task in which there is annotator disagreement, a dataset denoting each annotator's relevant group memberships, and an existing classification model that produces high quality embeddings for each item. In particular, we hope future work will investigate using jury learning for medical decision making and design tasks, which may rely on different perspectives. For instance: a doctor making use of a model to help them decide between different treatment options might benefit if their model's decisions were based on a jury that reflects a particular patient's preferences in quality of life trade-offs. Or an amateur designer making use of an AI-based tool for poster design might benefit from the ability to create juries reflecting different design sensibilities or artistic schools of thought.

8.3.2 Jury metaphor. Jury learning loosely draws on a metaphor of juries in the US legal system, but we do not intend this rhetorical device to indicate a complete isomorphism. Rather, jury learning draws on two specific aspects of juries: the notion of moving from a single decision maker to a group of voting decision makers, and the idea of some sort of juror selection process.

Juries in the US legal system are the sites of complex social behaviors facilitated through an intricate legal apparatus [42]. These behaviors yield benefits and challenges to justice (for instance, group polarization [79]) and are not the focus of our system. For instance, jury learning does not draw on the deliberative nature of juries, which has been the subject of decades of study in legal literature [22]. Jury learning's approach to juror selection also contrasts with the approach taken in the US legal system. Jury learning empowers practitioners and end-users to select their own jury composition. In the US legal system, jury selection is not in hands of single individual, but rather jurors are selected through a process in which stakeholders argue to determine its composition. As discussed above in our ethical considerations section, a stakeholder-centered selection process may sometimes be useful in jury learning, and existing work in the HCI literature [52, 53] demonstrates how such a process could be put into practice within our system.

8.3.3 Group identifiers. To demonstrate jury learning, we relied on an existing dataset that provided group membership information for each annotator. This dataset happened to focus on collecting this information for socio-demographic groups. One limitation to note is that the choice to use categories here has consequences. For instance, non-binary individuals find gender dropdown forms problematic unless they include appropriate nonbinary options and an open text box for description when appropriate [76]. One approach to creating inclusive interfaces in this respect is to ensure that all relevant options are represented in the jury interface. Another would be to allow the practitioner to explore the set of people who used the open-ended textbox and select a subset of them for inclusion as possible jurors.

Finally, our approach currently relies on datasets that include explicit information about the groups that each annotator belongs to. Future work should investigate unsupervised approaches to finding different voices within datasets [46], potentially rendering the jury learning approach possible with any existing dataset.

8.4 Positionality statement

The authors represent backgrounds ranging from computer science (HCI, machine learning) to media psychology. We acknowledge critical arguments making thoughtful cases for removing AI from

socio-technical systems, as well as arguments substantially increasing human control, oversight and audits of them. Our ideological commitment in this paper is to situations where improvement rather than outright removal of the AI is the appropriate mitigation strategy. We also acknowledge our shaping by the North American normative commitment to decisions being made by a jury of peers. Historically, juries have been sites of both progressive and regressive decision-making. Finally, we recognize that the term "toxic" is non-specific and often used as a catch-all term for a variety of forms of content that people do not wish to see online. In order to be consistent with the process used to collect this dataset, we draw upon this use of the term "toxic."

9 CONCLUSION

Machine learning often means learning to imitate people. So whose voices–whose labels–does a machine learning algorithm learn to imitate? Faced with endemic disagreement in user-facing tasks, we have to make a choice. But today's supervised learning pipelines typically abstract individual people out of the pipeline, treating people as abstractions or aggregated pseudo-humans. As a result, we lack the ability to reason over *who* disagrees and why. Jury learning is an attempt to bridge the realities of machine learning with the realities of contested tasks. Our approach enables practitioners to make explicit value judgements that inform how models resolve disagreement. If successful, we hope that this approach will help developers make more informed and intentional decisions about creating and deploying classifiers in these contexts.

ACKNOWLEDGMENTS

We thank Joseph Seering for his contributions to our user study. We thank Jane E, Harmanpreet Kaur, Danaë Metaxa, Ranjay Krishna, Matthew Joerke and James Landay for insightful discussions, feedback, and support. We thank Deepak Kumar for providing our toxic content dataset. We thank the reviewers for their helpful comments and suggestions. Mitchell L. Gordon was supported by the Apple Scholars in AI/ML PhD fellowship. This work was supported by the Computer History Museum, Patrick J. McGovern Foundation, and the Stanford Institute for Human-Centered Artificial Intelligence.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Oct 2016). https://doi.org/10.1145/2976749.2978318

[2] Rediet Abebe, Solon Barocas, Jon Kleinberg, Karen Levy, Manish Raghavan, and David G Robinson. 2020. Roles for computing in social change. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 252–260.

[3] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning*. PMLR, 60–69.

[4] Sohail Akhtar, Valerio Basile, and Viviana Patti. 2021. Whose Opinions Matter? Perspective-aware Models to Identify Opinions of Hate Speech Victims in Abusive Language Detection. arXiv:2106.15896 [cs.CL]

[5] Jennifer N L Allen, Antonio A Arechar, Gordon Pennycook, and David G Rand. 2020. Scaling Up Fact-Checking Using the Wisdom of Crowds. https://doi.org/10.31234/osf.io/9qdza

[6] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 291–300.

[7] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *Ai Magazine* 35, 4 (2014), 105–120.

[8] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–13.

[9] Natã M Barbosa and Monchu Chen. 2019. Rehumanized crowdsourcing: a labeling framework addressing bias and ethics in machine learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[10] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *Calif. L. Rev.* 104 (2016), 671.

[11] Paul M Barrett. 2020. Who Moderates the Social Media Giants? *Center for Business* (2020).

[12] Michael S Bernstein, Margaret Levi, David Magnus, Betsy Rajala, Debra Satz, and Charla Waeiss. 2021. ESR: Ethics and Society Review of Artificial Intelligence Research. *arXiv preprint arXiv:2106.11521* (2021).

[13] Jonathan Bragg, Mausam, and Daniel S. Weld. 2018. Sprout: Crowd-Powered Task Design for Crowdsourcing. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 165–176.

[14] Robyn Caplan and Tarleton Gillespie. 2020. Tiered governance and demonetization: The shifting terms of labor and compensation in the platform economy. *Social Media+ Society* 6, 2 (2020), 2056305120936636.

[15] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2334–2346.

[16] Kathy Charmaz. 2006. Constructing grounded theory: A practical guide through qualitative research. (2006).

[17] John Joon Young Chung, Jean Y Song, Sindhu Kutty, Sungsoo Hong, Juho Kim, and Walter S Lasecki. 2019. Efficient elicitation approaches to estimate collective crowd answers. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–25.

[18] Sasha Costanza-Chock. 2020. *Design justice: Community-led practices to build the worlds we need*. The MIT Press.

[19] Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2021. Dealing with Disagreements: Looking Beyond the Majority Vote in Subjective Annotations. arXiv:2110.05719 [cs.CL]

[20] Alexis De Toqueville. 1835. Democracy in America. *New York: A Mentor Book from New American Library* (1835).

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[22] Dennis J Devine, Laura D Clayton, Benjamin B Dunford, Rasmy Seying, and Jennifer Pryce. 2001. Jury decision making: 45 years of empirical research on deliberating groups. *Psychology, public policy, and law* 7, 3 (2001), 622.

[23] Michael A DeVito, Jeremy Birnholtz, Jeffery T Hancock, Megan French, and Sunny Liu. 2018. How people form folk theories of social media feeds and what it means for how we study self-presentation. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–12.

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

[25] Brian Dobreski. 2018. Toward a value-analytic approach to information standards. *Proceedings of the Association for Information Science and Technology* 55, 1 (2018), 114–122.

[26] Anca Dumitrache. 2015. Crowdsourcing disagreement for collecting semantic annotation. In *European Semantic Web Conference*. Springer, 701–710.

[27] Anca Dumitrache, Lora Aroyo, and Chris Welty. 2017. Crowdsourcing ground truth for medical relation extraction. *arXiv preprint arXiv:1701.02185* (2017).

[28] Anca Dumitrache, Lora Aroyo, and Chris Welty. 2018. Capturing ambiguity in crowdsourcing frame disambiguation. *arXiv preprint arXiv:1805.00270* (2018).

[29] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[30] Motahhare Eslami, Aimee Rickman, Kristen Vaccaro, Amirhossein Aleyasen, Andy Vuong, Karrie Karahalios, Kevin Hamilton, and Christian Sandvig. 2015. "I always assumed that I wasn't really that close to [her]" Reasoning about Invisible Algorithms in News Feeds. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 153–162.

[31] Jerry Alan Fails and Dan R Olsen Jr. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*. 39–45.

[32] Jenny Fan and Amy X Zhang. 2020. Digital juries: A civics-oriented approach to platform governance. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–14.

[33] Timnit Gebru. 2020. Lessons from Archives: Strategies for Collecting Sociocultural Data in Machine Learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA,

3609.
[34] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. Datasheets for datasets. *arXiv preprint arXiv:1803.09010* (2018).
[35] Spiros V Georgakopoulos, Sotiris K Tasoulis, Aristidis G Vrahatis, and Vassilis P Plagianakos. 2018. Convolutional neural networks for toxic comment classification. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*. 1–6.
[36] Tarleton Gillespie. 2018. *Custodians of the Internet: Platforms, content moderation, and the hidden decisions that shape social media*. Yale University Press.
[37] Mitchell L Gordon, Kaitlyn Zhou, Kayur Patel, Tatsunori Hashimoto, and Michael S Bernstein. 2021. The disagreement deconvolution: Bringing machine learning performance metrics in line with reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.
[38] Mary L Gray and Siddharth Suri. 2019. *Ghost work: How to stop Silicon Valley from building a new global underclass*. Eamon Dolan Books.
[39] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016), 3315–3323.
[40] Khalid Haruna, Maizatul Akmar Ismail, Suhendroyono Suhendroyono, Damiasih Damiasih, Adi Cilik Pierewan, Haruna Chiroma, and Tutut Herawan. 2017. Context-Aware Recommender System: A Review of Recent Developmental Process and Future Research Direction. *Applied Sciences* 7, 12 (2017). https://doi.org/10.3390/app7121211
[41] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. 2018. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*. PMLR, 1929–1938.
[42] Reid Hastie, Steven D Penrod, and Nancy Pennington. 2013. *Inside the jury*. Harvard University Press.
[43] Jialun Aaron Jiang, Morgan Klaus Scheuerman, Casey Fiesler, and Jed R Brubaker. 2021. Understanding international perceptions of the severity of harmful content online. *PloS one* 16, 8 (2021), e0256762.
[44] Jigsaw. 2019. Jigsaw Unintended Bias in Toxicity Classification. https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overview
[45] Jeremy Kahn. 2020. Can Facebook's new A.I. banish Pepe the Frog? https://fortune.com/2020/05/12/facebook-a-i-hate-speech-covid-19-misinformation/
[46] Sanjay Kairam and Jeffrey Heer. 2016. Parting crowds: Characterizing divergent interpretations in crowdsourced annotation tasks. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. 1637–1648.
[47] Kate Klonick. 2019. The Facebook Oversight Board: Creating an independent institution to adjudicate online free expression. *Yale LJ* 129 (2019), 2418.
[48] Yubo Kou, Xinning Gui, Shaozeng Zhang, and Bonnie Nardi. 2017. Managing Disruptive Behavior through Non-Hierarchical Governance: Crowdsourcing in League of Legends and Weibo. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article 62 (Dec. 2017), 17 pages. https://doi.org/10.1145/3134697
[49] Deepak Kumar, Patrick Gage Kelley, Sunny Consolvo, Joshua Mason, Elie Bursztein, Zakir Durumeric, Kurt Thomas, and Michael Bailey. 2021. Designing Toxic Content Classification for a Diversity of Perspectives. *arXiv preprint arXiv:2106.04511* (2021).
[50] Matthew Lease. 2011. On quality control and machine learning in crowdsourcing. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.
[51] Guillaume Lecué and Matthieu Lerasle. 2020. Robust machine learning by median-of-means: theory and practice. *The Annals of Statistics* 48, 2 (2020), 906–931.
[52] David Timothy Lee, Ashish Goel, Tanja Aitamurto, and Helene Landemore. 2014. Crowdsourcing for participatory democracies: Efficient elicitation of social choice functions. In *Second AAAI Conference on Human Computation and Crowdsourcing*.
[53] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, et al. 2019. WeBuildAI: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–35.
[54] Toby Jia-Jun Li, Lindsay Popowski, Tom Mitchell, and Brad A Myers. 2021. Screen2Vec: Semantic Embedding of GUI Screens and GUI Components. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.
[55] Arend Lijphart. 1999. *Patterns of democracy*. Yale university press.
[56] Tong Liu, Akash Venkatachalam, Pratik Sanjay Bongale, and Christopher Homan. 2019. Learning to predict population-level label distributions. In *Companion Proceedings of The 2019 World Wide Web Conference*. 1111–1120.
[57] VK Chaithanya Manam and Alexander J Quinn. 2018. Wingit: Efficient refinement of unclear task instructions. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*.
[58] Brandeis Marshall. 2021. Algorithmic misogynoir in content moderation practice. *Heinrich-Böll-Stiftung* (2021).
[59] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.
[60] Erwan Moreau, Carl Vogel, and Marguerite Barry. 2019. A paradigm for democratizing artificial intelligence research. In *Innovations in Big Data Mining and Embedded Knowledge*. Springer, 137–166.
[61] Michael Muller, Christine T. Wolf, Josh Andres, Michael Desmond, Narendra Nath Joshi, Zahra Ashktorab, Aabhas Sharma, Kristina Brimijoin, Qian Pan, Evelyn Duesterwald, and Casey Dugan. 2021. *Designing Ground Truth and the Social Life of Labels*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3411764.3445402
[62] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. arXiv:2005.10200 [cs.CL]
[63] Kayur Patel, James Fogarty, James A Landay, and Beverly L Harrison. 2008. Examining Difficulties Software Developers Encounter in the Adoption of Statistical Machine Learning.. In *AAAI*. 1563–1566.
[64] Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics* 7 (2019), 677–694.
[65] Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. 2019. Human uncertainty makes classification more robust. In *Proceedings of the IEEE International Conference on Computer Vision*. 9617–9626.
[66] Pew Research Center. 2021. *The State of Online Harassment*. Technical Report. Washington, D.C. https://www.pewresearch.org/internet/2021/01/13/the-state-of-online-harassment/
[67] Vinodkumar Prabhakaran, Aida Mostafazadeh Davani, and Mark Diaz. 2021. On Releasing Annotator-Level Labels and Information in Datasets. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 133–138. https://aclanthology.org/2021.law-1.14
[68] Iyad Rahwan. 2018. Society-in-the-loop: programming the algorithmic social contract. *Ethics and Information Technology* 20, 1 (2018), 5–14.
[69] William H Riker. 1992. The justification of bicameralism. *International Political Science Review* 13, 1 (1992), 101–116.
[70] Anna Rogers. 2021. Changing the World by Changing the Data. arXiv:2105.13947 [cs.CL]
[71] Christian Sandvig, Kevin Hamilton, Karrie Karahalios, and Cedric Langbort. 2014. Auditing algorithms: Research methods for detecting discrimination on internet platforms. *Data and discrimination: converting critical concerns into productive inquiry* 22 (2014).
[72] Mike Schaekermann, Graeme Beaton, Minahz Habib, Andrew Lim, Kate Larson, and Edith Law. 2019. Understanding expert disagreement in medical data analysis through structured adjudication. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–23.
[73] Mike Schaekermann, Graeme Beaton, Elaheh Sanoubari, Andrew Lim, Kate Larson, and Edith Law. 2020. Ambiguity-aware ai assistants for medical data analysis. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–14.
[74] Mike Schaekermann, Joslin Goh, Kate Larson, and Edith Law. 2018. Resolvable vs. irresolvable disagreement: A study on worker deliberation in crowd work. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–19.
[75] Morgan Klaus Scheuerman, Alex Hanna, and Emily Denton. 2021. Do Datasets Have Politics? Disciplinary Values in Computer Vision Dataset Development. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 317 (oct 2021), 37 pages. https://doi.org/10.1145/3476058
[76] Morgan Klaus Scheuerman, Aaron Jiang, Katta Spiel, and Jed R Brubaker. 2021. Revisiting Gendered Web Forms: An Evaluation of Gender Inputs with (Non-) Binary People. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–18.
[77] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 614–622.
[78] C Estelle Smith, Bowen Yu, Anjali Srivastava, Aaron Halfaker, Loren Terveen, and Haiyi Zhu. 2020. Keeping community in the loop: Understanding wikipedia stakeholder values for machine learning-based systems. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.
[79] Cass R Sunstein. 1999. The law of group polarization. *University of Chicago Law School, John M. Olin Law & Economics Working Paper* 91 (1999).
[80] Harini Suresh and John V Guttag. 2019. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002* (2019).
[81] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.
[82] Joseph D Tucker, Suzanne Day, Weiming Tang, and Barry Bayus. 2019. Crowdsourcing in medical research: concepts and applications. *PeerJ* 7 (2019), e6762.
[83] Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572* (2018).

[84] Alex Hai Wang. 2010. Detecting spam bots in online social networking sites: a machine learning approach. In *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 335–342.
[85] Jing Wang and Xin Geng. 2019. Classification with Label Distribution Learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (Macao, China) *(IJCAI'19)*. AAAI Press, 3712–3718.
[86] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems. In *Proceedings of the Web Conference 2021*. 1785–1797.
[87] Tian Wang, Yuri M Brovman, and Sriganesh Madhvanath. 2021. Personalized Embedding-based e-Commerce Recommendations at eBay. *arXiv preprint arXiv:2102.06156* (2021).
[88] Bowen Yu, Ye Yuan, Loren Terveen, Zhiwei Steven Wu, Jodi Forlizzi, and Haiyi Zhu. 2020. Keeping designers in the loop: Communicating inherent algorithmic trade-offs across multiple objectives. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. 1245–1257.
[89] Amy X Zhang, Grant Hugh, and Michael S Bernstein. 2020. PolicyKit: Building Governance in Online Communities. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 365–378.
[90] Biqiao Zhang, Georg Essl, and Emily Mower Provost. 2017. Predicting the distribution of emotion perception: capturing inter-rater variability. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. 51–59.
[91] Xinyi Zhou and Reza Zafarani. 2018. Fake news: A survey of research, detection methods, and opportunities. *arXiv preprint arXiv:1812.00315* (2018).
[92] Haiyi Zhu, Bowen Yu, Aaron Halfaker, and Loren Terveen. 2018. Value-sensitive algorithm design: Method, case study, and lessons. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–23.