FR-Train: A Mutual Information-Based Approach to Fair and Robust Training

**Yuji Roh** 1 **Kangwook Lee** 2 **Steven Euijong Whang** 1 **Changho Suh** 1

# Abstract

Trustworthy AI is a critical issue in machine learn- ing where, in addition to training a model that is accurate, one must consider both *fair* and *robust* training in the presence of data bias and poisoning. However, the existing model fairness techniques mistakenly view poisoned data as an additional bias to be ﬁxed, resulting in severe performance degradation. To address this problem, we pro- pose FR-Train, which *holistically performs fair and robust model training*. We provide a mutual information-based interpretation of an existing ad- versarial training-based fairness-only method, and apply this idea to architect an additional discrimi- nator that can identify poisoned data using a clean validation set and reduce its inﬂuence. In our ex- periments, FR-Train shows almost no decrease in fairness and accuracy in the presence of data poi- soning by both mitigating the bias and defending against poisoning. We also demonstrate how to construct clean validation sets using crowdsourc- ing, and release new benchmark datasets1.

# Introduction

As machine learning becomes widespread in the Software

2.0 era (Karpathy, 2017), *trustworthy AI* is becoming in- creasingly critical. In addition to simply training accurate models, there is an urgent need to address multiple require- ments including fairness, robustness, explainability, trans- parency, and accountability altogether (IBM, 2020). In particular, we focus on fairness and robustness, which are closely related issues that are affected by the same training data. For sensitive applications like healthcare, ﬁnance, and self-driving cars, a trained model must not discriminate cus-

1School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea 2Department of Electrical and Computer Engineering, University of Wisconsin- Madison, Madison, Wisconsin, USA. Correspondence to: Steven Euijong Whang *<*[swhang@kaist.ac.kr](mailto:swhang@kaist.ac.kr)*>*.

*Proceedings of the 97 th International Conference on Machine Learning*, Online, PMLR 119, 2020. Copyright 2020 by the au- thor(s).

1<https://github.com/yuji-roh/fr-train>

tomers based on sensitive attributes including age, sex, or religion. In addition, as applications often rely on external datasets for their training data, the model training must be resilient against noisy, subjective, or even adversarial data.

Traditionally, model fairness research (Venkatasubramanian, 2019; Chouldechova & Roth, 2018; Verma & Rubin, 2018) has focused on developing metrics such as disparate im- pact (Feldman et al., 2015), equalized odds (Hardt et al., 2016), and equal opportunity (Hardt et al., 2016), which cap- ture various notions of discrimination. More recently, there has been a surge in *unfairness mitigation* techniques (Bel- lamy et al., 2018b), which improve the model fairness by either ﬁxing the training data, training process, or trained model. Unfairness mitigation usually involves some tradeoff between the model’s accuracy and fairness. Most recently, generative adversarial networks (GANs) are being adapted to a fairness setting (Zhang et al., 2018a). The architecture of GANs is suitable because accuracy and fairness are not always aligned, and it makes sense to simultaneously train two models: a classiﬁer that predicts labels using input fea- tures and an adversary that predicts sensitive attributes using the classiﬁer’s predicted labels.

Robust model training is also important and needs to be concurrently taken into consideration. As dataset publishing is becoming mainstream as demonstrated by systems like Kaggle and Google Dataset Search (Noy et al., 2019), it is easy to publish data that is noisy, subjective, and even ad- versarial, which we hereafter refer to as *poisoned data*. As a result, there has been a proliferation of algorithms that make model training resilient to data poisoning as well (Natara- jan et al., 2013; Biggio et al., 2011; Fre´nay & Verleysen, 2014). However, data poisoning attacks have become in- creasingly sophisticated, and defending against all of them is difﬁcult (Koh et al., 2018).

Solving model fairness without addressing data poisoning may lead to a worse tradeoff between accuracy and fairness. For example, consider a banking system that is giving out loans where there are two sensitive groups: men and women. Suppose we use disparate impact (Feldman et al., 2015) as the fairness measure. If the model’s positive prediction rate is *M* for men and *W* for women, the disparate impact is min *M , W* where a value of 1 is considered perfectly fair. Figure 1 shows a toy example of ﬁve men and ﬁve women

*W*

*M*

*{ }*

who need loans. Each person is associated with a single- dimensional feature *x*, and only the ones with a rounded box would pay back their loans (i.e., their labels are positive). Let us train a threshold classiﬁer that divides the people into two groups where those on the left are denied loans

**Fair classifier**

(Acc, DI) = (0.8, 1)

**Clean**

**Non-fair classifier**

(Acc, DI) = (1, 0.5)



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F | M | F | F | M | F | M | M | F | M |





**X**

and those on the right are granted loans. On the clean data

**Poisoned**

**Non-fair classifier**

**Fair classifier**

above, a classiﬁer that does not consider fairness (non-fair

Acc

poi

= 0.9

Acc

poi

= 0.8

classiﬁer, red dotted line) can have perfect accuracy at the cost of having a disparate impact of 0.5 because 40% of females are granted loans while 80% of males are granted loans. On the other hand, a fair classiﬁer (blue solid line) can

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F | M | F | F | M | F | M | M | F | M |

(Accclean, DI) = (0.9, 0.67)



(Accclean, DI) = (0.6, 1)





**X**

divide the people such that the disparate impact is perfect, but the accuracy is only 0.8. Now suppose we poison the data where we ﬂip the labels of the 5*th* and 7*th* persons (both male) from positive to negative as shown below. While each classiﬁer is trained on the poisoned data, its accuracy is measured using the clean data labels. For the non-fair classiﬁer trained on this data, the results are mixed where the accuracy decreases from 1 to 0.9, but the disparate impact increases from 0.5 to 0.67. However, the fair classiﬁer has strictly worse results where the accuracy decreases from 0.8 to 0.6 without any change in the disparate impact. Hence, the fair classiﬁer’s accuracy-fairness tradeoff is worse when the data is poisoned. One proposal is to sanitize the data prior to the model training, but it is known that removing poisoning without any knowledge of the model is extremely difﬁcult (Koh et al., 2018).

Our main contribution is an integrated solution called FR-Train, which trains accurate models that are also fair and robust to poisoning. FR-Train extends a state-of- the-art fairness-only method called Adversarial Debiasing (AD) (Zhang et al., 2018a), which consists of a generator used for classiﬁcation and a discriminator that distinguishes predictions from one sensitive group against others, simi- lar to GANs (Goodfellow et al., 2014). The discriminator ensures that the prediction *y*ˆ is independent of the sensitive attribute *z*. We ﬁrst provide interpretation of such an ad- versarial learning approach using mutual information. We then use the results as an inspiration to add a new robustness discriminator that uses mutual information to distinguish (training examples, predictions) of the training data from (validation examples, validation labels) of a separate and clean validation set. This discriminator ensures that the model predictions on the training data are “consistent” with labels on clean data, where the clean validation set acts as a reference to the training. In addition, we also utilize the ro- bustness discriminator results to further improve the fairness training by re-weighting examples. In our experiments, we show that addressing robustness and fairness sequentially during model training is not as effective as addressing them concurrently as in FR-Train.

Another contribution is addressing the challenge of con- structing a clean validation set and gracefully handling the

*Figure 1.* A small dataset of 10 people who need loans (F: female, M: male). A rounded box indicates a positive label. The clean data (above) is poisoned by ﬂipping two labels (below). The vertical lines are the decision boundaries of non-fair and fair threshold classiﬁers. DI is disparate impact, and Accclean (Accpoi) is the accuracy on clean (poisoned) data.

case where it is small or unavailable. To this end, we demon- strate a practical crowdsourcing method using majority vot- ing for constructing a clean validation set, which has less poisoning than the input data. We construct clean validation sets from real datasets using Amazon Mechanical Turk and release them as a community resource. In the worst case when the validation set is non-existent, we show how the parameters of FR-Train can be adjusted to still maintain reasonable accuracy and fairness.

In the following sections, we demonstrate the weaknesses of current fairness methods, propose FR-Train with experi- ments, and present the related work.

# Vulnerability of Fairness Methods

We perform experiments to demonstrate that state-of-the-art fairness methods are indeed vulnerable even to simple poi- soning attacks. We generate a synthetic dataset as shown in Figure 2a (see the generation details in Section 4.1). There

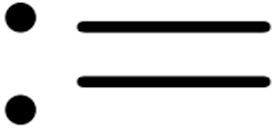
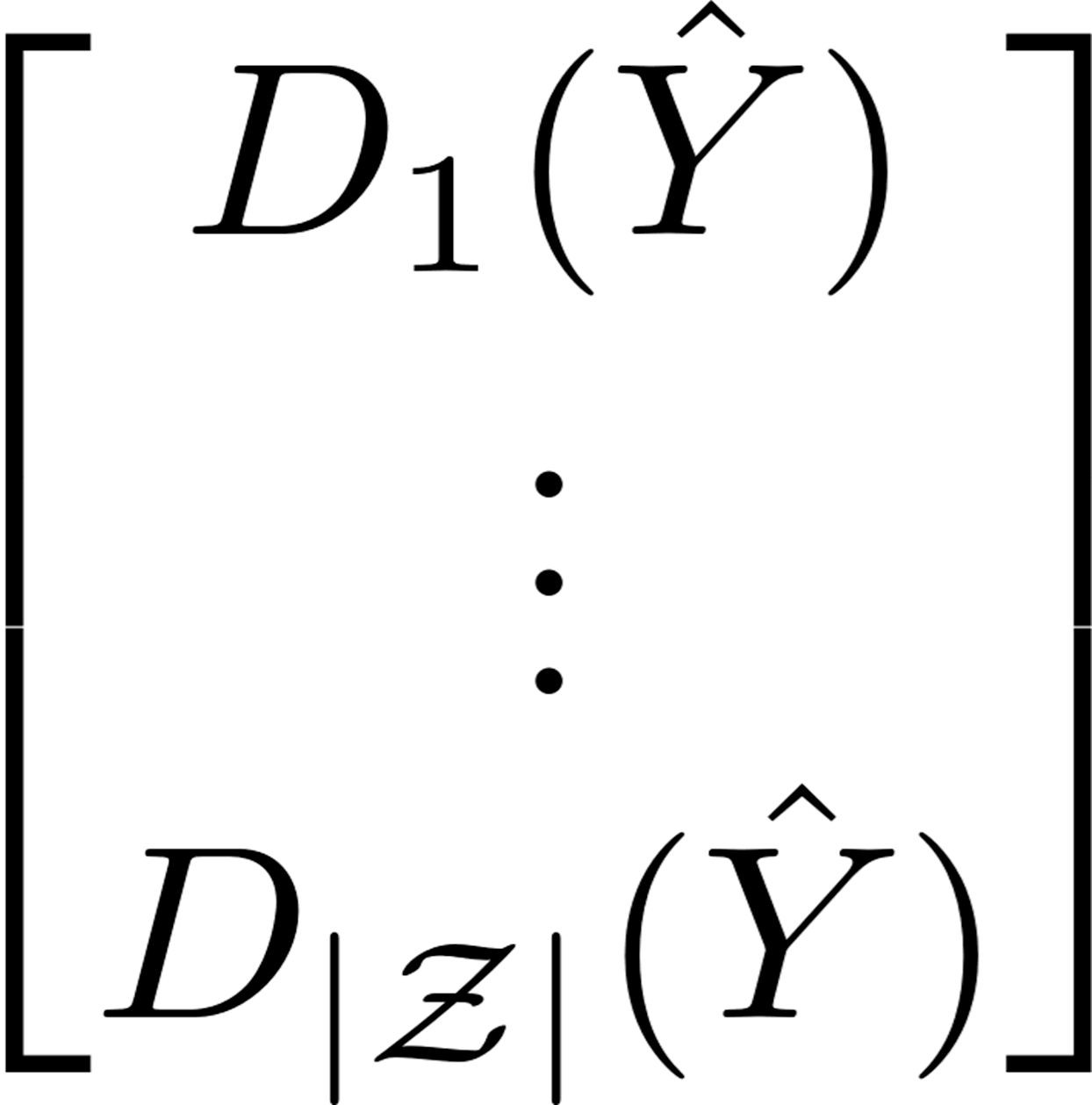
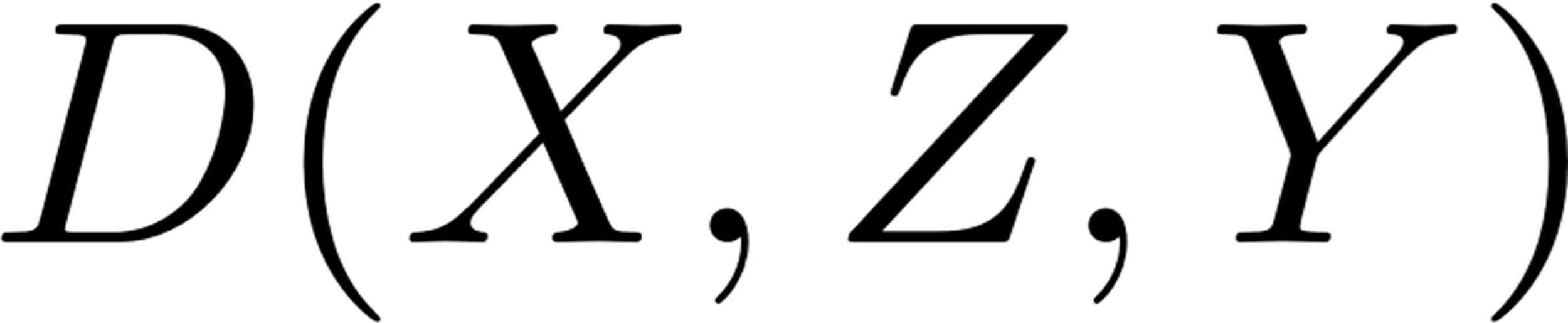
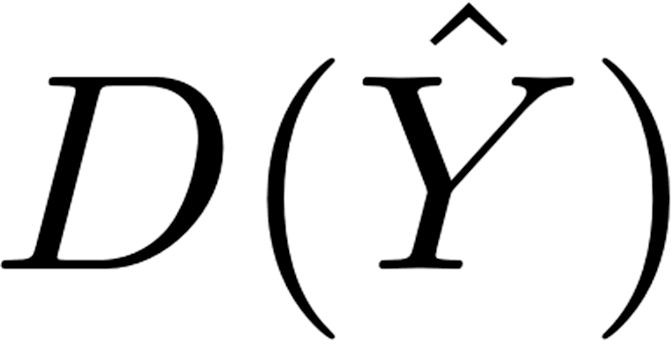
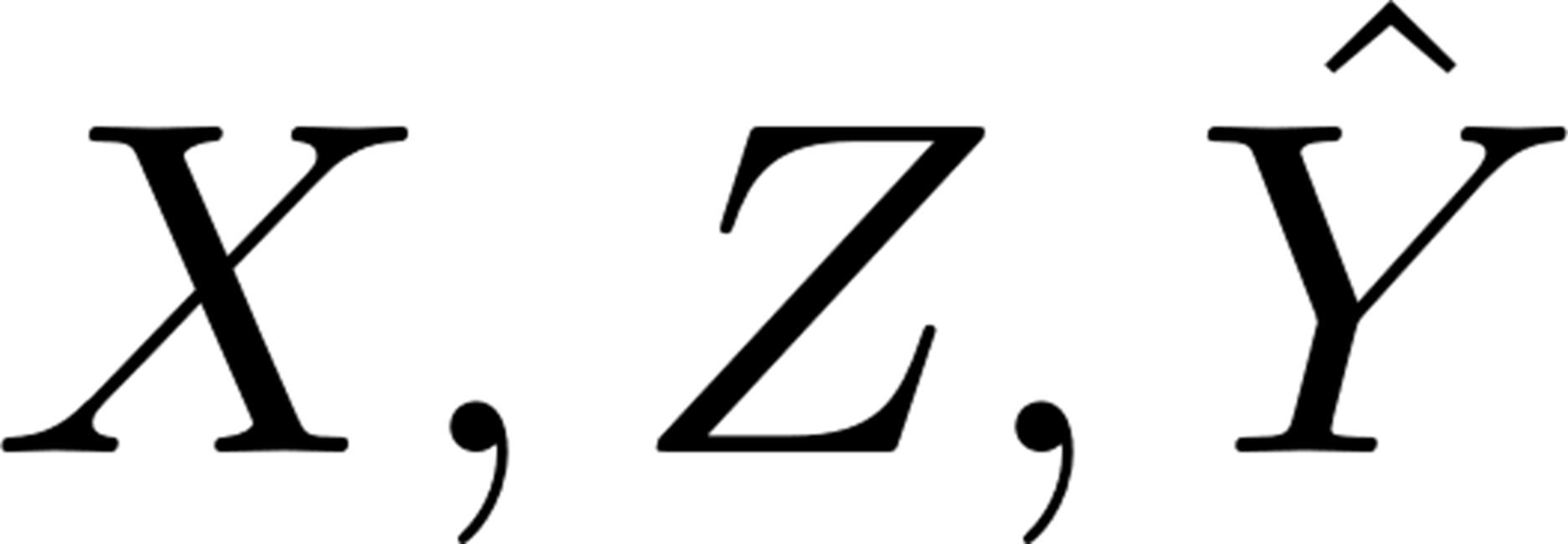
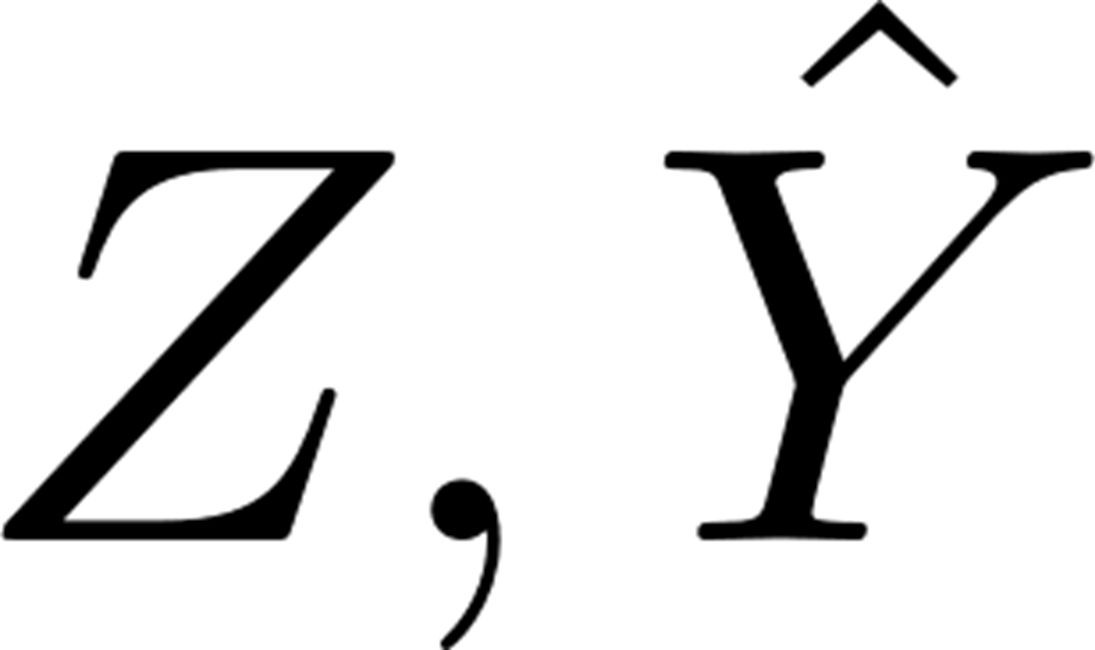
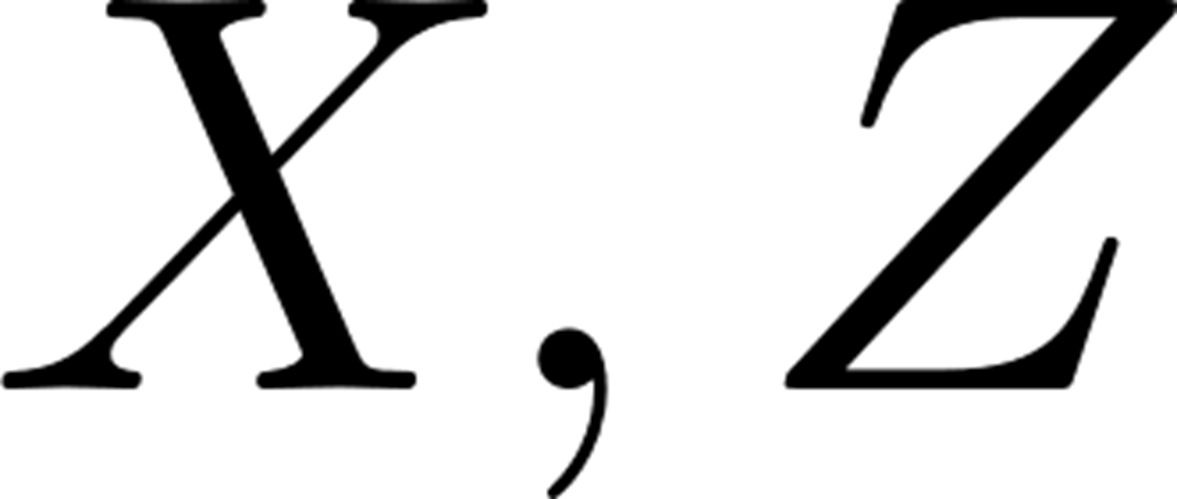
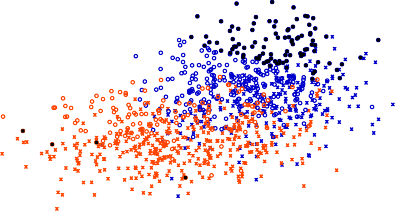
are two non-sensitive attributes *x*1 and *x*2, which are re- ﬂected in the x-axis and y-axis, respectively. The examples

are further divided into two classes based on the *sensitive* attribute *z*. For generation of poisoned data, we poison 10% of the training data by ﬂipping the labels of examples that belong to a speciﬁc *z* attribute (for this experiment *z*

= 1) so as to maximize the accuracy performance degrada- tion. This approach is similar to an existing label ﬂipping method (Paudice et al., 2018). To make a validation set, we randomly select clean examples that amount to 10% of the entire training data.

We use disparate impact as the fairness measure and evaluate a fairness method called Fairness Constraints (Zafar et al., 2017), which incorporates a regularization term that reﬂects fairness constraints in the context of convex margin-based classiﬁers such as logistic regression and support vector machines (SVMs). As this method involves a regularization

5.0



Softmax

Generator Discriminator

(Classifier) for Fairness

Discriminator

for Robustness

|  |  |
| --- | --- |
| **z = 0 (pos) z = 0 (neg) z = 1 (pos) z = 1 (neg)**  **poisoned data** |  |
|  |  |

0.0

**x2**



-5.0

0*·*8



**clean data poisoned data**

**Disparate Impact**

0*·*7

-10.0 -5.0 0.0 5.0 10.0

## x1

1. Synthetic data with label-ﬂipped poisoning



*Figure 3.* The architecture of FR-Train.

fairness training by re-weighting examples.

## Fairness

We denote by *Dtr* the training data set. Suppose *Dtr* has

*m* examples *{*(*x*(*i*), *z*(*i*), *y*(*i*))*}m*



where *x*(*i*) contains the

0*·*6

non-sensitive attributes,

*z*(*i*)

*i*=1

contains the sensitive attributes,

0*·*5



0*·*4

0*·*725 0*·*750 0*·*775 0*·*800 0*·*825 0*·*850 0*·*875 0*·*900

**Accuracy**

1. Accuracy-fairness tradeoff curves for Fairness Constraints

*Figure 2.* The top ﬁgure shows a synthetic dataset with data poison- ing. Examples are divided into *z* = 1 (marked with circles) and *z*

= 0 (crosses) as per a sensitive attribute *z*. The blue points indicate positive labels while the red points denote negative ones. For the poisoning, we ﬂipped labels of 10% of the examples with *z* = 1 so as to maximize the accuracy performance degradation (Paudice et al., 2018). The bottom ﬁgure shows that poisoning signiﬁcantly worsens the accuracy-fairness tradeoff (i.e., the curve shifts to the left) of the Fairness Constraints method (Zafar et al., 2017).

factor *λ* that balances the accuracy and fairness objectives, we can obtain a tradeoff curve by adjusting its value. Fig- ure 2b shows two accuracy-fairness tradeoff curves obtained with the clean and poisoned synthetic datasets. Notice that adding data poisoning clearly shifts the curve to the left, which means accuracy decreases. This coincides with our

and *y*(*i*) is the label. Both the sensitive attribute and label can be multi-class, i.e., they can have one of multiple values.

For notational simplicity, we assume there is one sensitive attribute, which can be viewed as a merged result of mul- tiple sensitive attributes with a larger alphabet size. For illustrative purposes, we focus on disparate impact, leaving in the supplementary our formulation and experimental re- sults for equalized odds and equal opportunity. Disparate impact aims for the same positive prediction ratio for each sensitive attribute *z* where is the set of possible sensitive attribute values. We use the following deﬁnition for disparate impact:

*e z z*

**Deﬁnition 1.** *(Disparate Impact)*

*P* (*Y*ˆ = 1*|Z* = *z*1) = *P* (*Y*ˆ = 1*|Z* = *z*2)*, Vz*1*, z*2 *e z.*

The ﬁrst discriminator in FR-Train distinguishes predictions

w.r.t. one sensitive group from those in the others. Disparate impact intends the sensitive attribute to be independent of

the model’s prediction, i.e., *I*(*Z*; *Y*ˆ) = 0.

We explain how FR-Train can enforce the above constraint. Let *PZ*(*z*) be the distribution of *Z* where *z e z*. Let

intuition. The poisoning confuses the model so that there

are more biased examples to ﬁx, which in turn makes it

*overreact* and thus sacriﬁce more on accuracy. We also

*Y*ˆ*|Z* = *z ~ PY*ˆ *|z* (*.*) and *Y*ˆ

*z∈Z PZ*(*z*)*PY*ˆ *|z* (*.*).

*~ PY*ˆ (*.*). Then *PY*ˆ (*.*) =

leave in the supplementary the accuracy-fairness tradeoff curves of Fairness Constraints on real datasets. The results clearly show that both accuracy and fairness decrease on

The following theorem asserts that mutual information is equivalent to the following function optimization where

the optimal discriminator *D\**(*y*ˆ) = *PZ|Y*ˆ (*z|y*ˆ) and

*z*

*z*

the poisoned data. In Section 4, we will show how data

poisoning affects other fairness methods.

*z∈Z*

*D\**(*y*ˆ) = 1*, Vy*ˆ *e y*.

# FR-Train

**Theorem 1.** *I*(*Z*; *Y*ˆ) =

max *PZ*(*z*) *P*

*Dz* (*y*ˆ):

*Dz*(*y*ˆ)=1*, ∀y*ˆ *z∈Z*

*Y z*

*z*

log *Dz*(*Y*ˆ) + *H*(*Z*)*.*

We now describe FR-Train (see Figure 3). Unlike traditional GANs, the generator is a classiﬁer that receives an example *x X* and returns a prediction *y*ˆ. There are two discrim- inators that respectively optimize fairness and robustness using mutual information. In addition, the outputs of the robustness discriminator can be used to further improve the

*e*

While deferring the detailed proof to the supplemental ma- terials, we provide a brief overview of the proof. As the optimization problem in the RHS is convex, we ﬁnd the optimal discriminator by solving the KKT conditions. We then show that the maximum value attained by the optimal discriminator is equal to the mutual information by using

the properties of mutual information and the generalized Jensen-Shannon divergence (Lin, 1991).

What is more involved than showing the above equality is designing the right optimization problem. One needs to carefully handcraft a plausible optimization problem so that its unique solution matches the desired quantity. Here, we design the optimization problem via a ‘guess-&-check’ approach aided by the structural insights across the KL di- vergences that appear in an alternative expression of mutual information.

We now discuss how to implement the above expression.

Section 4.1). In Section 3.3, we also use the robustness discriminator to further improve the fairness training using example re-weighting.

We ﬁrst deﬁne *X* = *V X* + (1 *V* )*X*val*, Z* = *V Z* + (1

*V* )*Z*val, and *Y* = *V Y*ˆ + (1 *V* )*Y*val. Here, note that *V*

*-*

*- -*

is an indicator random variable that denotes whether an example is generated (*V* = 1) or comes from the validation set (*V* = 0). We then want to ensure that the distribution

of (*X, Z, Y*ˆ) matches that of (*X*val*, Z*val*, Y*val). This can be

done by enforcing *I*(*V* ; *X, Z, Y* ) = 0, i.e., the predictions on the training data are indistinguishable from the labels

Since we do not know *PY*ˆ *z*

*|*

following empirical version:

(*.*) exactly, we compute the

of the validation set. Thus we can mimic the clean dataset

while expecting an indirect sanitization effect.

Analogous to the fairness discriminator, we show

max

*Dz* (*y*ˆ):

*PZ*(*z*)

1 log *Dz*(*y*ˆ(*i*)) + *H*(*Z*)*.*

that mutual information is equivalent to the fol-

*v*

lowing function optimization where the optimal dis-

Now for sufﬁciently large *m*, the number *mz*

*z Dz* (*y*ˆ)=1*, ∀y*ˆ *z∈Z*

*mz*

*i*:*z i* =*z*

of examples

criminator *D\**(*x, z, y*) = *P*

*V |X,Z,Y*

(*v|x, z, y*) and

with *z*(*i*) = *z* is approximately the same as *PZ*(*z*)*m*. There-

*v*

*v∈V*

*D\**(*x, z, y*) = 1*, V*(*x, z, y*) *e x × z × y*. The

fore, the above expression becomes:

proof is similar to that of Theorem 1.

**Theorem 2.** *I*(*V* ; *X, Z, Y* ) *=*

max

1 log *D* (*y*ˆ(*i*)) + *H*(*Z*)*.*

max

*Dz* (*y*ˆ): *z Dz* (*y*ˆ)=1*, ∀y*ˆ *z∈Z i*:*z* *i* =*z m*

*z*

*PV* (*v*) *P*

*Dv* (*x,z,y*): *v Dv* (*x,z,y*)=1*, ∀*(*x,z,y*)

*v∈V*

*x,z,Y v*

log *Dv*(*X, Z, Y* )

+ *H*(*V* )*.*

the original GAN (Goodfellow et al., 2014) when = 2. We also remark that our formulation does not require a prior knowledge on *PZ*(*z*).

*|z|*

Interestingly, this formulation is exactly the same as that in

We note that Adversarial Debiasing (AD) (Zhang et al., 2018a) has an additional projection term that is used to force the classiﬁer to never decrease the discriminator’s loss. However, we do not use this term in FR-Train because it worsens the training stability in our experiments.

## Robustness

The robustness discriminator ensures robust training by us- ing mutual information to distinguish examples and predic-

## Architecture

Σ

Σ

We describe the FR-Train architecture in Figure 3. For the loss function of the generator, we employ cross entropy:

1 *m*

*- - - -*

*L*1 = *y*(*i*) log *y*ˆ(*i*) (1 *y*(*i*)) log(1 *y*ˆ(*i*))*. m*

*i*=1

We set the loss function w.r.t. the fairness discriminator as:

*L* = max 1 log *D* (*y*ˆ(*i*)) + *H*(*Z*)

2

*D*(*·*)

*m*

*z*

*z∈Z i*:*z i* =*z*

where *D*(*.*) := (*D*1(*.*)*, . . . , D|Z|*(*.*)). The condition

*z*

tions from a clean validation set. For now, let us assume

*z∈Z*

*D\**(*Y*ˆ) = 1 can be enforced by adding a softmax

such a validation set exists (in Section 4.2, we demonstrate how to construct one). The discriminator then distinguishes the training data with predictions *{*(*x*(*i*)*, z*(*i*)*, y*ˆ(*i*))*}m* from

*i*=1

layer to the discriminator.

Finally, implementing *I*(*V* ; *X, Z, Y* ), we set the loss func- tion w.r.t. the robustness discriminator as:

(*i*)

(*i*)

(*i*)

*m*val

the validation set *{*(*x*val *, z*val *, y*val )*}i*=1 . Intuitively, if the

classiﬁer is confused by data poisoning in the training data, then its predictions will not be consistent with the labels of

3

*Dr* (*·*)

*m*

*val*

*val*

*val*

*L* = max

1 log *Dr*(*x*(*i*) *, z*(*i*) *, y*(*i*) )+

*i*:*v i* =0

the clean data, and the discriminator would be able to detect that difference. Our use of a validation set is inspired by meta learning-based robust training algorithms (Ren et al., 2018), which also defends against poisoning attacks by us- ing the validation data loss as a meta objective. However, a key difference is that we take an adversarial learning ap- proach, which introduces a knob that controls the emphasis of robust training. We ﬁnd that this knob enables FR-Train to be more robust to the validation set size (see details in

1 log(1 *Dr*(*x*(*i*)*, z*(*i*)*, y*ˆ(*i*))) + *H*(*V* )*. m*

*i*:*v i* =1

*-*

The ﬁnal objective function is the weighted sum of these value functions:

min *L*1 + *λ*1*L*2 + *λ*2*L*3*.*

*G*(*·*)

Here *λ*1 and *λ*2 are tuning knobs that play roles to emphasize fair and robust training, respectively.

## Example Re-weighting for Fairness Training In addi-

tion to the above architecture, we also utilize the decision values *Dr*(*X, Z, Y*ˆ) of the robustness discriminator as ex- ample weights to further improve the fairness training (in

Figure 3, the arrow from the robustness discriminator’s out- put to the classiﬁer’s input). In particular, the two losses *L*1 and *L*2 are now computed using the example weights. The intuition is that, by giving more weight to the clean examples, we can improve the accuracy-fairness tradeoff.

A question is when to apply these weights. If we apply the weights too early, then *D*(*X, Z, Y*ˆ) may not be accurate enough and actually harm the fairness training. Intuitively,

*Table 1.* Accuracy and fairness performances on the synthetic test datasets w.r.t. disparate impact (DI). Two types of methods are com- pared: (1) fairness methods: FC (Zafar et al., 2017), LBC (Jiang & Nachum, 2020), and AD (Zhang et al., 2018a) where “RML+” denotes the application of sanitization using RML (Ren et al., 2018) beforehand; (2) non-fairness methods: LR and RML. For FR-Train and RML, the validation set is 10% of *tr* . The amount of poisoning is 10% of *tr* . For each result of the poisoned data, we make a comparison with the clean data result and show the percentage increase or decrease.

Method Clean data Poisoned data

DI Acc. DI Acc.

we would like to use the discriminator’s results when we

know it is performing at least as well as the classiﬁer. Hence, for a more reliable signal, we use the *relative performance* between the classiﬁer and robustness discriminator to gen- erate the weights. Given the classiﬁer’s loss *Lc* and the robustness discriminator’s loss *Ld*, we compute the ﬁnal ex-

ample weights as *W* = *R* + *D*(*X, Z, Y*ˆ) (1 *R*) where

*× -*

*R* = *σ*( *Lc C*) is a conversion of the loss ratio into a prob- ability using the sigmoid function *σ* and hyperparameter *C*.

*Ld*

*-*

We note that *C* acts as a threshold on the loss ratio.

# Experiments

We provide experimental results for FR-Train. For the fairness measure, we use disparate impact, while leaving in the supplementary the results for equalized odds and equal opportunity. We evaluate all models on separate

clean test sets. In our experiments, we use two sensitive attributes *z*1 and *z*2, and disparate impact is measured as the ratio min*{ P* (*Y*ˆ=1*|Z*=*z* ) *, P* (*Y*ˆ=1*|Z*=*z* ) *}*. We use Py-

*P* (*Y*ˆ =1*|Z*=*z* ) *P* (*Y*ˆ =1*|Z*=*z* )

Torch (Paszke et al., 2017), and all experiments are per- formed on a server with Intel i7-6850 CPUs. More imple- mentation details are in the supplementary.

## Synthetic Data Results

For the synthetic data, we generate 2,000 examples with two non-sensitive attributes *x*1 and *x*2, a sensitive attribute *z*, and a label *y*, using a method similar to the algorithm proposed by (Zafar et al., 2017). Both *z* and *y* are bi- nary, and the (*x*1, *x*2) pair consists of two normal distri-

butions: (*x*1*, x*2) *y* = 0 ([ 2; 2]*,* [10*,* 1; 1*,* 3]) and

*| ~ N - -*

(*x*1*, x*2) *y* = 1 ([2; 2]*,* [5*,* 1; 1*,* 5]). The *z* attribute has the Bernoulli distribution *p*(*z* = 1) = *p*((*xj*1*, xj*2) *y* = 1)*/*[*p*((*xj*1*, xj*2) *y* = 0) + *p*((*xj*1*, xj*2) *y* = 1)] where (*xj*1*, xj*2) = (*x*1 cos(*π/*4) *x*2 sin(*π/*4)*, x*1 sin(*π/*4) + *x*2 cos(*π/*4)). Finally for each example, the *x*1 and *x*2

*-*

*| |*

*|*

*| ~ N*

values are sampled as per the normal distribution associated with the *y*. For data poisoning, we ﬂip the labels of exam- ples with *z* = 1 so as to maximize the accuracy performance degradation as described in Section 2, and the amount of poisoning is 10% of *Dtr*. In the supplementary, we also

FC .822 .806 .831 (1.1% *t*) .760 (5.7% *l*)

LBC .819 .760 .827 (1.0% *t*) .715 (5.9% *l*)

AD .807 .811 .834 (3.4% *t*) .769 (5.2% *l*)

RML+FC .822 .806 .802 (2.4% *l*) .529 (34.% *l*)

RML+LBC .819 .760 .810 (1.1% *l*) .752 (1.1% *l*)

RML+AD .807 .811 .808 (0.1% *t*) .756 (6.8% *l*)

LR .409 .885 .446 (9.1% *t*) .819 (7.5% *l*)

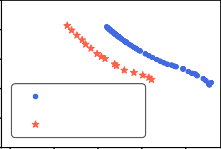
RML .471 .876 .395 (16.% *l*) .869 (0.8% *l*)

## FR-Train .818 .807 .827 (1.1% *t*) .814 (0.9% *t*)

perform FR-Train varying the amount of poisoning from 10% to 40%.

**Accuracy and Fairness** We compare FR-Train with vari- ous baselines. First, there are the fairness methods: Fairness Constraints (Zafar et al., 2017) (FC), Label Bias Correc- tion (Jiang & Nachum, 2020) (LBC), and Adversarial De- biasing (Zhang et al., 2018a) (AD). As described in the previous sections, FC adds a penalty term that captures the prediction differences across sensitive groups, while AD utilizes adversarial learning to achieve high fairness. LBC is an example re-weighting algorithm, which assumes the ex- istence of true *unbiased yet unknown* labels. LBC provides theoretical guarantees that training on the resulting loss corresponds to training on the true unbiased labels, which yields a fair model. While there exist other re-weighting techniques including (Agarwal et al., 2018), we choose LBC because it performs the best in experiments (Jiang & Nachum, 2020).

Since FR-Train is to our knowledge the ﬁrst method to address both fairness and robustness in model training, there is no fairness method that also performs data sanitization using a clean validation set. However, (Ren et al., 2018) is a state-of-the-art robust training method based on meta learning using a clean validation set, which we call RML. For a fair comparison, we thus extend the three fairness methods by ﬁrst performing RML and then utilizing the example weights in the fairness training in a straightforward fashion. In addition, we compare with non-fairness methods: logistic regression (LR) and RML.



* + 1. LBC



* + 1. AD

1*·*0

0*·*8

0*·*6

0*·*4

0*·*2

0*·*0

0*·*75 0*·*80 0*·*85

**Acc**

* + 1. FR-Train

1*·*0

0*·*8



**clean poisoned**

0*·*6

0*·*4

0*·*2

0*·*0

0*·*75 0*·*80 0*·*85



**clean poisoned**

**Acc**

* + 1. FR-Train

1*.*0

0*.*8

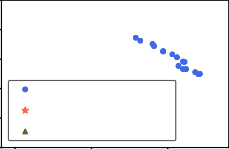
0*.*6

0*.*4

0*.*2

0*.*0

0*.*75 0*.*80 0*.*85



**clean**

**poisoned (***λ***2 = 0.4) poisoned (***λ***2 = 0.1)**

**Acc**

* + 1. FR-Train

(Val. set size = 10%)

(Val. set size = 5%)

(Val. set size = 0.1%)

*Figure 4.* Accuracy-fairness tradeoff curves. Figures (a) and (b) show that the poisoning worsens the accuracy-fairness tradeoffs of LBC (Jiang & Nachum, 2020) and AD (Zhang et al., 2018a). Figures (c) and (d) show that FR-Train maintains the tradeoffs even with a 5% validation set. When the validation set is too small (Figure (e)), FR-Train can adjust *λ*2 to reduce the adverse effect on training.

Table 1 compares FR-Train with the baselines. We use a validation set that amounts to 10% of *tr*. We also apply proper hyperparameters so that the disparate impacts are similar (around 0*.*8) across all methods, if possible. When

*D*

setting *λ*1 and *λ*2 for FR-Train, we usually ﬁx *λ*2 to some

value and then adjust *λ*1 using one-round cross validation. There is no hyperparameter tuning for logistic regression

and the meta learning-based robust training algorithm, as they have no knobs for adjusting fairness. The results show that for the fairness methods, data poisoning aggravates accuracy-fairness trade-offs. For example, the accuracy of FC falls by 5.7%, while the disparate impact of it remains a similar value. On the other hand, the performance for FR-Train does not degrade: disparate impact and accuracy increase by 1.1% and 0.9%, respectively. Table 1 also shows that combining the fairness methods with RML (rows 4–6) does not always yield better accuracy and fairness. In fact, using sanitization may lower the accuracy or fairness (e.g., RML+FC has an accuracy of 0.529 on poisoned data while FC has 0.760). The results suggest that removing poisoning and then bias is not that effective.

We observe how accuracy trades off with fairness on clean and poisoned datasets. The results for FC are shown in Figure 2b. For LBC, we employ the number of training as a knob to trade accuracy off fairness since LBC gradually improves fairness by repeatedly updating example weights per training. As shown in Figure 4a, the tradeoff curve shifts to the left, which demonstrates a clear tradeoff degradation.

For AD, we employ the *α* parameter (Zhang et al., 2018a) analogous to *λ*1 as a knob to trade accuracy off fairness. Figure 4b shows the tradeoff curve again shifts to the left.

**Validation Set Size** Figures 4c to 4e show how the valida- tion set size affects the robustness of FR-Train. In particular, we compare the accuracy-fairness tradeoff of FR-Train on clean data and that on poisoned data while varying the size

of the validation set. When running on poisoned data, we ﬁxed *λ*2 = 0*.*4 and varied *λ*1. We see that even a 5% vali- dation set (Figure 4d) is sufﬁcient to maintain the accuracy and fairness obtained on the clean data. When using 0.1% (Figure 4e), the validation set is too small and has an adverse

effect on the training. However, by decreasing the tuning knob *λ*2 down to 0.1, we can de-emphasize robust train- ing, thereby avoiding the adverse effect (Figure 4e, green triangles). This is in contrast to RML, which suffers from a non-negligible performance degradation for a very small

validation set. See details in the supplementary.

## Real Data Results

We use two real datasets: ProPublica COMPAS (Angwin et al., 2016) and AdultCensus (Kohavi, 1996), which have 7,214 and 45,222 examples, respectively. We use the same preprocessing as in IBM’s AI Fairness 360 (Bellamy et al., 2018a) and use the sensitive attribute SEX for both datasets. For data poisoning, we use the same method employed on synthetic data: ﬂipping the labels with *z* = 1 so as to maximize the accuracy performance degradation. The amount of poisoning is 10% of *Dtr*.

While we assumed that a small yet clean validation set is available in the previous synthetic data experiments, such an assumption does not hold in practice. Thus, for real- data experiments, we consider a scenario where one ﬁrst constructs a small (which amounts to 5% of *tr*) validation set based on crowdsourcing, and then uses it for FR-Train. We provide details on how to construct this validation set in Section 4.5.

*D*

Summarized in Tables 2 and 3 are the fairness and accu- racy performances of various training algorithms on the COMPAS and AdultCensus datasets, respectively. As in Table 1, we apply proper hyperparameters so that the dis- parate impacts are similar across all distinct methods, both for the clean and poisoned datasets. The results are similar to Table 1: the three fairness methods have worse disparate impact and accuracy due to data poisoning; LR and RML exhibit poor disparate impacts; and FR-Train again shows little degradation both in fairness and accuracy. Tables 2 and 3 also show that combining the fairness methods with sani- tization using RML (rows 4–6) does not always yield better accuracy and fairness and may even lower them, which is consistent with the results on synthetic data. One may won- der if the fairness baselines would perform better if they are

*Table 2.* Accuracy and fairness performances on COMPAS test data w.r.t. disparate impact (DI) where the training data is poisoned using the label ﬂipping attack. Two types of methods are compared:

*Table 4.* Confusion matrix on poisoned AdultCensus dataset w.r.t. disparate impact. Other settings are identical to Table 2.

Method Female Male

(1) fairness methods: FC, LBC, and AD where “RML+” denotes

|  |  |  |
| --- | --- | --- |
|  | *y*ˆ = 0 *y*ˆ = 1 | *y*ˆ = 0 *y*ˆ = 1 |

the application of sanitization using RML beforehand; (2) non-

fairness methods: LR and RML. For FR-Train and RML, the validation set is 5% of *tr* . The amount of poisoning is 10% of *tr* . For each result of the poisoned data, we compare with the

clean data result and show the percentage increase or decrease.

Method Clean data Poisoned data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | DI | Acc. | DI | Acc. |  |
| FC LBC | .777  .866 | .682  .671 | .794 (2.2% *t*) | .612 (10.% *l*) |  |

.838 (2.8% *l*) .671 (0.0% -)

.813 (6.1% *l*) .570 (16.% *l*)

AD .846 .680

RML+LBC .866 .671 .560 (28.% *l*) .645 (5.4% *l*)

RML+FC .777 .682

RML+AD .846 .680 .869 (0.4% *t*) .646 (3.7% *l*)

.820 (3.1% *l*) .573 (16.% *l*)

RML+FC

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RML+AD | *y* = 0  *y* = 1 | 2,345  289 | 723  96 | 3,966  1,792 | 947  473 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FR-Train | *y* = 0  *y* = 1 | 2,761  113 | 307  272 | 4,730  1,428 | 183  837 |

*Table 5.* Ablation study for FR-Train on COMPAS test data w.r.t. disparate impact (DI) where the training data is poisoned using the label ﬂipping attack. Four methods are compared: (1) FR- Train without R (*λ*2 = 0), (2) FR-Train without F (*λ*l = 0), (3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *y* = 0 | 2,990 | 78 | 4,842 | 71 |
| *y* = 1 | 238 | 147 | 1,952 | 313 |

LR .465 .674 .454 (5.0% *l*) .631 (6.4% *l*)

RML .493 .680 .575 (17.% *t*) .646 (5.0% *l*)

## FR-Train .838 .676 .846 (1.0% *t*) .670 (0.9% *l*)

trained on the clean validation set. In the supplementary, we show that the performances are actually worse than those in Tables 2 and 3. This is because the clean validation set is too small to be used as a stand-alone train data. Indeed, a similar observation is made in (Zhang et al., 2018b).

*Table 3.* Accuracy and fairness results on AdultCensus test data w.r.t. disparate impact (DI). Other settings are identical to Table 2.

Method Clean data Poisoned data DI Acc. DI Acc.

FC .825 .826 .741 (10.% *l*) .801 (3.0% *l*)

LBC .825 .825 .760 (7.9% *l*) .792 (4.0% *l*)

|  |  |  |
| --- | --- | --- |
| AD | .850 | .767 .755 (11.% *l*) .563 (27.% *l*) |
| RML+FC | .825 | .826 .821 (0.5% *l*) .780 (5.6% *l*) |
| RML+LBC RML+AD  LR | .825  .850  .328 | .825 .762 (7.6% *l*) .788 (4.5% *l*)  .767 .834 (1.9% *l*) .647 (16.% *l*) |
| RML | .327 | .847 .189 (42.% *l*) .819 (3.3% *l*)  .846 .268 (18.% *l*) .840 (0.7% *l*) |

## FR-Train .828 .824 .847 (2.3% *t*) .809 (1.8% *l*)

Table 4 shows the confusion matrix comparison for disparate impact between FR-Train, AD, and FC with sanitization us- ing RML, using the poisoned AdultCensus dataset. The re- sults are reported when FR-Train, AD, and FC achieve (Acc, DI) = (0.809, 0.847), (0.647, 0.834), and (0.780, 0.821),

respectively. FR-Train outperforms AD and FC in all as- pects because its robustness discriminator is more effective in sanitizing poisoned data.

FR-Train without example re-weighting (Without RW), and (4) FR-Train. For rows 2–4, the validation set is 5% of *tr* .

Method Clean data Poisoned data

DI Acc. DI Acc.

Without R .846 .678 .802 (5.2% *l*) .580 (14.% *l*)

Without F .482 .681 .420 (13.% *l*) .632 (7.2% *l*)

Without RW .832 .677 .840 (1.0% *t*) .624 (7.8% *l*)

## FR-Train .838 .676 .846 (1.0% *t*) .670 (0.9% *l*)

* 1. **Ablation Study**

In Table 5, we perform an ablation study to investigate the effect of each component of FR-Train. **(Without ‘R’)** When *λ*2 = 0 (i.e., no robust training), disparate impact is high, but accuracy is low on the poisoned data, just like the other fairness-only methods (Table 2, rows 1–3). **(Without ‘F’)** On the other hand, when *λ*1 = 0 (i.e., no fair training), the accuracy is high, but the disparate impact is low, just like the other non-fairness methods (Table 2, rows 7–8).

**(Without Re-weighting)** Finally, when not using example re-weighting, both accuracy and disparate impact are similar to or worse than FR-Train.

In summary, *only a holistic* framework like FR-Train *can achieve both* excellent model fairness and training robust- ness. In comparison, other methods tailored for only one of these objectives lose either accuracy, fairness or both.

## Error range of FR-Train

We investigate the error range of FR-Train. All the FR-Train experiments on the poisoned data are re-conducted with ten different random seeds to generate error ranges with mean

(*m*) and standard deviation (*s*) values. The performances are reported in the form of *m s/*2 in Table 6. On the syn- thetic and AdultCensus datasets, the lowest performances

*±*

(i.e., *m s/*2) of FR-Train are still better than the second- best performances in Tables 1 and 3, respectively. For the COMPAS dataset, the lowest performance of FR-Train is slightly worse than those of the LBC-related algorithms,

*-*

*Table 7.* Accuracy comparison of the crowdsourced labels (*N* : number of answers averaged per example) and predictions of a logistic regression model trained on ground truth labels.

Dataset Crowdsourcing Trained Model

which can be explained by the fact that the LBC algorithms

were not affected much by the poisoning in the ﬁrst place.

*Table 6.* Error range of FR-Train on the poisoned datasets w.r.t. disparate impact (DI). The poisoned settings are identical to the previous experiments.

Dataset Poisoned data

DI Acc.

Synthetic 0*.*795 *±* 0*.*019 0*.*805 *±* 0*.*008

*N* = 1 *N* = 5 *N* = 11

COMPAS 0.609 0.656 0.667 0.659

AdultCensus 0.645 0.721 0.743 0.804

*Table 8.* Accuracy and fairness of FR-Train when using crowd- sourced labels versus ground truth labels for the validation set. The training data is poisoned as in Tables 2 and 3.

Dataset Validation set DI Acc.

Crowdsourcing 0.846 0.670

COMPAS 0*.*827 *±* 0*.*027 0*.*653 *±* 0*.*005

COMPAS

Ground truth 0.899 0.674

AdultCensus 0*.*871 *±* 0*.*034 0*.*796 *±* 0*.*006

AdultCensus Crowdsourcing 0.847 0.809

## Constructing a Clean Validation Set

We now demonstrate how to construct a clean validation set using crowdsourcing. We construct validation sets for the COMPAS and AdultCensus datasets using Amazon Me- chanical Turk (AMT). Although these datasets have labels, we assume that they are not available to use as clean data. We also release the datasets as a community resource (see the supplementary for the description and data) and believe our construction can be generalized to other datasets. While crowdsourcing is not the only way to construct a clean vali- dation set, it is sufﬁcient for our purposes.

We design the AMT task for each dataset by asking a worker to classify each example. For the AdultCensus dataset, a worker looks at various attributes of a person and predicts if a person has an income of at least $50K. Instead of a yes/no answer, the answer must be on a scale of 1 to 4, which re- ﬂects the worker’s opinion more accurately. The COMPAS dataset has a similar setting where the only difference is that the workers need to predict if a criminal will reoffend in two years. Each task displays about 30 questions where we pay 3 cents per answer. For quality control, each task also contains quizzes to educate the workers, and some questions are used to evaluate the performance of the workers. Af- ter collecting answers, we ﬁlter out poor performers, take the average of at most a ﬁxed number of *N* responses per question, and compare with the threshold 2.5 to produce the ﬁnal labels. The number of answers per question can be fewer than *N* if inaccurate workers are ﬁltered out. We used workers of all demographics in the US, Canada, and UK. While this majority voting approach already works well in our experiments, one could additionally apply various qual- ity control techniques like peer-reviewing that are known to further reduce bias (Karger et al., 2011).

The important questions are how accurate the crowdsourced labels are and whether the constructed validation set results

Ground truth 0.864 0.809

in high accuracy and fairness for FR-Train. Table 7 shows the crowdsourced labels accuracies when *N* increases from 1 to 11. Even for the highest accuracies, the predictions are not perfect because the workers are looking at limited infor- mation (i.e., only the features) without any other context. To see if the workers can do better, we also train logistic regres- sion models on ground truth labels and show their accuracies on test data as upperbounds. As a result, the accuracies are comparable when *N* = 11 for both the COMPAS and Adult- Census datasets. We thus use this setting for all experiments. Table 8 shows how useful our constructed validation set is compared to using a “perfect” validation set of the same size made of ground truth labels. For both datasets, using a ground truth validation set results in slightly higher, but comparable disparate impacts while obtaining near-identical accuracies, justifying the use of crowdsourced validation sets for FR-Train.

# Related Work

**Model Fairness** The notion of discrimination has many deﬁnitions and usually comes from certain social goals that one wants to guarantee. As a result, many fairness mea- sures have been proposed (Verma & Rubin, 2018). While we focus on group fairness, which ensures similar statistics between two sensitive groups, an interesting future work is to consider individual fairness (Dwork et al., 2012), which guarantees similar prediction results across nearby examples. Recently, there has also been a surge of research on unfair- ness mitigation techniques (Bellamy et al., 2018b). Depend- ing on where a ﬁx occurs, there are mainly three approaches:

(1) *pre*-processing techniques (Kamiran & Calders, 2011; du Pin Calmon et al., 2017; Zemel et al., 2013; Feldman et al., 2015) that ﬁx the training data; (2) *in*-processing tech- niques (Zafar et al., 2017; Jiang & Nachum, 2020; Zhang

et al., 2018a; Kamishima et al., 2012; Cotter et al., 2019; 2018; Agarwal et al., 2018) that address the issue during model training; and (3) *post*-processing techniques (Hardt et al., 2016; Pleiss et al., 2017; Kamiran et al., 2012; Chzhen et al., 2019) that manipulate predictions while maintaining the model. Among the three, the in-processing techniques have the advantages that one can work with any data and that there is more control on model training (Venkatasubra- manian, 2019).

Although not our immediate focus, there are other note- worthy directions in fairness research. Causality-based fair- ness (Kilbertus et al., 2017; Kusner et al., 2017; Zhang & Bareinboim, 2018; Nabi & Shpitser, 2018; Khademi et al., 2019; Khademi & Honavar, 2020) suggests how to under- stand the causal relationship between attributes to over- come the limitations of non-causal approaches. Just as non-causal fairness can be captured by mutual informa- tion, we suspect there may be a connection between causal fairness and directed information. Another important ap- proach (Hashimoto et al., 2018) is based on distributionally robust optimization (DRO) (Sinha et al., 2017), which fo- cuses on when the sensitive attribute *z* is unknown. The DRO-based fairness approach ensures fair results by equal- izing risks over all distributions without the knowledge of *z*, but it does not directly minimize the fairness metrics such as disparate impact and equalized odds. In comparison, FR- Train assumes full knowledge of *z* and utilizes it to directly minimize the fairness metrics.

As we demonstrate in Section 2, the existing fairness tech- niques are not tailored for robust training, so they are vul- nerable to data poisoning attacks. In comparison, FR-Train addresses both model fairness and robust training within the same model training process because they are closely related and affected by the same training data.

**Robust Training** There is a heavy literature on how to make the model training robust against noisy or even ad- versarial data (Natarajan et al., 2013; Biggio et al., 2011; Fre´nay & Verleysen, 2014; Kurakin et al., 2017). A major challenge is that there can be a wide range of data poisoning attacks that keep on evolving. While sanitizing the training data before model training is an option, defending against all possible attacks seems fundamentally infeasible as demon- strated by (Koh et al., 2018). A more recent trend is to develop general defense algorithms for any attack *during model training* using meta learning (Veit et al., 2017; Li et al., 2017; Xiao et al., 2015; Hendrycks et al., 2018). Our FR-Train framework is inspired by robustness training with meta learning (Ren et al., 2018), but employs a GAN-based model to support fair and robust training without using meta learning. In particular, the design of FR-Train’s robustness discriminator is based on mutual-information-based theo- retical insights (Section 3.2). Another line of research is defending against adversarial attacks during *test* time (Big-

gio et al., 2013; Goodfellow et al., 2015; Wong & Kolter, 2018). In comparison, our focus is on defending against data poisoning on the *training* data.

# Conclusion

We proposed FR-Train, which is a holistic framework for trustworthy AI by performing both unfairness mitigation and robust training. Our key contribution is providing inter- pretation of an adversarial learning approach using mutual information and proposing a novel GAN architecture that en- joys the *synergistic effect* of combining two approaches: (1) employing a fairness discriminator that distinguishes predic- tions w.r.t. one sensitive group from others and (2) employ- ing a robustness discriminator that distinguishes training data with predictions from a clean validation set and is also used to further improve the fairness training through exam- ple re-weighting. In addition, we demonstrated how a clean validation set can be constructed using crowdsourcing and released two new datasets built from Amazon Mechanical Turk as a community resource. In our experiments, we showed that existing fairness methods are vulnerable to data poisoning, even when combined with data sanitization. In comparison, FR-Train is robust to the poisoning and can be adjusted to maintain reasonable accuracy and fairness even if the validation set is too small or unavailable.

# Acknowledgements

Yuji Roh and Steven E. Whang were supported by a Google AI Focused Research Award and by the Engineering Re- search Center Program through the National Research Foun- dation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921). This material is based upon work supported by the Air Force Ofﬁce of Scientiﬁc Research under award number FA2386-19-1-4050.

# References

Agarwal, A., Beygelzimer, A., Dud´ık, M., Langford, J., and Wallach, H. M. A reductions approach to fair classiﬁca- tion. In *ICML*, pp. 60–69, 2018.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Machine bias: There’s software used across the country to predict future criminals. And its biased against blacks., 2016.

Bellamy, R. K. E., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K. N., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varsh- ney, K. R., and Zhang, Y. AI Fairness 360: An exten- sible toolkit for detecting, understanding, and mitigat- ing unwanted algorithmic bias, October 2018a. URL <https://arxiv.org/abs/1810.01943>.

Bellamy, R. K. E., Dey, K., Hind, M., Hoffman, S. C.,

Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K. N., Richards,

J. T., Saha, D., Sattigeri, P., Singh, M., Varshney, K. R., and Zhang, Y. AI fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *CoRR*, abs/1810.01943, 2018b.

Biggio, B., Nelson, B., and Laskov, P. Support vector ma- chines under adversarial label noise. In *ACML*, pp. 97– 112, 2011.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *ECML PKDD*, pp. 387–402, 2013.

Chouldechova, A. and Roth, A. The frontiers of fairness in machine learning. *CoRR*, abs/1810.08810, 2018.

Chzhen, E., Denis, C., Hebiri, M., Oneto, L., and Pontil,

M. Leveraging labeled and unlabeled data for consistent fair binary classiﬁcation. In *NeurIPS*, pp. 12760–12770. 2019.

Cotter, A., Gupta, M. R., Jiang, H., Srebro, N., Sridharan, K., Wang, S., Woodworth, B. E., and You, S. Training well- generalizing classiﬁers for fairness metrics and other data- dependent constraints. *CoRR*, abs/1807.00028, 2018.

Cotter, A., Jiang, H., and Sridharan, K. Two-player games for efﬁcient non-convex constrained optimization. In *ALT*, pp. 300–332, 2019.

du Pin Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy,

K. N., and Varshney, K. R. Optimized pre-processing for discrimination prevention. In *NeurIPS*, pp. 3995–4004, 2017.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel,

R. Fairness through awareness. In *ITCS*, pp. 214–226, 2012. ISBN 978-1-4503-1115-1.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *KDD*, pp. 259–268, 2015.

Fre´nay, B. and Verleysen, M. Classiﬁcation in the pres- ence of label noise: A survey. *IEEE Trans. Neural Netw. Learning Syst.*, 25(5):845–869, 2014.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio,

Y. Generative adversarial nets. In *NeurIPS*, pp. 2672– 2680, 2014.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *NeurIPS*, pp. 3315–3323, 2016.

Hashimoto, T., Srivastava, M., Namkoong, H., and Liang,

P. Fairness without demographics in repeated loss mini- mization. In *ICML*, pp. 1929–1938, 2018.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using trusted data to train deep networks on labels cor- rupted by severe noise. In *NeurIPS*, pp. 10477–10486, 2018.

IBM. Trusting ai. [https://www.research.](https://www.research.ibm.com/artificial-intelligence/trusted-ai/) [ibm.com/artificial-intelligence/](https://www.research.ibm.com/artificial-intelligence/trusted-ai/) [trusted-ai/](https://www.research.ibm.com/artificial-intelligence/trusted-ai/), 2020.

Jiang, H. and Nachum, O. Identifying and correcting label bias in machine learning. In *AISTATS*, pp. 702–712, 2020.

Kamiran, F. and Calders, T. Data preprocessing techniques for classiﬁcation without discrimination. *Knowl. Inf. Syst.*, 33(1):1–33, 2011.

Kamiran, F., Karim, A., and Zhang, X. Decision theory for discrimination-aware classiﬁcation. In *ICDM*, pp. 924–929, 2012.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. Fairness-aware classiﬁer with prejudice remover regu- larizer. In *ECML PKDD*, pp. 35–50, 2012.

Karger, D. R., Oh, S., and Shah, D. Iterative learning for reliable crowdsourcing systems. In *NIPS*, pp. 1953–1961. 2011.

Karpathy, A. Software 2.0. [https://medium.com/](https://medium.com/%40karpathy/software-2-0-a64152b37c35) [@karpathy/software-2-0-a64152b37c35](https://medium.com/%40karpathy/software-2-0-a64152b37c35), 2017.

Khademi, A. and Honavar, V. G. Algorithmic bias in recidi- vism prediction: A causal perspective (student abstract). In *AAAI*, pp. 13839–13840, 2020.

Khademi, A., Lee, S., Foley, D., and Honavar, V. Fairness in algorithmic decision making: An excursion through the lens of causality. In *WWW*, pp. 2907–2914, 2019.

Kilbertus, N., Rojas-Carulla, M., Parascandolo, G., Hardt, M., Janzing, D., and Scho¨lkopf, B. Avoiding discrimina- tion through causal reasoning. In *NeurIPS*, pp. 656–666, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koh, P. W., Steinhardt, J., and Liang, P. Stronger data poisoning attacks break data sanitization defenses. *CoRR*, abs/1811.00741, 2018.

Kohavi, R. Scaling up the accuracy of naive-bayes clas- siﬁers: A decision-tree hybrid. In *KDD*, pp. 202–207, 1996.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. In *ICLR*, 2017.

Kusner, M. J., Loftus, J., Russell, C., and Silva, R. Counter- factual fairness. In *NeurIPS*, pp. 4066–4076. 2017.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L. Learn- ing from noisy labels with distillation. In *ICCV*, pp. 1928– 1936, 2017.

Lin, J. Divergence measures based on the Shannon entropy.

*IEEE Transactions on Information theory*, 37(1):145–151,

1991.

Nabi, R. and Shpitser, I. Fair inference on outcomes. *AAAI*, pp. 1931–1940, 2018.

Natarajan, N., Dhillon, I. S., Ravikumar, P., and Tewari, A. Learning with noisy labels. In *NeurIPS*, pp. 1196–1204, 2013.

Noy, N., Burgess, M., and Brickley, D. Google dataset search: Building a search engine for datasets in an open web ecosystem. In *28th Web Conference (WebConf 2019)*, 2019.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer,

A. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Paudice, A., Mun˜oz-Gonza´lez, L., and Lupu, E. C. Label sanitization against label ﬂipping poisoning attacks. In *ECML PKDD*, pp. 5–15, 2018.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J. M., and Weinberger, K. Q. On fairness and calibration. In *NeurIPS*, pp. 5684–5693, 2017.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *ICML*, pp. 4331–4340, 2018.

Sinha, A., Namkoong, H., and Duchi, J. C. Certifying some distributional robustness with principled adversarial training. In *ICLR*, 2017.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and Belongie, S. J. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pp. 6575–6583, 2017.

Venkatasubramanian, S. Algorithmic fairness: Measures, methods and representations. In *PODS*, pp. 481, 2019.

Verma, S. and Rubin, J. Fairness deﬁnitions explained. In

*FairWare@ICSE*, pp. 1–7, 2018.

Wong, E. and Kolter, J. Z. Provable defenses against adver- sarial examples via the convex outer adversarial polytope. In *ICML*, pp. 5283–5292, 2018.

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. Learn- ing from massive noisy labeled data for image classiﬁca- tion. In *CVPR*, pp. 2691–2699, 2015.

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gum- madi, K. P. Fairness constraints: Mechanisms for fair classiﬁcation. In *AISTATS*, pp. 962–970, 2017.

Zemel, R. S., Wu, Y., Swersky, K., Pitassi, T., and Dwork,

C. Learning fair representations. In *ICML*, pp. 325–333, 2013.

Zhang, B. H., Lemoine, B., and Mitchell, M. Mitigating unwanted biases with adversarial learning. In *AIES*, pp. 335–340, 2018a.

Zhang, J. and Bareinboim, E. Fairness in decision-making - the causal explanation formula. In *AAAI*, 2018.

Zhang, X., Zhu, X., and Wright, S. J. Training set debugging using trusted items. In *AAAI*, 2018b.