International Journal of Natural Sciences and Engineering Innovations
Vol. 1, No. 1 (2025)

Enhancing FR-Train in Data Imbalance

Tengyu Song
Department of Statistics
Columbia University
ts3464@columbia.edu

Abstract

This report presents a comprehensive analysis of the FR-Train model, an architecture designed to concurrently enhance fairness and robustness in artificial intelligence (AI) systems. Our study offers a detailed investigation into the discriminator modules of FR-Train, complementing the original authors' ablation study. We re-examined the key theorems using information theory principles, reproduced the model to verify its claims, and conducted additional experiments. Our main contributions are threefold: i) we provided additional information theory insights on the function of the fair discriminator; ii) we introduced an alternative approach by replacing the model's fair discriminator with a naive mutual information estimate, yielding comparable performance with a faster and more stable training process; and iii) we evaluated FR-Train's performance in extreme data imbalance scenarios and proposed a "normalized mutual information" loss function that can significantly alleviate the performance deterioration.

1 Introduction

Artificial Intelligence (AI) has been widely adopted to automate the decision making processes, largely due to its superior predictive power. However, many of its applications involve datasets containing sensitive human attributes, such as hiring, financial risk assessment, and facial recognition [1]. The pervasive role of AI in these fields has brought to the front a critical ethical question. How can we ensure that AI systems are fair and not biased? Additionally, in today's world, where large amounts of data are readily accessible for everyone to modify, the robustness of AI systems is another key concern. In our context, robustness refers to the ability for an algorithm to handle malicious or erroneous inputs that could impair its performance. Historically, research in fairness and robustness has been conducted in isolation. However, observations have been made that the pursuits towards fairness may compromise robustness [2]. Therefore, it is essential to explore integrated approaches that simultaneously address fairness and robustness in AI development.

In an effort to address this challenge, Roh, Lee, Whang and Suh proposed FR-Train [3], a fair and robust architecture in the spirit of generative adversarial networks (GANs) [4]. It consists of a prediction generator, a fair discriminator and a robust discriminator, which competes with each other during training. The authors also provided theoretical interpretation for this approach, utilizing the connection between mutual information and cross entropy.

In this report we take a deeper dive into the design of FR-Train. Specifically, we will focus on the discriminator modules which were not fully investigated in the original paper. Our work is structured in three distinct parts. Firstly, we re-derive the key theorems using the principles of information theory. Secondly, we reproduce the model and verify the claim that the discriminator modules are capable of effectively representing mutual information. Finally, we carried out additional experiments

Code is available at https://github.com/st3nv/RE-fr-train.

by replacing the fair discriminator with direct calculation of mutual information and evaluated the performance of FR-Train under extreme cases.

2 Preliminaries

2.1 Fairness

The definition of fairness varies a lot across different scenarios[5]. In the original paper, it is defined as the systematic independence between the prediction and sensitive predictor, i.e. the chances of having a positive prediction are equal across different sensitive groups. A good measure of this fairness performance is disparate impact. Here we provide its formal definition.

Definition 1. *(Disparate Impact (DI))*

Let $Z \in \{0, 1\}$ be the sensitive feature and let $\hat{Y}$ be the class prediction of a model. We can define

$$\mathrm{DI} = \frac{\min\left\{\mathbb{P}(\hat{Y} = 1 \mid Z = 1), \mathbb{P}(\hat{Y} = 1 \mid Z = 0)\right\}}{\max\left\{\mathbb{P}(\hat{Y} = 1 \mid Z = 1), \mathbb{P}(\hat{Y} = 1 \mid Z = 0)\right\}}. \tag{1}$$

Obviously DI will a value between 0 and 1, where a value of 1 signifies the highest level of fairness. Suppose we have a model to predict whether or not to give out loans. If probability of getting a loan for a white person is 0.8, and for a black person is 0.4, then the DI is 0.5.

2.2 Information theory

Here we review some important ideas of information theory that are useful in this report. In this section, all the definitions and theorems are directly sourced from *Elements of Information Theory* by Thomas M. Cover and Joy A. Thomas [6].

Definition 2. *(Entropy)*

Let $X \in \mathcal{X}$ be a discrete random variable, the entropy of X is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x). \tag{2}$$

Entropy is a measure of the uncertainty level of a random variable. In our experiments, we use the natural e as the base of logarithm. So the entropy is in the unit of *nat*.

Definition 3. *(Conditional Entropy) Let X, Y be two discrete random variables and $(X, Y) \sim p(x, y)$, then the conditional entropy is defined as*

$$H(Y \mid X) = \sum_{x \in \mathcal{X}} p(x) H(Y \mid X = x). \tag{3}$$

$H(Y \mid X)$ represent the extra information gain from Y given that we have known X. Now we are ready to give a formal definition of mutual information.

Definition 4. *(Mutual Information (MI)) In the same settings as above, the mutual information between X and Y is defined as*

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \tag{4}$$

Based on the formula, mutual information can be interpreted as the KL-divergence between the joint distribution $p(x, y)$ and the product of the marginal distribution $p(x)p(y)$. We can also calculate mutual information using entropy and conditional information, which is more convenient in many cases.

Theorem 1.

$$I(X; Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \tag{5}$$

Similar to conditional information, we can also define conditional mutual information as follows.

Definition 5. *(Conditional Mutual Information)*

The conditional mutual information of random variables X and Y given Z is defined by

$$\begin{aligned} I(X;Y \mid Z) &= H(X \mid Z) - H(X \mid Y, Z) \\ &= \mathbb{E}_{p(x,y,z)} \log \frac{p(x, y \mid Z = z)}{p(x \mid Z = z)p(y \mid Z = z)}. \end{aligned} \tag{6}$$

Another key theorem to calculate mutual information with multiple variables is the chain rule. We will use this theorem later in our experiment.

Theorem 2. *(Chain Rule for Mutual Information)*

$$I\left(X_1, X_2, \ldots, X_n; Y\right) = \sum_{i=1}^{n} I\left(X_i; Y \mid X_{i-1}, X_{i-2}, \ldots, X_1\right). \tag{7}$$

2.3 FR-Train

Here we provide a short overview of FR-Train. As proposed by the authors, it is a unified architecture designed to simultaneously enhance fairness and robustness for classification tasks, based on mutual information.

Overall Architecture

FR-Train is comprised of three primary components: a generator, a fair discriminator, and a robust discriminator, all of which are implemented as neural networks. During the training process, the generator aims to predict labels as accurately as possible while also confuse the discriminators. On the other hand, the fair discriminator uses the output from the generator to predict the sensitive feature. Intuitively, if the fair discriminator performs well, it suggests that our generator is not good at fairness. Hence, there exists a competitive dynamic between these two components to achieve a balance where the final model is both fair and accurate in its predictions. Meanwhile, the robust discriminator employs a clean validation set to identify and mitigate the impact of poisoned samples during training, achieved by both re-weighting the samples and loss feedback to the generator.

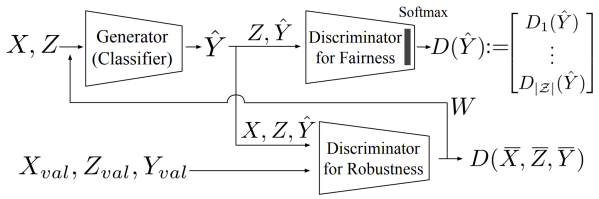


Figure 1: Structure of FR-Train from the original paper [3].

Details of robust discriminator

The robustness discriminator plays a crucial role in FR-Train. Its mechanism is slightly different from the fair discriminator. We show its flow chart in Figure 2. It ensures robust training by distinguishing the training features and predictions from a clean validation set. If the robust discriminator predicts that a sample is from the training set, then it's more likely to be poisoned since it's different from the clean data. As a result, the model will down-weight those suspicious samples during training. Additionally, the loss of the robust discriminator is also feedback to the loss of the generator, so that the generator can learn how to deal with the "outliers" by itself. The authors showed that the loss of the robust discriminator serves as a proxy for $I(V; \bar{X}, \bar{Z}, \bar{Y})$, where V is an indicator for whether or not it is from the training set and $\bar{X} = VX + (1 - V)X_{\text{val}}$, $\bar{Z} = VZ + (1- V)Z_{\text{val}}$, $\bar{Y} = V\hat{Y} + (1 - V)Y_{\text{val}}$.

Loss functions

Based on the architecture in Figure 1, here we provide the loss functions of all three component that are used in practice.

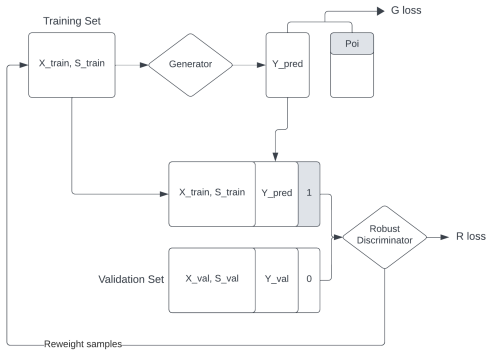


Figure 2: Flow chart of robustness discriminator of FR-Train.

(i) Fair Discriminator

$$\mathcal{L}_F = \sum_{z \in \mathcal{Z}} \sum_{i:z^{(i)}=z} \frac{1}{m} \log D_z^f \left(\hat{y}^{(i)} \right). \tag{8}$$

Notice that it's the cross entropy between output logits of fair discriminator D^f and sensitive feature Z.

(ii) Robust Discriminator

$$\mathcal{L}_R = \max_{D^r(\cdot)} \sum_{i:v^{(i)}=0} \frac{1}{m} \log D^r \left(x_{\text{val}}^{(i)}, z_{\text{val}}^{(i)}, y_{val}^{(i)} \right) + \sum_{i:v^{(i)}=1} \frac{1}{m} \log \left(1 - D^r \left(x^{(i)}, z^{(i)}, \hat{y}^{(i)} \right) \right). \tag{9}$$

Similarly it's the cross entropy between output logits of robust discriminator D^r and the binary indicators of whether samples are in training set (0) or clean validation set (1).

(iii) Generator

$$\mathcal{L}_G = (1 - \lambda_f - \lambda_r)\text{CE}_G - \lambda_f \mathcal{L}_F - \lambda_r \mathcal{L}_R. \tag{10}$$

where CE_G is the cross entropy of label prediction, defined as

$$\text{CE}_G = \frac{1}{m} \sum_{i=1}^{m} -y^{(i)} \log \hat{y}^{(i)} - \left(1 - y^{(i)} \right) \log \left(1 - \hat{y}^{(i)} \right) \tag{11}$$

and λ_f and λ_r are tuning knobs for tradeoff between accuracy, fairness and robustness

From the definition of loss functions we can find the minimax game between the discriminators and generator, similar to the structure of GANs[4].

3 Relationship between discriminator and mutual information

The core idea of FR-Train is to achieve fairness by minimizing the mutual information between the predicted logits and sensitive labels. In the training process, the generator is actually maximizing the loss of the fair discriminator. This design relies on the following key theorem to build connection between cross entropy loss and mutual information.

Theorem 3. *Suppose $Z \in \mathcal{Z}$ is a discrete random variable and $\hat{Y} \in \mathcal{Y}$ is a random variable that can be continuous or discrete. The mutual information can be shown to be equivalent to the maximum of an optimization problem.*

$$I(Z; \hat{Y}) = \max_{D_z(y): \sum_z D_z(\hat{y})=1, \forall \hat{y}} \sum_{z \in \mathcal{Z}} \mathbb{P}_Z(z) \mathbb{E}_{\hat{Y}|z} \left[\log D_z(\hat{Y}) \right] + H(Z). \tag{12}$$

Here $D_z(\hat{Y})$ can be seen as the probability of $P(Z = z \mid \hat{Y})$ to be "optimized" since the sum over z is $\sum_z D_z(\hat{y}) = 1$.

In the paper, the authors proved this theorem using KKT conditions. Here we present an alternative solution using information theory. The intuition is that the term we are maximizing resembles what we have seen in the minimum expected code length problem on a change of variable. And the constraint $\sum_z D_z(\hat{y}) = 1$ has close connection with the relaxed Kraft inequality.

Proof. We only prove the case $\hat{Y}$ is continuous since it is essentially the same for the discrete case. Let $\hat{Y} \sim f(\hat{y})$. From Theorem 1 we know

$$I(Z, Y) = H(Z) - H(Z \mid \hat{Y}). \tag{13}$$

Compare it with (12), it turns out we need to prove

$$H(Z \mid \hat{Y}) = - \max_{D_z(y):\sum_z D_z(\hat{y})=1,\forall \hat{y}} \sum_{z\in\mathcal{Z}} \mathbb{P}_Z(z)\mathbb{E}_{\hat{Y}|z}\left[\log D_z(\hat{Y})\right]. \tag{14}$$

Let $L(z;\hat{y}) = \log \frac{1}{D_z(\hat{y})}$, we have

$$\begin{aligned}
&- \max_{D_z(y):\sum_z D_z(\hat{y})=1,\forall \hat{y}} \sum_{z\in\mathcal{Z}} \mathbb{P}_Z(z)\mathbb{E}_{\hat{Y}|z}\left[\log D_z(\hat{Y})\right] \\
=& \min_{\sum_z D_z(\hat{y})=1,\forall \hat{y}} \sum_{z\in\mathcal{Z}} \mathbb{P}_Z(z)\mathbb{E}_{\hat{Y}|z}\left[\log \frac{1}{D_z(\hat{Y})}\right] \\
=& \min_{\sum_z e^{-L(z;\hat{y})}=1,\forall \hat{y}} \sum_{z\in\mathcal{Z}} \mathbb{P}_Z(z)\mathbb{E}_{\hat{Y}|z} L(z;\hat{Y}) \\
=& \min_{\sum_z e^{-L(z;\hat{y})}=1,\forall \hat{y}} \sum_{z\in\mathcal{Z}} \mathbb{P}_Z(z) \int f(\hat{y} \mid z) L(z;\hat{y}) d\hat{y} \\
=& \min_{\sum_z e^{-L(z;\hat{y})}=1,\forall \hat{y}} \int f(\hat{y}) \sum_{z\in\mathcal{Z}} \mathbb{P}_Z(z \mid \hat{y}) L(z;\hat{y}) d\hat{y}.
\end{aligned} \tag{15}$$

We can see $\hat{y}$ as a constant, the minimize problem becomes

$$\begin{aligned}
&\operatorname{argmin}_{L(z;\hat{y})} \sum_{z\in\mathcal{Z}} \mathbb{P}_Z(z \mid \hat{y}) L(z;\hat{y}) d\hat{y} \\
&\text{s.t.} \sum_z e^{-L(z;\hat{y})} = 1.
\end{aligned} \tag{16}$$

(16) is equivalent to the optimal code length problem under Kraft inequality (uniquely decodable) without the integer constraint. The details can be seen at Chapter 5.3 of *Elements of Information Theory*[6]. We know that it is lower bounded by the entropy $H(Z \mid \hat{Y} = \hat{y})$ with equality if and only $e^{-L(z;\hat{y})} = \mathbb{P}(z \mid \hat{y})$. Since $\hat{y}$ is arbitrary and $f(\hat{y}) \geqslant 0$

$$\begin{aligned}
&\min_{\sum_z e^{-L(z;\hat{y})}=1,\forall \hat{y}} \int f(\hat{y}) \sum_{z\in\mathcal{Z}} \mathbb{P}_Z(z \mid \hat{y}) L(z;\hat{y}) d\hat{y} \\
=& \int f(\hat{y}) H(Z \mid \hat{y}) = H(Z \mid \hat{Y}).
\end{aligned} \tag{17}$$

□

Having established the proof of the theorem, we can have a detailed examination of its implications. Deriving the empirical version of (12), we have

$$\max_{D_z(\hat{y}):\sum_z D_z(\hat{y})=1,\forall \hat{y}} \sum_{z\in\mathcal{Z}} \mathbb{P}_Z(z) \sum_{i:z^{(i)}=z} \frac{1}{m_z} \log D_z\left(\hat{y}^{(i)}\right) + H(Z). \tag{18}$$

where m_z is the number of observations with $Z = z$. Let m be the total number of observations. Using law of large numbers, $m_z \approx m\mathbb{P}_Z(z)$ when m is large enough, we can further simplify (18) to

$$\max_{D_z(\hat{y}):\sum_z D_z(\hat{y})=1,\forall \hat{y}} \sum_{z\in\mathcal{Z}} \sum_{i:z^{(i)}=z} \frac{1}{m} \log D_z\left(\hat{y}^{(i)}\right) + H(Z). \tag{19}$$

Notice that the first term we want to maximize is exactly the cross entropy loss $\mathrm{CE}(Z, D_z(Y))$ with a flip of sign. This justify the design of our architecture—If we train a discriminator according to cross entropy loss, the negative loss function will directly reflect the mutual information between the response and predictors[7].

4 Scope of analysis

In this report, we first reproduced the FR-Train model in Pytorch and inspected its training process. Then we carried out our own experiments, focusing on the following three questions.

(i) Alignment between discriminator loss and MI

Given that the optimization problem for disparate impact is non-convex [5], the authors proposed minimizing mutual information as an alternative. This approach is underpinned by the sufficient and necessary relationship between DI and mutual information, as previously discussed. Additionally, they employed cross entropy loss as a proxy for mutual information during the training process, supported by Theorem 1. Since the global minimum of (12) is not guaranteed, this leads to a natural question: how effectively does the discriminator loss mirror the behavior of mutual information in the training process?

(ii) Replace fair discriminator by MI estimators

In the FR-Train model, the effectiveness of the system relies on the proficient learning of both the generator and the discriminators. It is crucial for these two models to effectively learn and adapt in order to achieve optimal performance. However, it is noteworthy that GANs-like models often encounter challenges during training. These include mode collapse, vanishing gradients and more, making the model converge to local minimum or not converge at all. Since the fair discriminator is just a proxy for mutual information, can we estimate the mutual information directly so that we can remove the fair discriminator altogether?

(iii) Performance of FR-Train under imbalanced data

This concern comes from an important observation: While mutual information (MI) equates to zero if and only if Disparate Impact (DI) is one, the scenario becomes complex when the dataset comprises a small fraction of positive labels. Considering the inequality

$$I(Z, \hat{Y}) \leqslant H(\hat{Y}) \tag{20}$$

If the dataset is imbalanced, $H(\hat{Y})$ will be small, which means the mutual information is small as well. However, this does not automatically translate into a high DI value. This intuition is empirically demonstrated in our simulation, where 1000 pairs of probabilities $\{\mathbb{P}(\hat{Y} = 1), \mathbb{P}(\hat{Y} = 1|Z = 0)), \mathbb{P}(\hat{Y} = 1|Z = 1)\}$ are randomly generated from the standard uniform distribution. The relationship between MI and DI is shown in Figure 3.

It's evident that even when mutual information is negligible, the DI values still range between 0 and 1 evenly. Therefore we will examine the variation of FR-Train's performance when the datasets are increasingly imbalanced. We will also provide a possible solution to this program and compare the results.

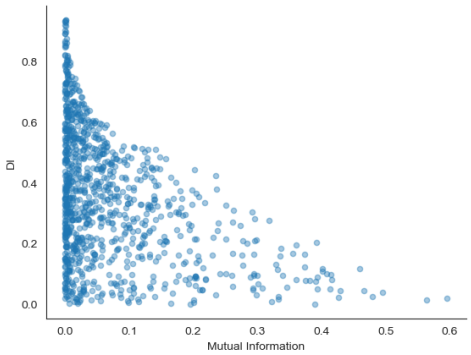


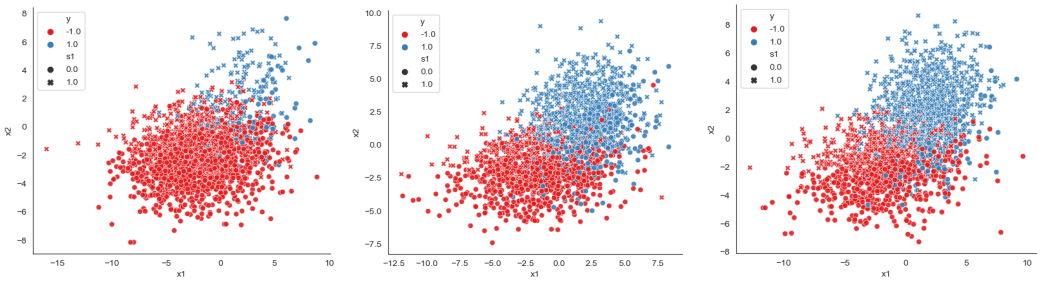
Figure 3: Mutual information and DI for 1000 randomly generated samples.

5 Methodology

5.1 Datasets and Metrics

In the first two experiments, we use the same clean and poisoned dataset provided by the authors, which contains 1800 training data, 200 validation data for robust discriminator and 1000 testing data. For each datapoint, it has two non-sensitive feature $X_1, X_2 \in \mathbb{R}$, one sensitive binary feature $Z \in \{0, 1\}$ and a binary label $Y \in \{0, 1\}$.

In the third experiment, we extended the data generation process from the original paper to create a family of datasets with imbalances in both the response variable Y and the sensitive attribute Z. This modification is facilitated by two parameters: $P_{Y=1}$ sets the probability of obtaining a positive label in the dataset, while c influences the likelihood of the sensitive feature Z being positive. The details of the process is presented in Appendix B. In Figure 4, we provide the visualizations of the datasets under various configurations.



(a) $p_{Y=1} = 0.1, c = 1$ (b) $p_{Y=1} = 0.5, c = 1$ (c) $p_{Y=1} = 0.5, c = 2$

Figure 4: Visualization of datasets generated using different parameters, y corresponds to the response label, which takes on values 1 or -1, $s1$ corresponds to the sensitive feature which takes on values 1 or 0.

In our experiments, we used accuracy and disparate impact, based on the prediction from the generator and ground truth, as our main metrics. The definition of disparate impact was shown in the preliminaries.

5.2 Hyperparameters

To provide a complete overview of the experiments, we use this section to list all the hyperparameters. For all the experiments we conducted, unless specified, the settings match those of Table 1.

Table 1: Default hyperparameters used for experiments.

Hyperparameters	Clean	Poisoned
Model	FR-Train	FR-Train
Iteration	4000	10000
Optimizer	Adam	Adam
Warmup	500	500
Updates ratio (Generator vs. Discriminator)	3	5
Learning rate (Fair discriminator)	0.01	0.001
Learning rate (Robust discriminator)	0.001	0.001

6 Results

(i) Alignment between discriminator loss and MI

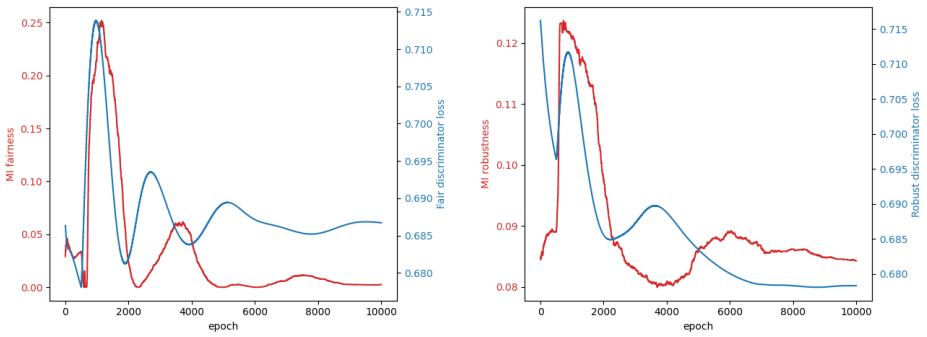
To evaluate the alignment between the proxy loss and mutual information, we adapted the original authors' code and rerun the whole experiment. At every iteration, we compute the mutual information using the prediction and compared it with the values of the loss function.

The calculation of $I(\hat{Y}, Z)$ for the fair discriminator is trivial since both variables are binary. When examining the robust discriminator, the mutual information is decomposed as follows:

$$I(V; \bar{X}, \bar{Z}, \bar{Y}) = I(V; \bar{X}) + I(V; \bar{Z}|\bar{X}) + I(V; \bar{Y}|\bar{X}, \bar{Z}) \tag{21}$$

We estimated these values using the `knncmi` package in Python, developed by Mesner et al. [8]. This package estimates conditional mutual information by using the nearest neighbors methodology to estimate the data distributions and is able to handle datasets with a mixture of discrete and continuous variables.

We present the graphs displaying the change of discriminator loss and estimated mutual information in the training process in Figure 5. Our result shows that after 2000 iterations into training, the discriminator loss and mutual information start to fluctuate in the opposite direction. This aligns with Theorem 1 and supports the authors' model design. However, in the first 2000 iteration, the discriminator haven't converges and the loss can't serve as an accurate proxy for mutual information.



(a) Fair discriminator (b) Robust discriminator

Figure 5: Evolution of discriminator loss and estimated mutual information with epochs: Training process of the original FR-Train model on poisoned data using author-tuned parameters.

(ii) Replacing fair discriminator by MI estimators

To address the challenges posed by adversarial networks, we proposed the following strategy: replacing the fair discriminator with direct estimation of mutual information (MI) in the generator's loss function. Therefore, the generator's loss, as previously defined in Equation 10, is modified as follows:

$$\mathcal{L}_G = (1 - \lambda_f - \lambda_r)\text{CE}_G - \lambda_f \mathcal{L}_{MI} - \lambda_r \mathcal{L}_R. \tag{22}$$

However, we can't use the definition to calculate mutual information between the binary prediction $\hat{Y}$ and sensitive feature Z directly because the logit outputs from the generator need to be converted to binary labels via a standard step function. This poses a challenge due to its non-differentiability at 0 and a gradient value of 0 in all other regions. To work around this issue, we introduced a parameterized sigmoid function with a narrow learning window, as depicted in Figure 6, to replace the step function. This modification ensures a well-defined gradient across all values. Subsequently, the logits are processed through this sigmoid function to generate a 'pseudo label'. This label is then utilized to calculate the empirical mutual information just as a binary label. We call this the naive smooth mutual information loss. We provide the details of our algorithm in Algorithm 1.

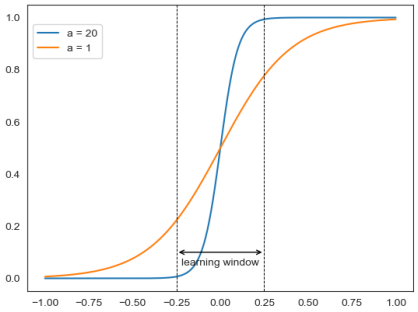
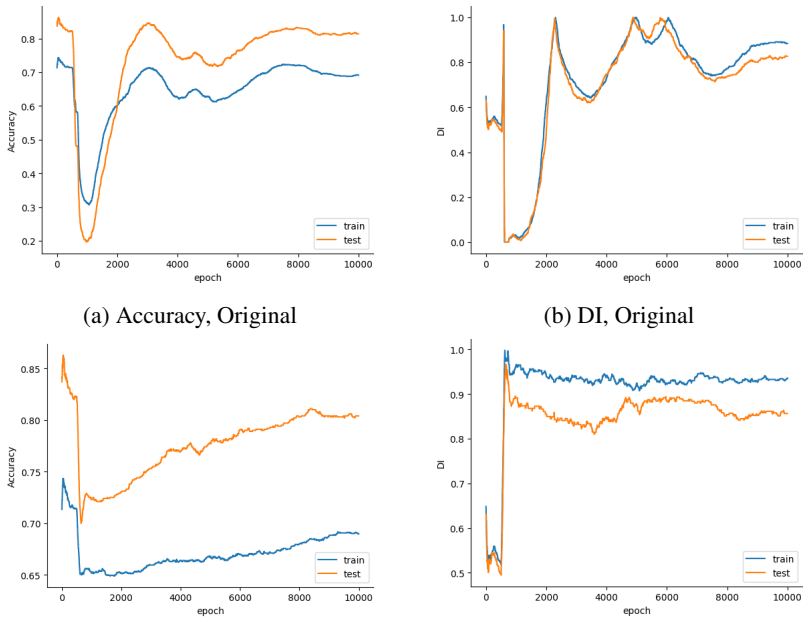


Figure 6: Examples of sigmoid function.

We implemented our modifications based on the code of the original FR-Train and trained it on the provided clean and poisoned datasets. During the training process, we monitored the model's performance to observe the impact of our changes. The comparison between our modification and original FR-Train is showcased in Figure 7.



(a) Accuracy, Original

(b) DI, Original

(c) Accuracy, Modified

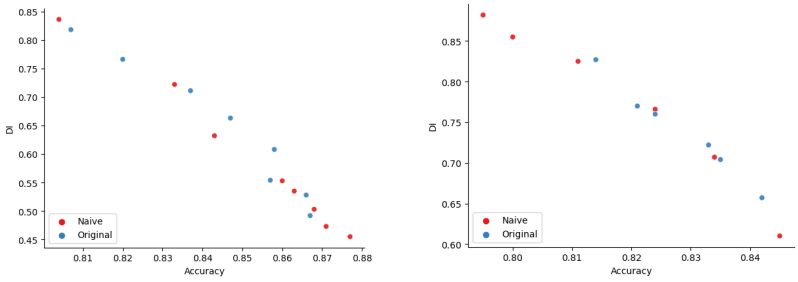
(d) DI, Modified

Figure 7: Comparison of performance change between original FR-Train and our modified model trained on poisoned data.

The results underscore the benefits of our proposed approach. For the original FR-Train model, both testing accuracy and DI exhibit significant fluctuations due to the dynamics of the adversarial

networks. Notably, there are instances where the DI abruptly peaks near 1 before rapidly declining, making the convergence of the model unpredictable. In contrast, our modified model demonstrates a more stable learning process, as the generator concurrently optimizes for accuracy and fairness. This stability suggests the potential for increasing the learning rate to expedite convergence.

Furthermore, we analyzed the accuracy-fairness trade-off curves as seen in Figure 8. The performance differences between the two models are minimal.



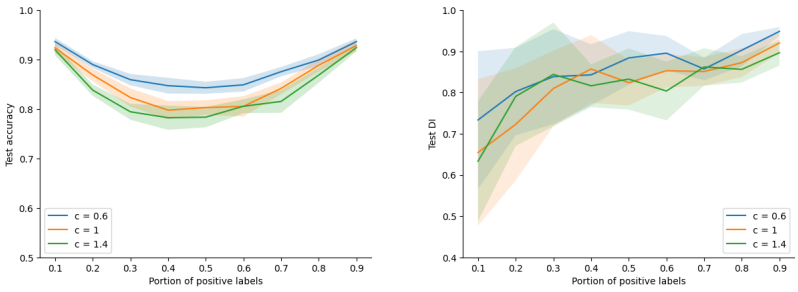
(a) Clean data (b) Poisoned data (10%)

Figure 8: Accuracy-Fairness trade-off curves of the original FR-Train and our modified model, tested on both clean and poisoned data.

(ii) Performance under imbalanced datasets

We evaluated the performance of FR-Train under varying conditions of data imbalance using a simulation approach. We generated 10 distinct datasets for each set of parameters $(p_{Y=1}, c)$, where $p_{Y=1} \in \{0.1, 0.2, \ldots, 0.9\}$, and $c \in \{0.6, 1, 1.4\}$. For each dataset, we rigorously cross-validated the optimal tuning parameters, aiming to optimize the combined measure of Accuracy + DI. The performance metrics were then assessed on a separate testing set to ensure unbiased evaluation.

The outcomes of the performance evaluation are illustrated in Figure 9. Notably, the test accuracy remains relatively unchanged across various levels of data imbalance. However, the testing disparate declines significantly and displays higher variance when the datasets have fewer positive target samples. Furthermore, we found that both accuracy and disparate impact appear to be largely unaffected by the proportion of samples featuring the positive sensitive attribute.



(a) Test accuracy (b) Test DI

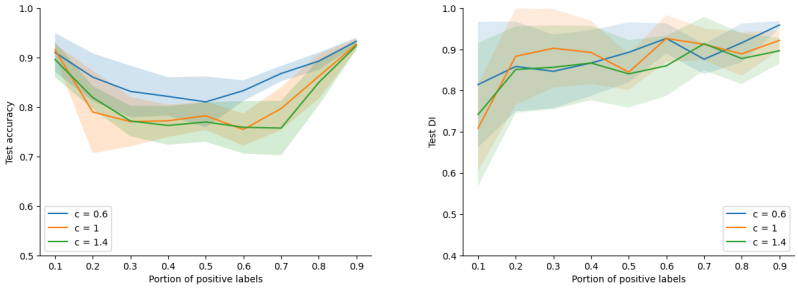
Figure 9: Performance variation of the original FR-Train under imbalanced data.

To address the observed fairness deterioration, we proposed "normalized mutual information". It is defined as

$$\mathcal{L}_{MI} = \frac{I(Z, \hat{Y})}{\hat{H}(Y)}. \tag{23}$$

where $\hat{H}(Z)$ s the entropy estimated from the training set, and the mutual information $I(Z, \hat{Y})$ is calculated using the naive method previously described. We conducted the same imbalance experiment after modification and evaluated both accuracy and disparate impact. The results,

illustrated in Figure 10, demonstrate a significant improvement in mitigating the decrease in test disparate impact. However, it is important to note that the test accuracy under this new model exhibited greater variance compared to the original FR-Train model. This indicates a potential trade-off between accuracy variance and fairness, which needs further exploration in future research.



(a) Test accuracy (b) Test DI

Figure 10: Performance variation of our modified model using normalized mutual information loss under imbalanced data.

7 Discussion

Our exploration of FR-Train's discriminator modules sheds light on the intricate architecture design required to achieve both fairness and robustness. Our theoretical and technical analyses support the original authors' claims regarding the representation of mutual information by the discriminators. However, we also noted that the original model faces challenges in converging due to its adversarial training dynamics.

In our experiments, we replaced the fair discriminator with our direct naive mutual information estimation. This approach not only led to a more stable training process but also maintained equivalent performance, demonstrating its advantages.

The performance analysis under extreme data imbalance scenarios reveals a crucial aspect of AI fairness: traditional metrics like mutual information may suffer from severe performance deterioration in imbalanced datasets. Our proposal of normalized mutual information as a loss function relatively mitigates this issue, but at the cost of some accuracy.

These findings lead us to propose several areas for future investigation based on FR-Train. One important direction is the analysis of the trade-offs between fairness and accuracy and how it's related to data imbalance. Understanding the dynamics of these trade-offs could lead to the development of more sophisticated models that achieve superior performance. Another promising area is the theoretical guarantee for the robust discriminator and designing a better approach to achieve robustness without the requirement for the clean validation dataset.

8 Conclusion

To conclude, our study on FR-Train's discriminator modules underlines the ongoing challenges in developing AI systems that are both fair and robust. Our re-implementation results support the original author's model design. Moreover, we proposed naive mutual information estimate and normalized mutual information to successfully stabilize the training process and enhance FR-Train's performance in data imbalance scenarios.

References

[1] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. Trustworthy ai: From principles to practices, 2022.

[2] Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In Marina Meila and Tong Zhang, editors, *Proceedings of the*

38th International Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, pages 11492–11501. PMLR, 18–24 Jul 2021.

[3] Yuji Roh, Kangwook Lee, Steven Whang, and Changho Suh. FR-train: A mutual information-based approach to fair and robust training. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8147–8157. PMLR, 13–18 Jul 2020.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[5] Jaewoong Cho, Gyeongjo Hwang, and Changho Suh. A fair classifier using mutual information. *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2521–2526, 2020.

[6] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[7] Jirong Yi, Qiaosheng Zhang, Zhen Chen, Qiao Liu, and Wei Shao. Mutual information learned classifiers: an information-theoretic viewpoint of training deep learning classification systems, 2022.

[8] Octavio César Mesner and Cosma Rohilla Shalizi. Conditional mutual information estimation for mixed discrete and continuous variables with nearest neighbors, 2019.

Appendix A Naive mutual information estimation

Algorithm 1 Naive Smooth Mutual Information Loss

Require: S: Target variable values, Y: Output logits, w (optional): weight vector
Ensure: Mutual Information (MI)

if w is given **then**
  $w_i \leftarrow \frac{w_i}{\sum_{i=1}^{n} w_i}, i = 1, 2, \ldots, N$ ▷ Make sure weights add up to 1
else
  $w_i = \frac{1}{n}, i = 1, 2, \ldots, N$
end if
$Y_i = \sigma(Y_i, \beta = 20)$
$p_{S1} \leftarrow \sum_{i=1}^{N} S_i \cdot w_i$ ▷ Calculate marginal probabilities
$p_{S0} \leftarrow \sum_{i=1}^{N} (1 - S_i) \cdot w_i$
$p_{Y1} \leftarrow \sum_{i=1}^{N} Y_i \cdot w_i$
$p_{Y0} \leftarrow \sum_{i=1}^{N} (1 - Y_i) \cdot w_i$
$p_{S1Y1} \leftarrow \sum_{i=1}^{N} S_i \cdot Y_i \cdot w_i$ ▷ Calculate joint probabilities
$p_{S1Y0} \leftarrow p_{S1} - p_{S1Y1}$
$p_{S0Y1} \leftarrow p_{Y1} - p_{S1Y1}$
$p_{S0Y0} \leftarrow 1 - p_{S1Y1} - p_{S1Y0} - p_{S0Y1}$
$p_{S1|Y1} \leftarrow \frac{p_{S1Y1}}{p_{Y1}}$ if $p_{Y1} \neq 0$ else 0 ▷ Calculate conditional probabilities
$p_{S1|Y0} \leftarrow \frac{p_{S1Y0}}{p_{Y0}}$ if $p_{Y0} \neq 0$ else 0
$p_{S0|Y1} \leftarrow \frac{p_{S0Y1}}{p_{Y1}}$ if $p_{Y1} \neq 0$ else 0
$p_{S0|Y0} \leftarrow \frac{p_{S0Y0}}{p_{Y0}}$ if $p_{Y0} \neq 0$ else 0
$H_S \leftarrow -p_{S1} \log(p_{S1} + 0.1^{10}) - p_{S0} \log(p_{S0} + 0.1^{10})$ ▷ Calculate entropy
$H_{S|Y} \leftarrow -(p_{S1Y1} \log(p_{S1|Y1} + 0.1^{10}) + p_{S1Y0} \log(p_{S1|Y0} + 0.1^{10}) + p_{S0Y1} \log(p_{S0|Y1} + 0.1^{10}) + p_{S0Y0} \log(p_{S0|Y0} + 0.1^{10})$
MI $\leftarrow H_S - H_{S|Y}$
return MI

Appendix B Imbalanced data generation

Algorithm 2 Data Generation Process

Require: $n \in \mathbb{N}_+$, $0 < p_{Y=1} < 1$, $c > 0$
Ensure: Dataset (X, Y, Z)
Initialize $X = [\,]$, $Y = [\,]$, $S = [\,]$
for $i = 1$ to n **do**
  $U_1 \sim \text{Uniform}(0, 1)$
  if $U_1 < p_{Y=1}$ **then**
    $x \sim \mathcal{N}_1 \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 5 & 1 \\ 1 & 5 \end{bmatrix} \right)$
    $y \leftarrow 1$
  else
    $x \sim \mathcal{N}_2 \left(\begin{bmatrix} -2 \\ -2 \end{bmatrix}, \begin{bmatrix} 10 & 1 \\ 1 & 3 \end{bmatrix} \right)$
    $y \leftarrow -1$
  end if
  $x' \leftarrow \begin{bmatrix} \cos\left(\frac{\pi}{4}\right) & -\sin\left(\frac{\pi}{4}\right) \\ \sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) \end{bmatrix} \cdot x$
  $p_{S=1} \leftarrow c \cdot \frac{\mathcal{N}_1(x')}{\mathcal{N}_1(x') + \mathcal{N}_2(x')}$
  $U_2 \sim \text{Uniform}(0, 1)$
  if $U_2 < p_{S=1}$ **then**
    $s \leftarrow 1$
  else
    $s \leftarrow 0$
  end if
  $X \leftarrow X \cup \{x\}$, $Y \leftarrow Y \cup \{y\}$, $S \leftarrow S \cup \{s\}$
end for
return X, Y, S