10 Years of Fair Representations: Challenges and Opportunities

Mattia Cerrato [*][†]
mcerrato[@]uni-mainz.de

Marius Köppel [*][‡]
mkoepp[@]phys.ethz.ch

Philipp Wolf [*][†]
pwolf01[@]students.uni-mainz.de

Stefan Kramer [†]
kramer[@]informatik.uni-mainz.de

Abstract

Fair Representation Learning (FRL) is a broad set of techniques, mostly based on neural networks, that seeks to learn new representations of data in which sensitive or undesired information has been removed. Methodologically, FRL was pioneered by Richard Zemel et al. about ten years ago. The basic concepts, objectives and evaluation strategies for FRL methodologies remain unchanged to this day. In this paper, we look back at the first ten years of FRL by i) revisiting its theoretical standing in light of recent work in deep learning theory that shows the hardness of removing information in neural network representations and ii) presenting the results of a massive experimentation (225.000 model fits and 110.000 AutoML fits) we conducted with the objective of improving on the common evaluation scenario for FRL. More specifically, we use automated machine learning (AutoML) to adversarially "mine" sensitive information from supposedly fair representations. Our theoretical and experimental analysis suggests that deterministic, unquantized FRL methodologies have serious issues in removing sensitive information, which is especially troubling as they might seem "fair" at first glance.

1 Introduction

Biased machine learning systems have been shown to have detrimental impacts on society, perpetuating social inequalities and reinforcing harmful stereotypes. For instance, in Amazon's attempt to automate its hiring process, the company's computer programs, developed since 2014, reportedly aimed to streamline talent acquisition by analyzing resumes. However, the system was reported to display gender bias, penalizing resumes containing terms like “women's,” disadvantaging female applicants for technical roles [17]. Similarly, in the United States, algorithms like COMPAS have been used in nine states to assess a criminal defendant's risk of recidivism. An analysis of COMPAS revealed discriminatory outcomes: black defendants who did not recidivate were more frequently misclassified as high risk compared to their white counterparts, while white re-offenders were often mislabeled as low risk [5].

Both examples show the concern that such models trained on biased data might then learn those biases [7], therefore perpetuating historical discrimination against certain groups of individuals. Machine learning methodologies designed to avoid these situations are often said to be “group-fair” in the sense that they seek to distribute resources equally across groups. This paper focuses on a specific kind of algorithm – Fair Representation Learning (FRL) – which is part of this domain.

[*] These authors contributed equally.

[†] Institute of Computer Science Johannes Gutenberg-Universität Mainz, Germany

[‡] Institute for Particle Physics and Astrophysics, ETH Zurich, Switzerland

Preprint. Under review.

FRL is a broad set of techniques that seek to remove undesired information from data. FRL is based mostly but not exclusively on neural network techniques. Our focus in this paper is however the theoretical and experimental evaluation of neural network-based FRL. The goal of such techniques is to learn the parameters θ for a projection $f_\theta : X \rightarrow Z$ from the feature space X to a latent feature space Z. The task was pioneered by Zemel et al., about ten years ago [57].

The two competing goals for Z are to remove all information about a sensitive attribute S while retaining as much information as possible about some task for which labeled data Y is available. An alternative formulation is based on the auto-encoding concept: information about X should still be present as much as possible in Z. While it is of course possible to simply remove S from the dataset columns, this does not generally prevent statistical inference on S. Discarding sensitive data is usually termed "fairness by unawareness" and does not in general grant group-anonymity (we refer the interested reader to the book by Barocas et al. [7], Chapter 3, Figure 3.4). A simple way to understand this phenomenon is to reason about the correlation between ZIP code (sometimes deemed non-sensitive information) and ethnicity (usually deemed sensitive) in the US and other countries. FRL improves on fairness by unawareness by actively seeking to "stamp out" and remove any correlation between the learned representation Z and the sensitive information S. It has been observed in practice in the last 10 years that information removal contributes to other fairness metrics such as independence/disparate impact or separation/disparate mistreatment [52, 35, 11, 32, 28].

One advantage of these methodologies is that any classifier can be trained on the learned fair representation [57], while other methods may rely on a specific model or technique. Due to this, FRL enables a "separation of concerns" scenario [37]. Here, a data user is assumed to be interested in developing an ML-based automated or semi-automated decision-making system for which fairness concerns are relevant. A trusted data regulator, who is also allowed access to sensitive information, will then employ an FRL algorithm and share the obtained fair representations privately with the data user. This setup gives the opportunity for increased trust into the overall ML-based decision-making system, as the regulator would able to evaluate the amount of correlation between Z and S while the user will not have access to S or any of its correlations. It is relatively common for work in FRL to perform the above investigation by training some number of classifiers on Z and observing whether their performance is close enough to random guessing for the dataset at hand. If it is, then this provides some empirical evidence that an FRL method is working as intended.

All these advantages notwithstanding, it is not straightforward to conclude that FRL should be the go-to methodology for fairness-sensitive applications. One significant limitation here is that neural network-based FRL is not transparent and quite hard to interpret [13]. When fairness is relevant, the application is by default high-stakes [44]: it is then hard to justify employing neural networks, esp. when the data is tabular and other methodologies are therefore better than, or at least competitive with, FRL [25]. The one advantage that remains unique to FRL is therefore, in our view, the aforementioned separation of concerns scenario.

In this paper, we revisit the first 10 years of fair representation learning and discuss its unique limitations and opportunities for real-world impact. We start by summarizing the most visible contributions in this area and how they relate to one another. Then, we discuss the general theoretical setup of FRL and discuss its limitations by relating them to theoretical advancements in understanding the information dynamics of deep neural networks [23]. We then move on to showing the result of a massive experimentation – a total of around 225.000 model fits and 110.000 AutoML fits – we ran across 6 datasets. We release `EvalFRL`, the experimental library we developed for severe testing of FRL methodologies, which can found at `https://anonymous.4open.science/r/EvalFRL/`.

2 Related Work

Algorithmic fairness has garnered considerable interest from both academia and the general public in recent years, largely due to the ProPublica/COMPAS controversy [5, 45]. However, the earliest contribution in this field appears to date back to 1996, when Friedman and Nisselbaum [21] highlighted the necessity for automatic decision systems to be aware of systemic discrimination and ethical considerations. The importance of addressing automatic discrimination is also reflected in EU legislation, particularly in the GDPR, Recital 71 [36]. One approach to addressing these issues involves eliminating the influence of the "nuisance factor" S from the data X through fair representation learning. This method involves learning a projection of X into a latent feature space Z

where all information about S has been removed. A pioneering contribution to this area is by Zemel et al. [57]. Since then, neural networks have been widely employed in this context. Some approaches [52, 35] use adversarial learning, a technique introduced by Ganin et al. [22], which involves two networks working against each other to predict Y while removing information about S. Another line of research [32, 39] uses variational inference to approximate the intractable distribution $p(Z \mid X)$. This involves combining architectural design [32] and information-theoretic loss functions [39, 24] to promote the invariance of neural representations with respect to S. Recently, neural architectures have been proposed for other fairness-related tasks such as fair ranking [56, 10, 41] and fair recourse [47].

Another line of investigation that focuses on the information theory of DNNs and provides context for this work is the information bottleneck (IB) problem [49] and its applications to the understanding of deep neural networks (DNNs) training dynamics. Originally, Swartz-Ziv and Tishby [48] put forward the idea of computing the mutual information term $I(X; Z)$ via quantization and observed that deeper networks undergo a faster compression phase – a reduction of $I(X; Z)$ that happened earlier in the training process. These results inspired a reproducibility study by Saxe et al. [46], who observed information compression in networks that employ certain non-linearities (tanh, sigmoids), but no compression when other activations were considered (ReLU). With regard to the original investigation [48], Polyanskiy and Goldfeld [23] retorted that computing $I(X; Z)$ via quantization introduces quantization artifacts and that compression of $I(X; Z)$ is theoretically impossible in deterministic DNNs with injective or bi-Lipschitz activation functions. Another limitation was described by Amjad and Geiger [4], who contributed an analysis of the IB framework under discrete datasets, concluding that the IB functional (its optimization objective) is piecewise constant and is therefore hard to optimize with gradient descent and its variants. To the best of our knowledge, these fundamental results in the information theory of deep learning have not been analyzed in the context of FRL.

3 Challenges in Fair Representation Learning

Let us denote the dataset of individuals as a matrix $X \in \mathcal{X}^{n \times d}$, where each individual $i \in 1 \dots n$ is described by a feature vector x_i with d dimensions. The sensitive attribute is denoted with the random variable $S \in \mathcal{S}$, and the corresponding labels are denoted as $Y \in \mathcal{Y}$. In fair representation learning, the goal is to learn a representation $Z \in \mathcal{Z}^{n \times m}$ of the data such that it preserves relevant information for the task at hand while removing information about S. Usually, but not necessarily, $m < d$. With a slight abuse of notation, we will discuss X, Y, Z as random variables with their sample spaces being $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, respectively. Concretely, we define $\phi^i(x) = \sigma(A^i \phi^{i-1}(x) + b^i)$ as the i-th layer function of a deep neural network with L layers, where A is a matrix of real-valued weights and b is a bias vector. We assume that σ is applied to each dimension of its argument without any aggregation, as is common in DNNs. We note that $\phi^0(x) = x$ and $\phi^L(x) = \hat{y} \in \hat{Y}$, the prediction or reproduction of Y. Thus, we name $Z^i = \phi^i(x)$ as the random variable representing the representation extracted from the data by the i-th layer of the network.

It follows that if for some $i < L$ it is true that $Z^i \perp S$, then the output $\hat{Y}$ of the network will also be independent of S [35], leading for instance to the "independence" definition of group fairness in classification [7]. To achieve this goal, most fair representation learning approaches employ a loss function with two terms: a classification loss to ensure predictive performance on the task of interest, and a fairness loss to encourage fairness in the learned representation. Therefore, the overall objective function for fair representation learning can be formulated as a combination of the classification loss and the fairness loss. This can be achieved using a weighted sum of the two losses, where the relative importance of each component is controlled by the hyperparameter γ:

$$\min_{\theta} \, (1 - \gamma)\mathcal{L}_{\text{class}}(\theta) + \gamma \mathcal{L}_{\text{fair}}(\theta) \tag{1}$$

where θ represents the parameters of the model, $\mathcal{L}_{\text{class}}$ is the classification loss, $\mathcal{L}_{\text{fair}}$ is the fairness loss. As discussed by various authors [2, 15], it is possible to formulate this task in an information-theoretic manner by relying on mutual information. A theoretical formulation of fair representation learning using mutual information between Z and S can be defined by starting from the mutual information between representation and sensitive data:

$$\mathcal{L}_{fair}(\theta) = I(Z;S) = \int_{s\in S}\int_{z\in Z} P(z,s)\log\frac{P(z,s)}{P(z)P(s)} \tag{2}$$

where $P(z,s)$ is the joint probability density and $P(z)$, $P(s)$ are the marginal probability distributions of Z and S, respectively. Usually, in FRL S is taken to be discrete, representing some quantized sensitive characteristic that may lead to unacceptable harm, discrimination, or both. To achieve a fair or invariant representation, this term needs to be minimized. Ideally, at the same time the representation Z would be informative for the prediction task, i.e., it would retain sufficient information about the labels Y.

$$\begin{aligned} &\min_{Z} I(Z;S). \\ &s.t.\ I(Z;Y) \geq \alpha \end{aligned} \tag{3}$$

The trade-off between preserving task-relevant information and minimizing the mutual information with the sensitive attribute is the key challenge in fair representation learning.

FRL techniques are commonly evaluated over two different frames:

- **Fair allocation.** Suppose that certain values of $\hat{Y}$ lead to desirable outcomes for the individuals represented in X. In classification, this may be easily understood as $\hat{Y} = 1$ representing, for instance, being selected for a job interview by a CV-scanning application. Then, a FRL technique succeeds if obtains a fair allocation of desirable outcomes by removing information about S in Z and then using Z in a further classification stage of the network. The fairness of the allocation is then computed via any application-relevant metric, e.g. discrimination [57, 52, 32, 11], disparate mistreatment [54], etc..
- **Invariant representations.** Suppose that the representation Z is computed by some trusted party that is allowed access to both X and S. Then, a FRL technique succeeds if it may be employed by this trusted party to obtain Z such that $Z \perp S$. In practice, this may be evaluated by training a supervised classifier on Z and computing its accuracy in predicting S. Invariant representations may then be safely distributed to data users which may use them to train any ML methodology which will be by construction unaware about S and any of its correlations to X.

Fair allocation is a high-stakes task for which, however, interpretability may be required, on ethical [44] or even legal [36] grounds. Interpretable FRL is an active area of research [30, 14] and it is in general not straightforward to interpret the meaning of Z. Currently, it may be preferable to employ better-understood methodologies, such as fair reductions [55]. If it is acceptable to use S at test time, post-processing techniques are provably optimal [27]. Thus, learning invariant representations would be the main – and nominal – selling point of FRL. We report in the following some relatively well-known results in the information theory of deep learning whose consequences for FRL, to the best of our knowledge, have not been previously discussed.

Information and Mutual Information in Neural Networks

The optimization problem in Eq. 3 bears a close resemblance to the information bottleneck (IB) problem introduced in [49] and then famously applied to neural network training dynamics [50]. The authors propose to understand learning representations as the problem of compressing X into Z while losing minimal information about Y. The only significant difference with the information-theoretic formulation of FRL is then that X is substituted by S. Then, we prove in the following that previous work on mutual information in deep neural networks [23, 16] also applies to FRL:

Theorem 1. *Let X, Y, and S be the random variables representing data, labels and the sensitive attributes, respectively. Let $\phi^i(x) = \sigma(A^i\phi^{i-1}(x) + b^i)$ be the i-th layer function of a DNN, where A is a weight matrix, b a bias vector, and σ an injective non-linearity. Let $Z^i = \phi^i(x)$ be a random variable. Let thus $S \rightarrow Y \rightarrow X \rightarrow \cdots \rightarrow Z^k \rightarrow \dots \hat{Y}$ be a Markov chain, where $\hat{Y}$ is an estimation of Y. Then, $I(X;S) = I(Z^i;S)\ \forall i \in \{1\dots L\}$, where L is the number of layers in the network.*

Proof. We note that each $Z^i = \phi^i(x)$ is a deterministic, one-to-one mapping of the previous layer's input, or of the input itself. Thus, $H(X) = H(Z^i)\ \forall i \in \{1\dots M\}$ [23, 16]. Then, we rewrite the mutual information terms as follows:

$$I(S;X) = H(X) - H(X \mid S)$$
$$I(S;Z^i) = H(Z^i) - H(Z^i \mid S)$$

since $H(X) = H(Z^i)$, it follows that the theorem is true if

$$H(X \mid S) = H(Z^i \mid S) \tag{4}$$

We first note that the joint entropies $H(Z^i, S)$ and $H(X, S)$ are equal as Z^i is computed via a one-to-one mapping of X [43]. Then, by the chain rule of entropy, we have $H(Z^i|S) = H(Z^i, S) - H(S)$ and $H(X|S) = H(X, S) - H(S)$. Substituting these equalities in Eq. 4 concludes the proof. □

This result derives straightforwardly from the fact that deterministic neural networks with injective activation functions are one-to-one mappings of the input data. Thus, two different $x_1, x_2 \in \mathcal{X}^{n \times d}$ will always be mapped onto two different representations $z_1^i, z_2^i \in \mathcal{Z}_i^{n \times m}$, even if $m < d$. It follows that sensitive information is not removed in general when descending the layers of a neural network. Therefore, Theorem 1 is an impossibility theorem for FRL on infinite-precision deterministic networks with tanh or sigmoid activations, and may be extended to bi-Lipschitz functions [23] such as Leaky-ReLU. It is important to note that non-injective activations such as ReLU escape the theorem; as pointed out by Amjad and Geiger [4], however, another practical limitation applies. Specifically, $I(Z; X)$ will only take finitely many values provided that the data features X is discrete. In FRL, S is usually assumed to be discrete and thus $I(Z; S)$ is piecewise constant, which makes for a difficult objective to optimize for. Another caveat is provided by the fact that invariance in the mutual information does not imply invariance in its estimation. Supervised classifiers trained on (Z, S) may as well display a lesser degree of accuracy compared to ones trained on (X, S) – as data samples with different values of $s \in \mathcal{S}$ get mapped closer together, it may be harder in practice to distinguish between them. It is also critical to note here as well that Z^i does not have infinite precision on a discrete computer [46]. The sigmoid function only saturates to 1 as $x \to \infty$; however, a computer will return 1 as the result of $\frac{1}{e^{-x}+1}$ much sooner than that, depending on how many bits are used to represent the result of the computation. Using low-bit representations, or "hard" clusterings, is therefore another way to escape the limitations put forward by Theorem 1 [15, 30]. Lastly, we note that Theorem 1 does not apply to neural network models that incorporate stochasticity in their process, as $H(X)$ is in general not equal to $H(Z^i)$ in that situation.

4 Experiments

In this section, we report on an in-depth experimentation that we conducted with the aim of evaluating the significance of the impossibility theorem stated in the previous section. We are especially interested in testing whether it is still possible to predict the sensitive attribute S with a high degree of accuracy from the fair representations learned by deterministic FRL methodologies, as our theoretical result suggests that it should be so. To this end, we evaluated a total of 8 FRL methodologies: BinaryMI and DetBinaryMI, [15], DebiasClassifier [22, 52, 35], NVP [12], VFAE [32], ICVAE [39], LFR [57] and Deep Domain Confusion [51]. While some of these are fully determinisic, others incorporate stochasticity by sampling from a parametric distribution which parameters are learned in the network. Our expectation is that stochastic models will show greater invariance in their learned representations, as Theorem 1 does not apply and sensitive information may in principle be removed. We tested these models on six different commonly employed fairness-sensitive datasets, focusing on the task of fair classification and independence as a fairness definition (i.e. $\hat{Y} \perp S$). We give in-depth descriptions of models and datasets in the Appendix, Sections A.1 and A.2, respectively.

We release `EvalFRL`[4], the experimental framework we employed to perform our experimentation alongside all experimental metadata, including best hyperparameters for all models and performance at the outer fold level[5] [6]. Our framework was developed to perform in-depth evaluation of FRL methodologies across the two main frames discussed in Section 3 – fair allocation and invariant

[4] `https://anonymous.4open.science/r/EvalFRL/`

[5] `https://drive.google.com/drive/folders/1koZd8cgBJMVGuH3uRqvpTFEUJo0Sd23q?usp=sharing`

[6] We discuss the used compute infrastructure to run the experiments in the Appendix E.

representations. Thus, we evaluate both the fairness of the allocations learned by FRL methods and approximate the mutual information between the representations Z and the sensitive attribute S with the performance of external estimators.

4.1 `EvalFRL`: An Evaluation Library for Fair Representation Learning

As mentioned above, if an estimator can predict S better than random guessing, this indicates that information on S is still contained in the fair representation. To investigate whether this may happen in commonly employed FRL techniques, we developed the `EvalFRL` framework, wherein every tested dataset-model combination follows a standardized testing pipeline. This process is fully reproducible, thereby ensuring comparability between the models. In previous work tackling FRL, the experimental setup has focused on training a few classifiers on (Z^i, S) [11, 57, 32, 52]. However, the number of classifiers may vary and the optimal hyperparameters are not always reported, leading to results which are hard to compare across different papers. We employ automated machine learning (AutoML) to handle this problem. AutoML automatically searches for the optimal machine learning solution for a given problem. This includes, among others, feature preprocessing, model selection and hyperparameter tuning.

Algorithm 1 Overview of `EvalFRL` logic when ran for one dataset-model combination.

```
$r \leftarrow 15$, $k \leftarrow 3$, $seed \leftarrow 123$, $results \leftarrow []$, $repr_{\text{train}} \leftarrow []$, $repr_{\text{test}} \leftarrow []$
for $i$ in $1$ to $r$ do
    $X_{\text{train}}, X_{\text{test}}, y_{\text{train}}, y_{\text{test}}, s_{\text{train}}, s_{\text{test}} \leftarrow \text{TRAIN\_TEST\_SPLIT}(X, y, s, \frac{2}{3}, \text{random\_state} = seed)$
    $cv \leftarrow \text{RANDOMIZEDSEARCHCV}(model, param\_distributions, cv = k, n\_iter = 100)$
    $\text{BESTMODEL} \leftarrow cv.\text{FIT}(X_{\text{train}}, y_{\text{train}}, s_{\text{train}}, \gamma = 0)$
    for $\gamma_0$ in $\{0, 0.1, \ldots 1\}$ do
        $\text{BESTMODEL.FIT}(X_{\text{train}}, y_{\text{train}}, s_{\text{train}}, \gamma = \gamma_0)$    ▷ bestmodel is fit from scratch
        $results.\text{APPEND}(\text{EVALUATION}(\text{BESTMODEL}, X_{\text{test}}, y_{\text{test}}, s_{\text{test}}))$
        $repr_{\text{train}}.\text{APPEND}(\text{GET\_REPRESENTATIONS}(\text{BESTMODEL}, X_{\text{train}}))$
        $repr_{\text{test}}.\text{APPEND}(\text{GET\_REPRESENTATIONS}(\text{BESTMODEL}, X_{\text{test}}))$
    end for
    for $repr_{train_i}, repr_{test_i}$ in $(repr_{\text{train}}, repr_{\text{test}})$ do
        $AutoML \leftarrow \text{AUTOML}()$    ▷ AutoML is trained from scratch at each CV iteration
        $AutoML.\text{FIT}(repr_{train_i}, s_{\text{train}})$
        $results.\text{APPEND}(\text{EVALUATION}(AutoML, repr_{test_i}, s_{\text{test}}))$
    end for
end for
```

We show in Algorithm 1 the main use case of `EvalFRL`. A detailed description of the steps and a graphical representation of the framework's logic is shown in the Appendix A. The whole pipeline is built using the Kedro framework [3] and can be easily extended to include other models, datasets and metrics beyond the ones we consider in this work. The data preprocessing step starts with the segmentation of the data into features X, the label y and the sensitive attribute S. Additionally, y gets transformed to either a positive ($y = 1$) or a negative ($y = 0$) outcome and S to either the privileged ($S = 1$) or underprivileged ($S = 0$) group. Categorical features undergo encoding, while continuous features are normalized with mean 0 and variance 1. In the subsequent step, hyperparameter-tuning is performed utilizing r-times k-fold cross-validation [9]. In our experiments we set $r = 15$ and $k = 3$ by following the recommendations in Naudeau and Bengio [40]. The best hyperparameters found in every outer-loop, along with a range of γ values regulating the fairness/accuracy tradeoff (Equation 1), are then utilized to evaluate the model. The evaluation step contains the model's predictive performance on the label y (Acc, AUC), as well as multiple fair allocation metrics. Finally, we seek to evaluate Z and S, for each fair representation generated in the previous step. We utilize an AutoML library called MLJAR [42] for this process. AutoML searches the best model and trains it using the representations on the train-data and the corresponding sensitive information S. Its performance gets tested using the representations on the held out test data and S.

The output of the framework allows for an exploration of the trade-off between predicting the label y and the fairness of the model according to known fairness-metrics. Additionally, it provides researchers with a common, high-effort evaluation framework for FRL methodologies in the invariant representation framing. If it is possible to have a higher performance than guessing S randomly, this leads to the conclusion that there is still information on S contained in the representation. These investigations are done over a range of γ tradeoff values, which makes it possible to understand the impact of the γ parameter on both representation invariance and fairness of allocations.

4.2 Fairness Metrics

Area under Discrimination Curve (AUDC) We quantify disparate impact through discrimination, following the approach introduced by Zemel et al. [57]. The discrimination metric, denoted as yDiscrim, is defined as:

$$\text{yDiscrim} = \left| \frac{\sum_{n:s_n=1}^{n} \hat{y}_n}{\sum_{n:s_n=1}^{n} 1} - \frac{\sum_{n:s_n=0}^{n} \hat{y}_n}{\sum_{n:s_n=0}^{n} 1} \right|, \tag{5}$$

where $n : s_n = 1$ indicates that the n-th example has a value of s equal to 1. To generalize this metric akin to how accuracy generalizes to obtain a classifier's area under the curve (AUC), we evaluate the above measure for different classification thresholds and then compute the area under this curve. In our experiments, we utilized 100 equispaced thresholds. We call this measure AUDC, following conventions established in the literature [13]. Unlike AUC, lower values are indicative of better performance.

rND To measure fairness in learning to rank applications, we use the rND metric [53]. This metric evaluates differences in exposure across multiple groups and is defined as:

$$\text{rND} = \frac{1}{Z} \sum_{i \in \{10,20,\ldots\}}^{N} \frac{1}{\log_2(i)} \left| \frac{| S^{+}_{1\ldots i} |}{i} - \frac{| S^{+} |}{N} \right| . \tag{6}$$

rND measures the difference between the ratio of the protected group in the top-i documents and in the overall population. The maximum value, Z, serves as a normalization factor and is computed with a dummy list where all members of the underprivileged group are placed at the end, representing "maximal discrimination." The metric also penalizes over-representation of protected individuals at the top compared to their overall population ratio.

4.3 Results in Fair Allocation

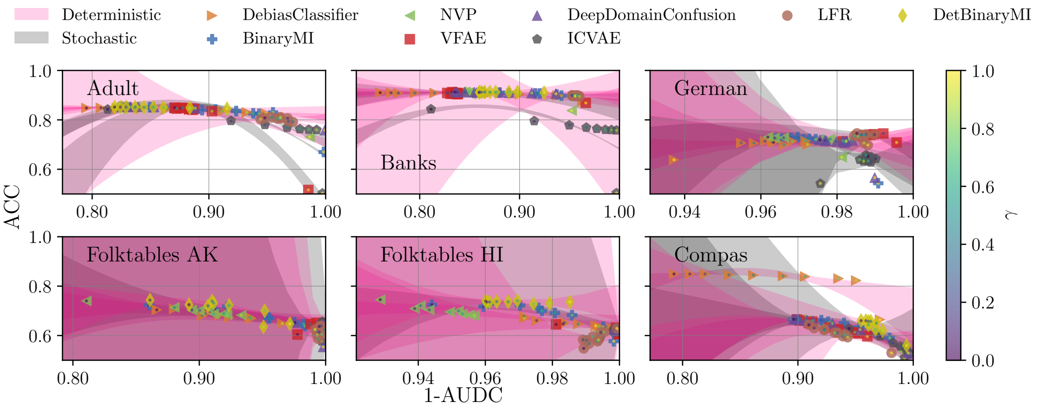

Figure 1: Accuracy vs. 1 - AUC-Discrimination tradeoff for all six dataset and eight model combinations. Each model is displayed for different γ values indicated via a colored point inside the model marker.

The model accuracy vs. fairness tradeoff results are shown in Figure 1 (ACC vs. 1-AUDC) and 2 (AUC vs. 1-AUDC), as well as in Figure 6 (ACC vs. 1-Discrimination), 7 (AUC vs. 1-Discrimination), 8 (ACC vs. Statistical Parity Difference), 9 (AUC vs. Statistical Parity Difference), 10 (ACC vs.

Delta), 11 (AUC vs. Delta), 12 (ACC vs. 1-rND), 13 (AUC vs. 1-rND) of the appendix. Each figure shows all combinations of the six datasets and eight models. The colored points in the symbols of the models show the used γ value, while the colored areas show the variance employing 100 Gaussian bootstrapping fits using the mean and variance of the model performance optained from the 15 hold-out splits. For most datasets and metric combinations (e.g. Adult, Banks and German) one can observe that the performance of the the majority of the models is equal. Most models show a well defined tradeoff behaviour when changing the γ value.

In Figure 1 we observe that the performance for the DebiasClassifier on the Compas dataset outperforms all other models. However, it takes mostly fair allocation decisions at higher values of gamma. We conclude that the performance of FRL methodologies in the task of fair allocation is approximately equal. Varying the tradeoff parameter γ, as expected, leads to fairer decisions.

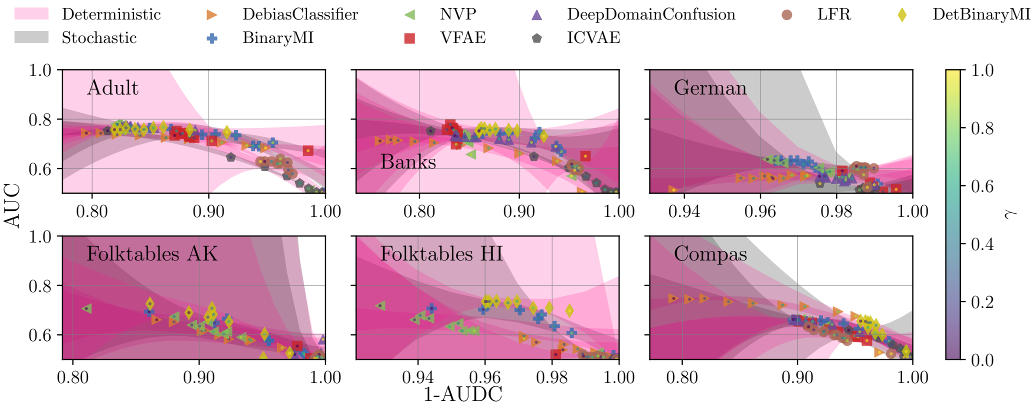

Figure 2: AUC vs. 1 - AUC-Discrimination tradeoff for all six dataset and eight model combinations. Each model is displayed for different γ values indicated via a colored point inside the model marker.

4.4 Results in Invariant Representations

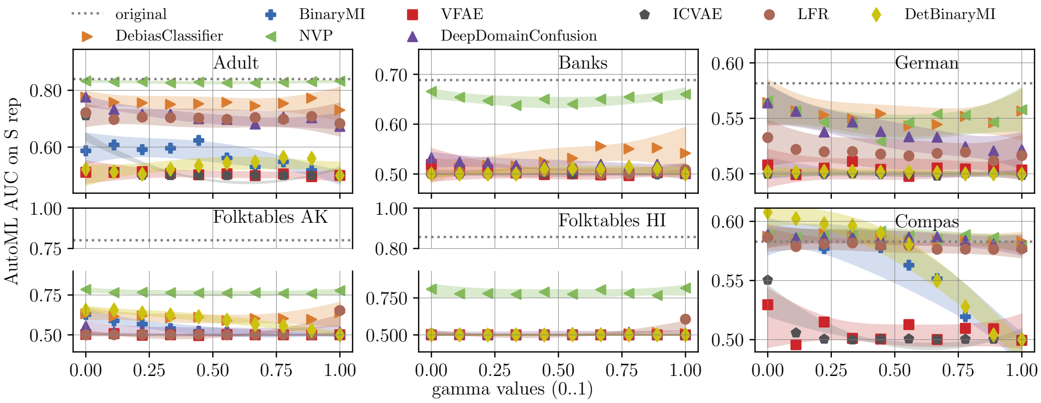

Figure 3: AutoML AUC vs. gamma results for all six dataset and eight model combinations.

We now take the same models reported in the previous subsection and investigate whether AutoML is able to recover information about the sensitive attributes from their representations. Here, we expect that the accuracy of AutoML will approach the proportion of the majority group in the dataset (Figure 3), and that the AUC will approach 0.5 (Figure 4), as the tradeoff parameter γ increases. The results are particularly striking: while this is indeed what happens for stochastic or quantized models (BinaryMI, DetBinaryMI, VFAE, ICVAE) at higher γ values, deterministic models have serious issues removing information (DebiasClassifier, NVP, DeepDomainConfusion, LFR) as the performance of AutoML remains well above random guess at many or all settings of γ. These results experimentally confirm the theoretical impossibility theorem of Section 3, and are of particular concern as they regard models that take overall fair allocation decisions. For instance, the DebiasClassifier is able to learn fair allocations on COMPAS; however, the representations still contain information about S and should therefore not be considered safe for distribution to data users interested in employing them in other ML tasks. We give a comparison of a ReLU-activated DebiasClassifier in the Appendix, Section C, where we observe similarly that information is not consistently removed. A similar trend is clearly visible for the NVP model across all datasets.

We conclude by offering an explanation for the phenomenon of fair allocation models not learning invariant representations. We note that the observation that when S and Y are correlated, a weak classification model will also be relatively fair in terms of allocation. Thus, FRL methodologies which are not severely tested for representation invariance may still obtain fair allocation decisions via a weak classification stage $Z^i \to \hat{Y}$.

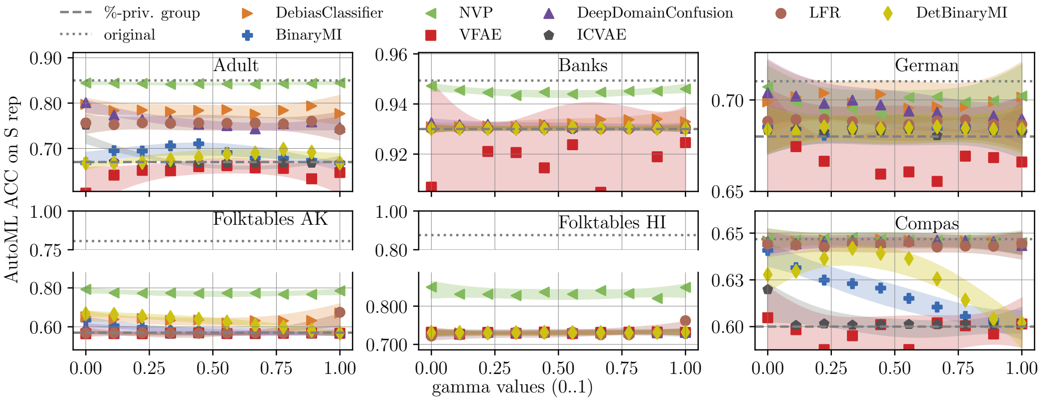

Figure 4: AutoML ACC vs. gamma results for all six dataset and eight model combinations.

5 Fair Representation Learning: The Next 10 Years

In this paper we have discussed a fundamental theoretical challenge to fair representation learning and experimentally analyzed its relevance to several methodologies proposed in the first ten years of this field. To ensure that FRL develops into an influential methodology and achieves real-world impact, we put forward the following recommendations for further research.

- **Clarify information reduction strategy.** As Theorem 1 shows, many common assumptions in deep neural network learning (deterministic representations, injective activation functions) lead to serious theoretical FRL challenges. Future FRL research proposing new methodologies should discuss these fundamental information-theoretic results and clarify how the mutual information $I(T^i; S)$ may be actually reduced. Models that are currently understood to be information-reducing include stochastic [32] or highly quantized [6, 15] representations.
- **Severe testing across both FRL frames.** As highlighted in Section 4.4, it may happen that FRL methods will display fairness in terms of allocation but will not be in terms of learning invariant representations. Both evaluation frames are critical, especially since FRL methods are overall opaque and other simpler methods are provably optimal in term of fair allocation [27]. To facilitate future severe testing in FRL we release `EvalFRL`, our extensible experimentation library `https://anonymous.4open.science/r/EvalFRL`.
- **Testing on datasets with known distributions.** One way to obtain rigorous baselines for information removal is to obtain and test on datasets for which the distributions are known a priori. To avoid testing on simplified toy datasets, recent developments in data generators for

biased data [8] should be considered. We elaborate on other sources for complex real-world data with known distributions and show initial results in Appendix D.

References

[1] R. Aaij et al. "Search for the lepton flavour violating decay $\tau \to \mu^- \mu^+ \mu^-$". In: *Journal of High Energy Physics* 2015.2 (2015). ISSN: 1029-8479. DOI: 10.1007/jhep02(2015)121. URL: http://dx.doi.org/10.1007/JHEP02(2015)121.

[2] Alessandro Achille and Stefano Soatto. "Emergence of invariance and disentanglement in deep representations". In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 1947–1980.

[3] Sajid Alam et al. *Kedro*. Version 0.19.5. Apr. 2024. URL: https://github.com/kedro-org/kedro.

[4] Rana Ali Amjad and Bernhard C Geiger. "Learning representations for neural network-based classification using the information bottleneck principle". In: *IEEE transactions on pattern analysis and machine intelligence* 42.9 (2019), pp. 2225–2239.

[5] Julia Angwin et al. *Machine Bias*. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing. Accessed: 2023-04-24. 2016.

[6] Mislav Balunovic, Anian Ruoss, and Martin Vechev. "Fair normalizing flows". In: *International Conference on Learning Representations*. 2021.

[7] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. http://www.fairmlbook.org. Accessed: 2023-04-24. 2019.

[8] Joachim Baumann et al. "Bias on Demand: A Modelling Framework That Generates Synthetic Data With Bias". In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 1002–1013. URL: https://doi.org/10.1145/3593013.3594058.

[9] Remco R. Bouckaert and Eibe Frank. "Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Honghua Dai, Ramakrishnan Srikant, and Chengqi Zhang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 3–12. ISBN: 978-3-540-24775-3.

[10] M. Cerrato et al. "Fair pairwise learning to rank". In: *DSAA* (2020).

[11] Mattia Cerrato, Roberto Esposito, and Laura Li Puma. "Constraining Deep Representations with a Noise Module for Fair Classification". In: *ACM SAC*. 2020.

[12] Mattia Cerrato et al. "Fair Group-Shared Representations with Normalizing Flows". In: *CoRR* abs/2201.06336 (2022). arXiv: 2201.06336. URL: https://arxiv.org/abs/2201.06336.

[13] Mattia Cerrato et al. *Fair Interpretable Representation Learning with Correction Vectors*. 2022.

[14] Mattia Cerrato et al. "Fair Interpretable Representation Learning with Correction Vectors". In: *CoRR* abs/2202.03078 (2022). arXiv: 2202.03078. URL: https://arxiv.org/abs/2202.03078.

[15] Mattia Cerrato et al. "Invariant representations with stochastically quantized neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 6. 2023, pp. 6962–6970.

[16] Paweł Czyż et al. "Beyond normal: On the evaluation of mutual information estimators". In: *Advances in Neural Information Processing Systems* 36 (2024).

[17] Jeffrey Dastin. *Amazon scraps secret AI recruiting tool that showed bias against women*. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G. Accessed: 2024-05-08. 2018.

[18] Frances Ding et al. "Retiring Adult: New Datasets for Fair Machine Learning". In: *Advances in Neural Information Processing Systems* 34 (2021).

[19] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using Real NVP". In: *ICLR* (2017).

[20] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. http://archive.ics.uci.edu/ml. Accessed: 2023-04-24. 2017.

[21] Batya Friedman and Helen Nissenbaum. "Bias in computer systems". In: *ACM Transactions on Information Systems (TOIS)* 14.3 (1996), pp. 330–347.

[22] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, et al. "Domain-adversarial training of neural networks". In: *JMLR* 17 (2016).

[23] Ziv Goldfeld and Yury Polyanskiy. "The information bottleneck problem and its applications in machine learning". In: *IEEE Journal on Selected Areas in Information Theory* 1.1 (2020), pp. 19–38.

[24] Arthur Gretton et al. "A kernel two-sample test". In: *The Journal of Machine Learning Research* 13.3 (2012), pp. 723–773.

[25] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. "Why do tree-based models still outperform deep learning on typical tabular data?" In: *Advances in neural information processing systems* 35 (2022), pp. 507–520.

[26] Aditya Grover et al. *AlignFlow: Cycle Consistent Learning from Multiple Domains via Normalizing Flows*. 2019. arXiv: 1905.12892.

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

[28] Kaiming He et al. "Deep residual learning for image recognition". In: *CVPR* (2016).

[29] Hans Hofmann. *Statlog (German Credit Data)*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NC77. 1994.

[30] Nikola Jovanović et al. "Fare: Provably fair representation learning with practical certificates". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 15401–15420.

[31] kaggle.com. *Flavours of Physics: Finding $\tau \to \mu\mu\mu$*. https://www.kaggle.com/c/flavours-of-physics. [Online; accessed 22-May-2024]. 2015.

[32] Christos Louizos et al. "The Variational Fair Autoencoder". In: *ICLR* (2016).

[33] Gilles Louppe, Michael Kagan, and Kyle Cranmer. "Learning to pivot with adversarial networks". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. 2017, pp. 982–991.

[34] David Madras et al. *Learning Adversarially Fair and Transferable Representations*. 2018. arXiv: 1802.06309.

[35] David Madras et al. "Learning Adversarially Fair and Transferable Representations". In: *CoRR* abs/1802.06309 (2018). arXiv: 1802.06309. URL: http://arxiv.org/abs/1802.06309.

[36] Gianclaudio Malgieri. "The Concept of Fairness in the GDPR: A Linguistic and Contextual Interpretation". In: *FAT** (2020).

[37] Daniel McNamara, Cheng Soon Ong, and Robert C Williamson. *Provably fair representations*. 2017. arXiv: 1710.04394.

[38] Sérgio Moro, Paulo Cortez, and Paulo Rita. "A Data-Driven Approach to Predict the Success of Bank Telemarketing". In: *Decision Support Systems* 62 (June 2014).

[39] Daniel Moyer et al. "Invariant representations without adversarial training". In: *NeurIPS* (2018).

[40] Claude Nadeau and Yoshua Bengio. "Inference for the generalization error". In: *Machine learning* 52.3 (2003), pp. 239–281.

[41] Harikrishna Narasimhan et al. "Pairwise Fairness for Ranking and Regression". In: *AAAI* (2020).

[42] Aleksandra Płońska and Piotr Płoński. *MLJAR: State-of-the-art Automated Machine Learning Framework for Tabular Data. Version 0.10.3*. Łapy, Poland, 2021. URL: https://github.com/mljar/mljar-supervised.

[43] Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Book draft. Cambridge University Press, 2023.

[44] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature machine intelligence* 1.5 (2019), pp. 206–215.

[45] Cynthia Rudin, Caroline Wang, and Beau Coker. "The Age of Secrecy and Unfairness in Recidivism Prediction". In: *Harvard Data Science Review* 2.1 (Mar. 31, 2020). https://hdsr.mitpress.mit.edu/pub/7z10o269. DOI: 10.1162/99608f92.6ed64b30.

[46] Andrew M Saxe et al. "On the information bottleneck theory of deep learning". In: *Journal of Statistical Mechanics: Theory and Experiment* 2019.12 (2019), p. 124020.

[47] Shubham Sharma et al. "FaiR-N: Fair and Robust Neural Networks for Structured Data". In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. Association for Computing Machinery, 2021.

[48] Ravid Shwartz-Ziv and Naftali Tishby. "Opening the Black Box of Deep Neural Networks via Information". In: *CoRR* abs/1703.00810 (2017). arXiv: `1703.00810`. URL: `http://arxiv.org/abs/1703.00810`.

[49] Naftali Tishby, Fernando C. Pereira, and William Bialek. *The information bottleneck method*. 2000. arXiv: `physics/0004057 [physics.data-an]`.

[50] Naftali Tishby and Noga Zaslavsky. "Deep learning and the information bottleneck principle". In: *2015 IEEE Information Theory Workshop (ITW)*. IEEE. 2015, pp. 1–5.

[51] Eric Tzeng et al. *Deep domain confusion: Maximizing for domain invariance*. 2014. arXiv: `1412.3474`.

[52] Qizhe Xie et al. "Controllable Invariance through Adversarial Feature Learning". In: *NeurIPS* (2017).

[53] Ke Yang and Julia Stoyanovich. "Measuring Fairness in Ranked Outputs". In: *SSDBM* (2017).

[54] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, et al. "Fairness beyond disparate treatment & disparate impact". In: *WWW* (2017).

[55] Muhammad Bilal Zafar et al. "Fairness Constraints: A Flexible Approach for Fair Classification". In: *Journal of Machine Learning Research* 20.75 (2019), pp. 1–42. URL: `http://jmlr.org/papers/v20/18-262.html`.

[56] Meike Zehlike and Carlos Castillo. "Reducing disparate exposure in ranking: A learning to rank approach". In: *WWW* (2019).

[57] Rich Zemel et al. "Learning fair representations". In: *ICML* (2013).

A Detailed Experimental Setup

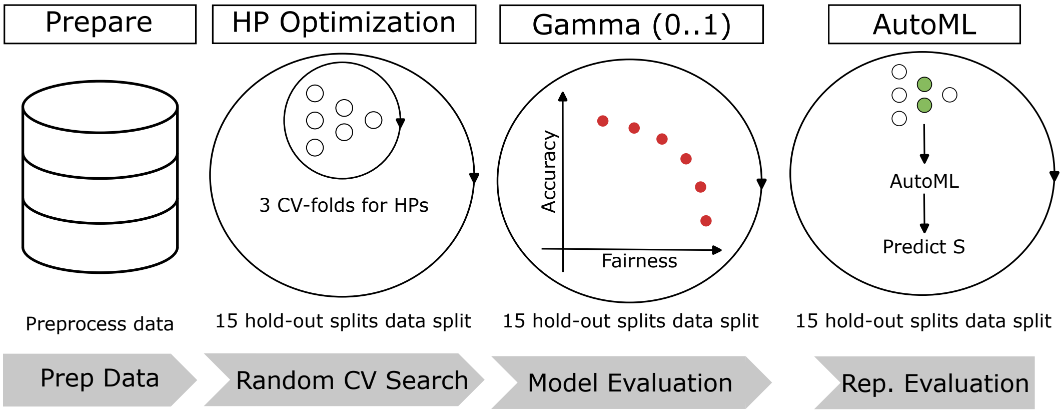

Figure 5: A graphical summary of EvalFRL, our experimentation library for FRL algorithms. It shows an overview of the Kedro pipeline used for preprocessing, hyperparameter optimization, γ experiments and AutoML evaluation.

We show a graphical summary of EvalFRL in Figure 5. Our experimentation library employs separate Kedro pipelines to perform preprocessing, hyperparameter optimization, tradeoff analysis and the final evaluation for each model/dataset combination. In detail, EvalFRL performs the following steps for every available dataset and model:

1. Data preprocessing via encoding (discrete features) and normalization (continuous features).
2. Employing a 15-by-3 hold-out/CV split [40], tune the hyperparameters via random search so to maximise the AUC w.r.t. the classification task for each dataset. We test 100 different hyperparameter combinations.
3. Utilizing the found hyperparameter combinations, models are then re-trained with gamma values ranging from 0 to 1. The trained models are used generate fair representations Z by computing the activations of the second-to-last layer $Z^{L-1} = \phi^{L-1}(x)$.
4. Use AutoML to predict the sensitive attribute S from the representations Z^{L-1}.

This procedure repeated across 8 models and 6 datasets implies a total of 335.000 model fits. The tested hyperparameter ranges for each methodology and dataset are available at https://anonymous.4open.science/r/EvalFRL/runs/experiment.yml. The best hyperparameters found for each outer fold are available in the experimental metadata https://drive.google.com/drive/folders/1koZd8cgBJMVGuH3uRqvpTFEUJo0Sd23q?usp=sharing. We refer to the README.md file contained therein for instructions.

A.1 Models

BinaryMI The BinaryMI model leverages stochastically activated binary layer(s) to compute the mutual information between these layers and the sensitive attributes. By treating neurons as bernoulli random variables, this approach directly calculates the mutual information, which is then used as a regularization factor during gradient descent to ensure fairness in the learned representations [15]. n our experiments, we also utilize this model to determine whether the fairness of the representations is due to the Bernoulli layer or the quantization process. Both factors, as discussed in Section 3, may be employed to circumvent Theorem 1. To achieve this, we remove the Bernoulli sampling from the training phase and use a quantized sigmoid activation instead. We refer to this deterministic version of the BinaryMI model as DetBinaryMI.

DebiasClassifier The DebiasClassifier leverages adversarial training to create fair representations by integrating an adversarial network that discourages the encoding of protected attributes. We implemented this method via gradient reversal as proposed by Ganin et al. [22] in domain adaptation and then employed in fairness by Xie et al. [52] and Madras et al. [34].

NVP Out of several available FRL normalizing flow methodologies, we test a fair normalizing flow model [14] that leverages two RealNVP [19] models. We choose this method as it does not require the sensitive attribute at test time (differently to e.g. [6]) and as it does seek to break the bijective relationship inherent to normalizing flows (e.g. present in AlignFlow [26]) by "funnelling" information about sensitive attribute into one latent variable, which is then set to zero when entering the second RealNVP.

VFAE The Fair Variational Autoencoder (VFAE) is a methodology proposed by Louizos et al. [32] that leverages variational autoencoders to learn fair representations. The fairness of the representations is obtained via architectural constraints and a loss term based on the Maximum Mean Discrepancy (MMD) [24].

ICVAE The Independent Conditional Variational Autoencoder (ICVAE) is also based on variational autoencoders [39]. Here, fairness is obtained via the well-known relationship between mutual information and the KL divergence. A probability density for $P(Z)$ is made available by employing variational autoencoders.

LFR Learning Fair Representations (LFR) was proposed by Zemel et al. [57] and poineered the field of FRL. In LFR every individual gets stochastically mapped to so-called prototypes, which are points in the same space as X. This mapping $g : X \to Z$ combined with another mapping $f : Z \to Y$ are optimized to satisfy three goals: 1. g statisfies group fairness, 2. g retains all information on X and 3. $f \circ g$ is close to the real classification. While the formulation in the original paper [57] appears to us to be discrete and stochastic, we note that its implementation in the `AIF360` library consistently returns continuous representations without any variance across different calls of the `transform(X)` function. We employed the `AIF360` implementation in our experimentation.

DeepDomainConfusion [51] was introduced by Tzeng et al. as a domain adaptation method. Similarly to VFAE [32], it employs the MMD [24] between representations of different domains as a loss function term. We instead encode domains as different values of sensitive attributes.

A.2 Dataset Information

COMPAS. This dataset (called Compas in the following), introduced by ProPublica [5], focuses on evaluating the risk of future crimes among individuals previously arrested, a system commonly used by US judges. The ground truth is whether an individual commits a crime within the following two years. The sensitive attribute is ethnicity.

Adult. This dataset, available in the UCI repository [20], pertains to determining whether an individual's annual salary exceeds $50,000. We take gender to be the sensitive attribute [32, 57].

Bank marketing. Here, the classification task involves predicting whether an individual will subscribe to a term deposit. This dataset (called Banks in the following) exhibits disparate impact and disparate mistreatment concerning age, particularly for individuals under 25 and over 65 years old. [38]

German. The German Credit dataset, contains credit data of individuals with the objective of predicting their credit risk as either high or low risk. The gender of the individuals serves as the sensitive attribute [29].

Folktables. The folktables datasets are a collection of datasets derived from US cencus data, which span multiple years and all states of the USA [18]. Although the dataset supports multiple prediction tasks, we only used the income task, in which the objective is to predict whether an individual´s income exceeds 50.000$. The sensitive attribute is the race of the individual. We picked the datasets from Alaska (AK) and Hawaii (HI) for our experiments, by comparing the performance of AutoML and logistic regression in predicting the sensitive attribute S using the features X. We observed that on AK and HI AutoML performed remarkably better than linear models, indicating a complex relationship between the features X and the sensitive attribute S. Thus, we concluded that learning invariant representations on these datasets would be a relatively hard task.

A.3 Other Fairness Metrics

Before introducing further results in fair allocation, we report here other two classical fair allocation metrics commonly employed in the literature.

Statistical Parity Difference Statistical Parity Difference (SPD) measures the difference in the probability of favorable outcomes between protected and unprotected groups. It is defined as:

$$\text{SPD} = P(\hat{Y} = 1 \mid S = 0) - P(\hat{Y} = 1 \mid S = 1) \tag{7}$$

where $\hat{Y}$ is the predicted outcome, and S is the sensitive attribute (e.g., gender, race). A value of 0 indicates perfect fairness, while values closer to -1 or 1 indicate higher disparity.

Delta Introduced by Zemel et al. [57], Delta is defined as Delta = yDiscrim − yAcc, with yAcc being the prediction accuracy

$$\text{yAcc} = 1 - \frac{1}{N}\sum_{n=1}^{N} |y_n - \hat{y}_n|. \tag{8}$$

[57] This metric indicates the relative gain in terms of fairness vs. accuracy.

B Other Results in Fair Allocation

The following plots demonstrate how different models perform in terms of accuracy and fairness across a range of gamma values. Deterministic models (pink shading) generally show higher accuracy and less variability compared to stochastic models (gray shading). Notably, datasets like Banks and German exhibit more significant changes, highlighting the importance of gamma tuning.

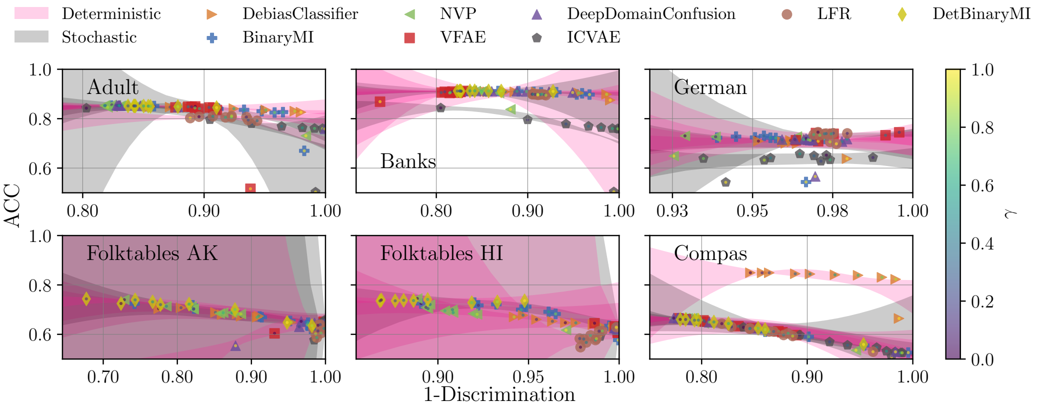

Figure 6: Accuracy vs. 1 - Discrimination tradeoff for all six dataset and eight model combinations.

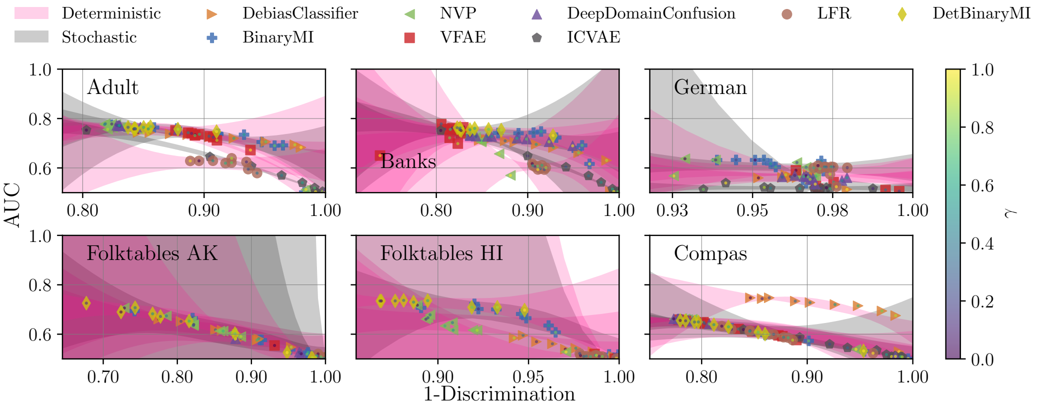

Figure 7: AUC vs. 1 - Discrimination tradeoff for all six dataset and eight model combinations.

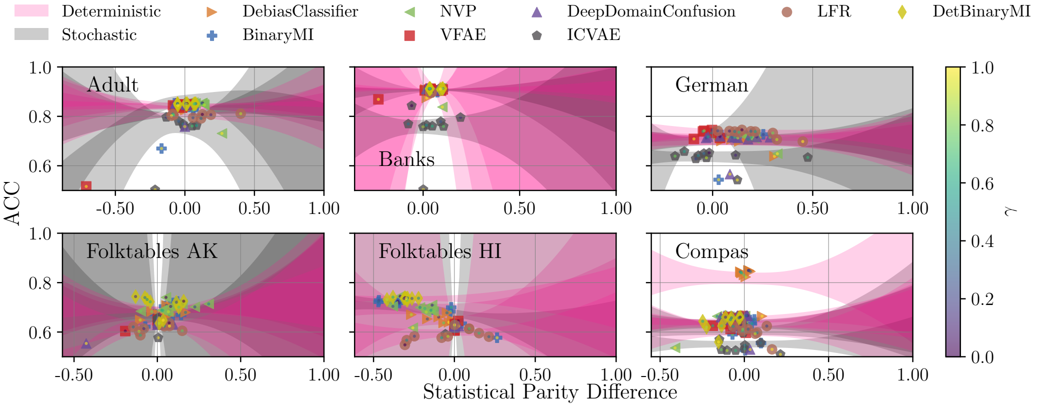

Figure 8: Accuracy vs. 1 - statistical parity difference tradeoff for all six dataset and eight model combinations.

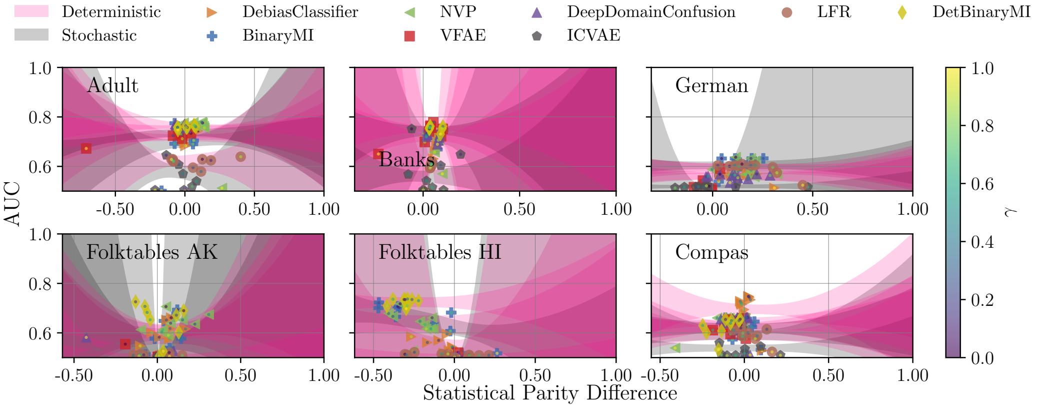

Figure 9: AUC vs. 1 - statistical parity difference tradeoff for all six dataset and eight model combinations.

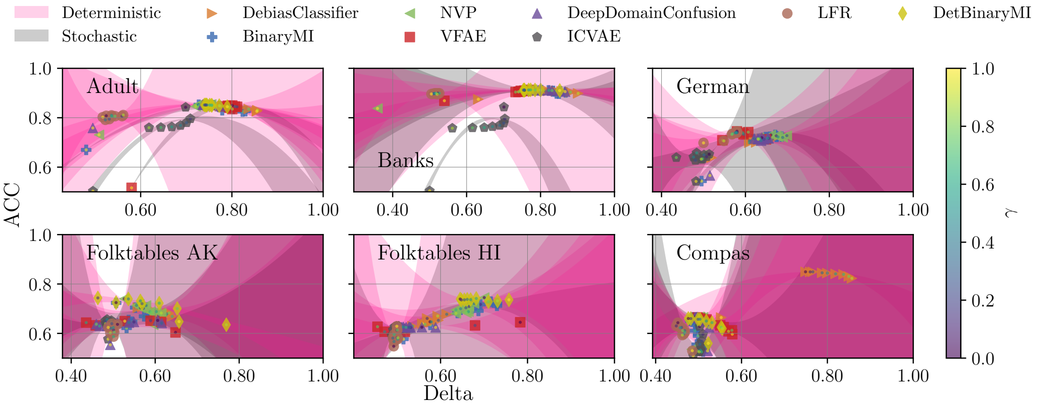

Figure 10: Accuracy vs. delta tradeoff for all six dataset and eight model combinations.

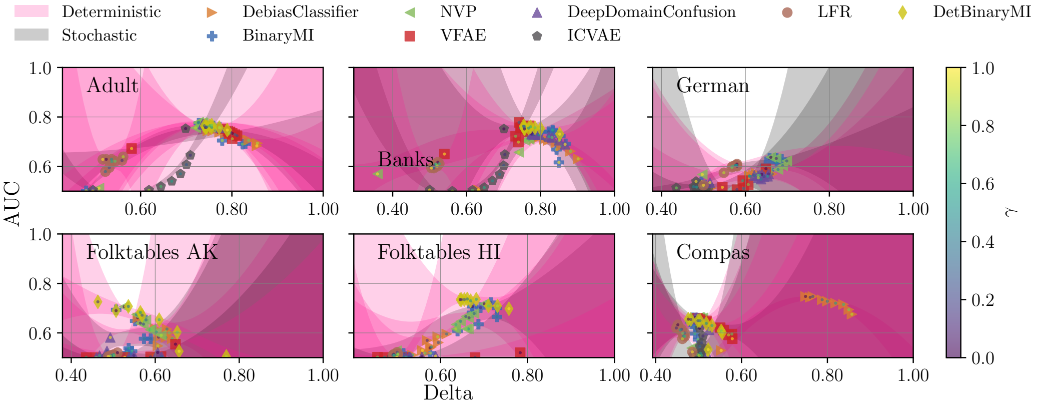

Figure 11: AUC vs. 1 - rND tradeoff for all six dataset and eight model combinations.

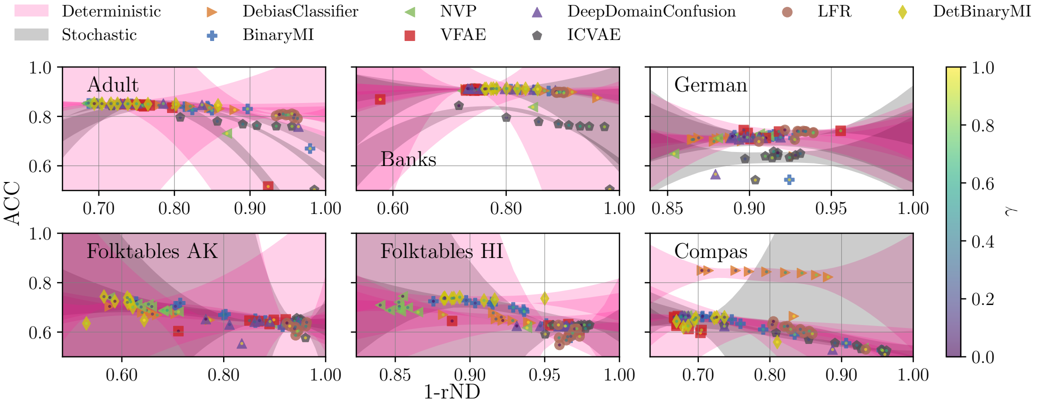

Figure 12: Accuracy vs. delta tradeoff for all six dataset and eight model combinations.

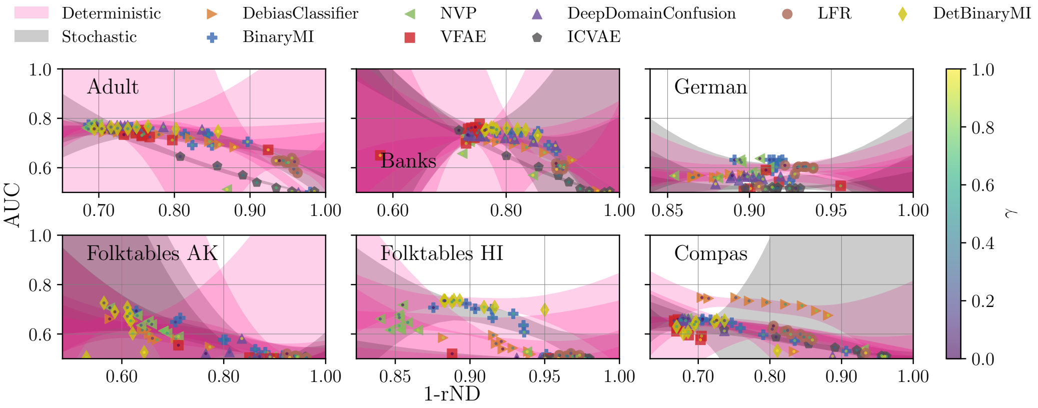

Figure 13: AUC vs. 1 - rND tradeoff for all six dataset and eight model combinations.

C ReLU Activation Tests

C.1 ReLU Model Tests

Figures 14, 15, 16, 17,18, 19, 20 and 21 show how DebiasClassifier and its ReLU variant perform across different gamma values in terms of accuracy and fairness. Generally, the accuracy remains stable with slight variations across datasets, indicating that ReLU activation does not drastically alter the fairness-accuracy trade-off. For the Compas the normal DebiasClassifier outperforms the ReLU variant significantly.

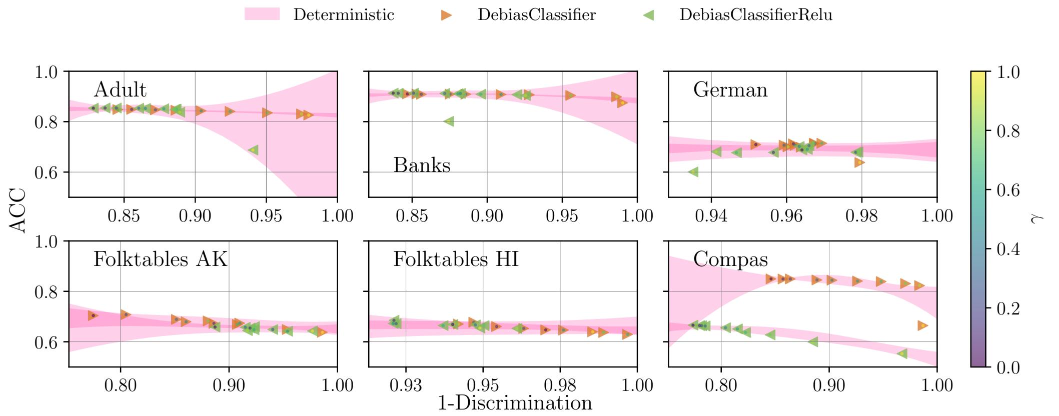

Figure 14: Accuracy vs. 1 - Discrimination tradeoff for the DebiasClassifier with $\tanh$ and ReLU activation for the six datasets.

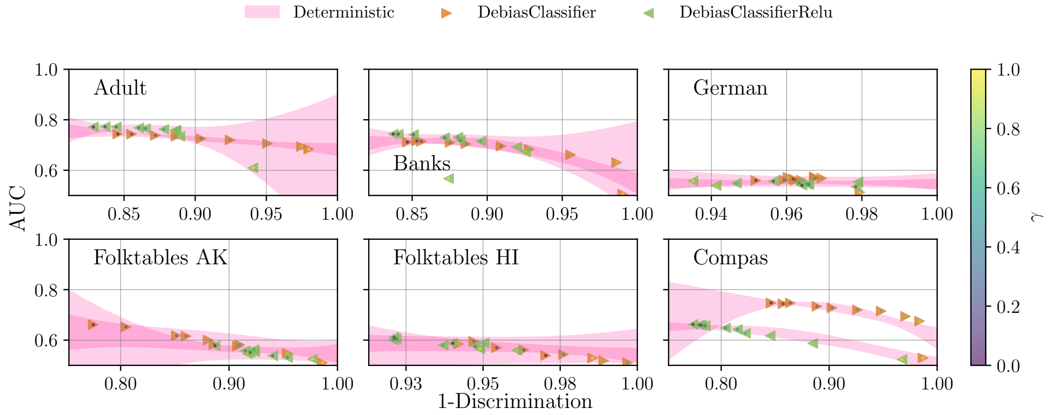

Figure 15: AUC vs. 1 - Discrimination tradeoff for the DebiasClassifier with $\tanh$ and ReLU activation for the six datasets.

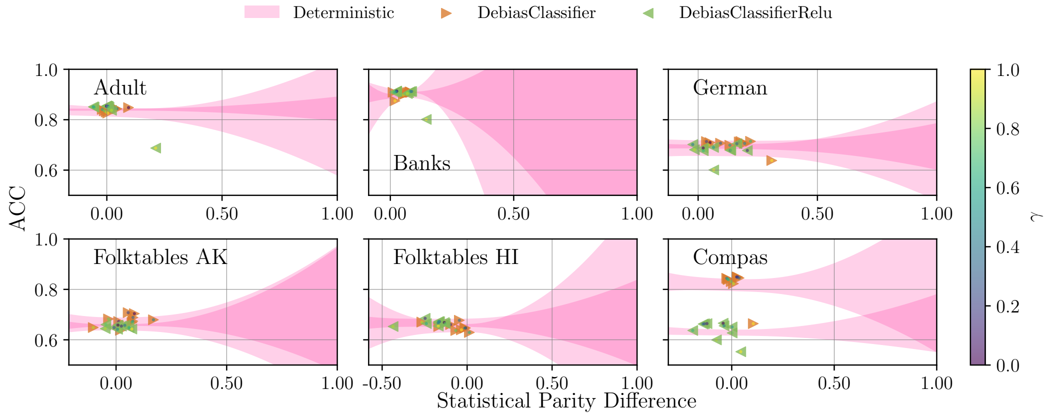

Figure 16: Accuracy vs. 1 - statistical parity difference tradeoff for the DebiasClassifier with tanh and ReLU activation for the six datasets.

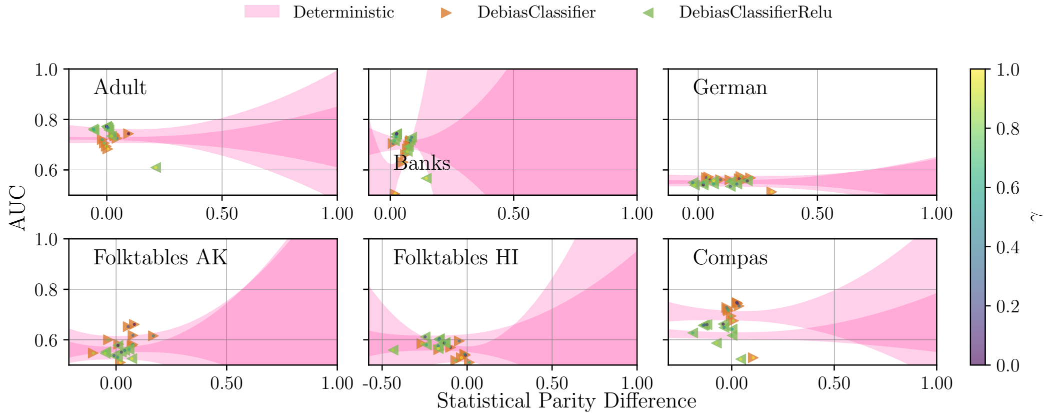

Figure 17: AUC vs. 1 - statistical parity difference tradeoff for the DebiasClassifier with tanh and ReLU activation for the six datasets.

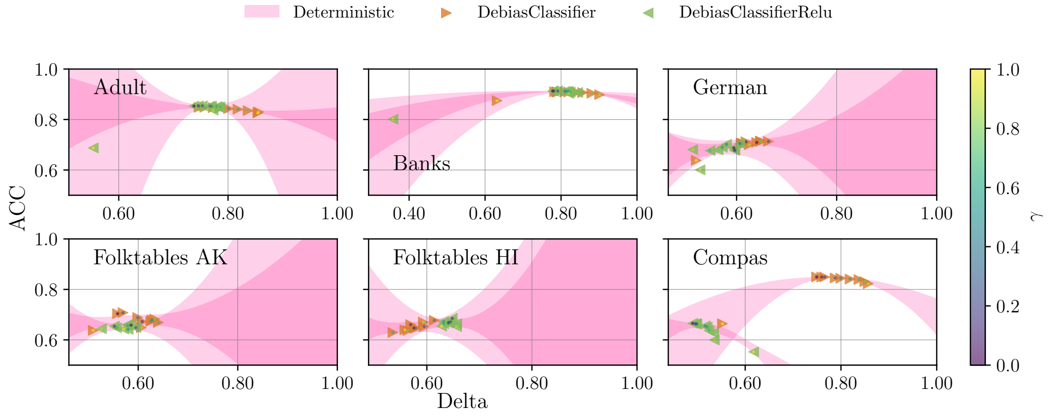

Figure 18: Accuracy vs. delta tradeoff for the DebiasClassifier with tanh and ReLU activation for the six datasets.

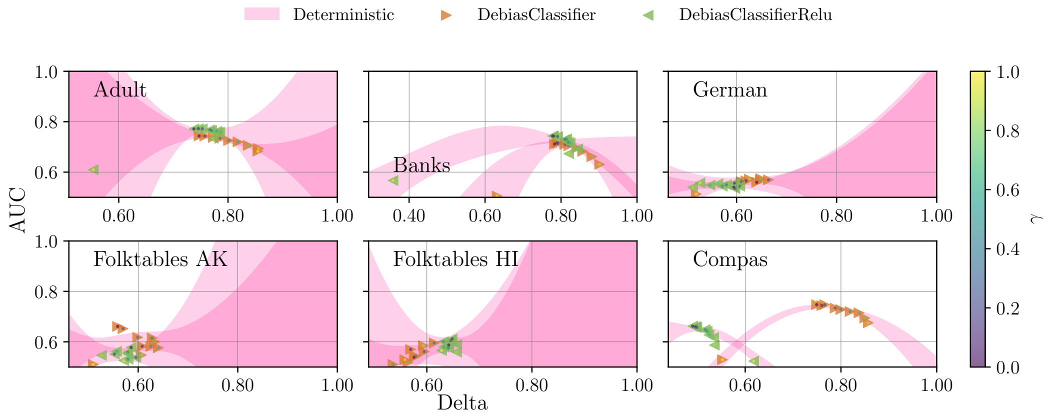

Figure 19: AUC vs. 1 - rND tradeoff for the DebiasClassifier with tanh and ReLU activation for the six datasets.

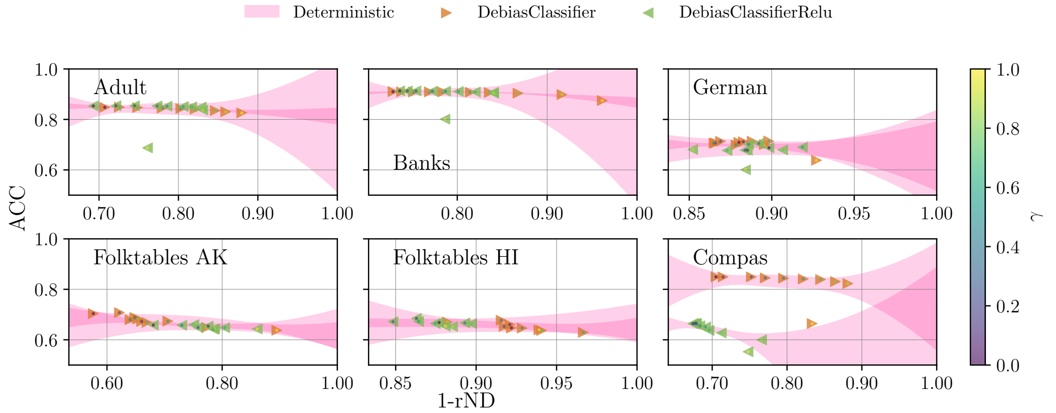

Figure 20: Accuracy vs. delta tradeoff for the DebiasClassifier with tanh and ReLU activation for the six datasets.

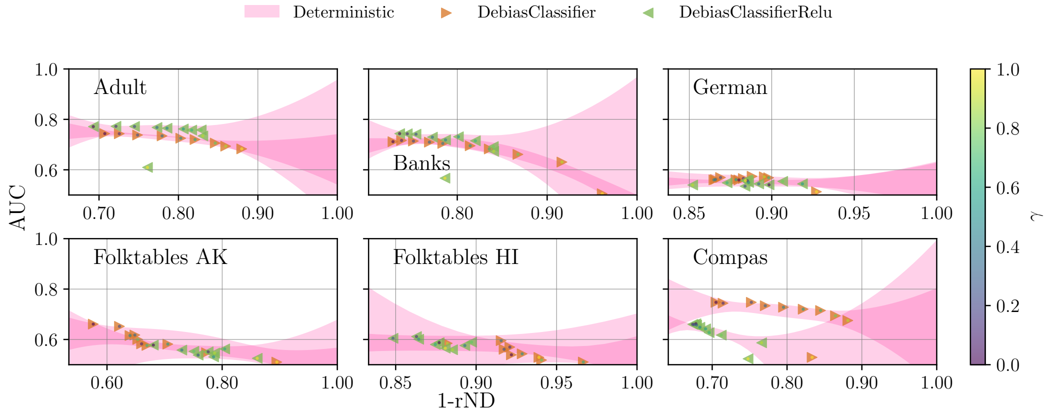

Figure 21: AUC vs. 1 - rND tradeoff for the DebiasClassifier with tanh and ReLU activation for the six datasets.

C.2 ReLU AutoML Tests

Figures 23 and 22 illustrates the performance of AutoML to predict the sensitive attribute from the representations from the DebiasClassifier and DebiasClassifierRelu across a range of γ values. The results indicate that the AutoML accuracy and AUC for both representations maintain relatively stable results over different gamma values, and seems to be relatively constant across all considered γ values.

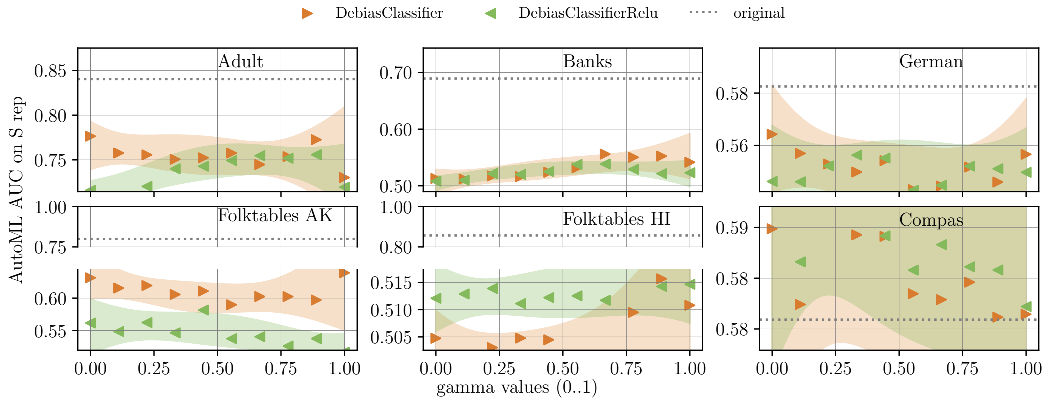

Figure 22: AutoML AUC of predicting S versus gamma values on on the representations from DebiasClassifier and DebiasClassifierRelu across multiple datasets. The dotted lines represent the performance on the original data.

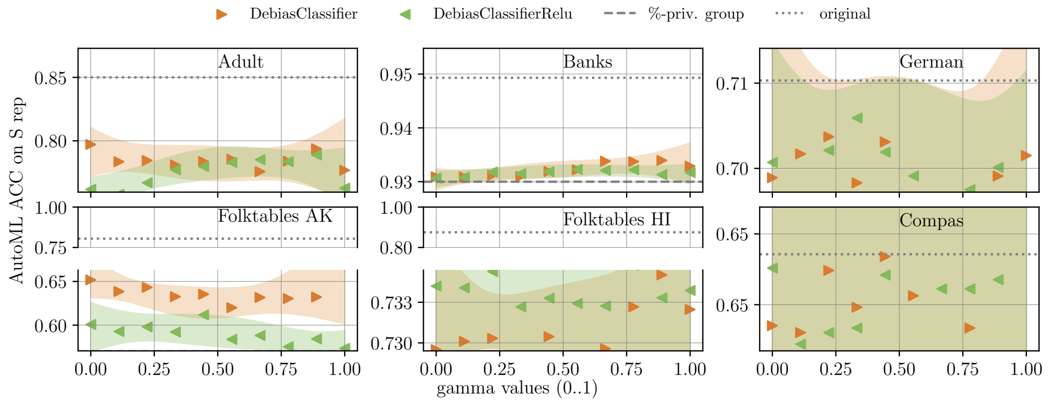

Figure 23: AutoML AUC of predicting S versus gamma values on on the representations from DebiasClassifier and DebiasClassifierRelu across multiple datasets. The dashed and dotted lines represent the percentage of the privileged group and the performance on the original data, respectively.

D Particle Physics Data

To rigorously assess whether a machine learning model has effectively removed unwanted information, it is crucial to comprehend the underlying data generation process. However, in many real-world scenarios where fairness is a concern, data is often sourced from human interactions, making it challenging to fully understand the origins of bias.

One approach to address this challenge while maintaining the complexity of real-world datasets is to explore domains where data creation and collection processes are meticulously documented and

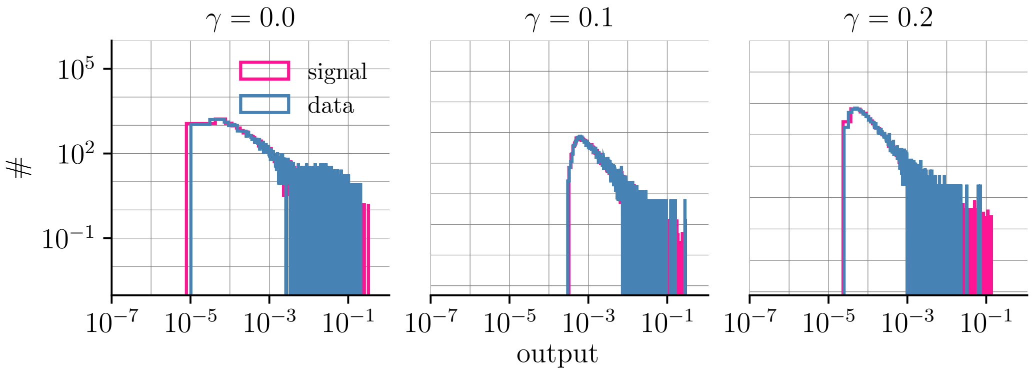

Figure 24: Comparison of signal and background data at various gamma (γ) levels using the BinaryMI model on the Kaggle 'Flavours of Physics' dataset. The plot showcases how different gamma values ($\gamma = 0.0, 0.1, 0.2$) impact the distribution of the output variable, demonstrating the model's ability to be invariant between signal and background events.

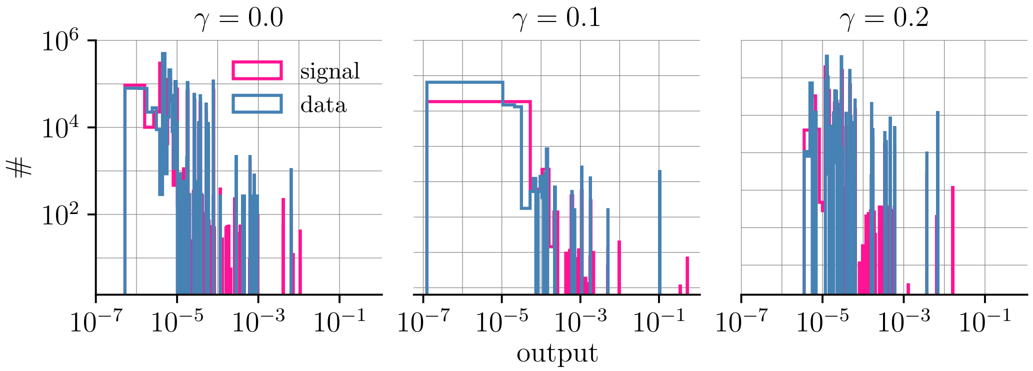

Figure 25: Comparison of signal and background data at various gamma (γ) levels using the DebiasClassifier model on the Kaggle 'Flavours of Physics' dataset. The plot showcases how different gamma values ($\gamma = 0.0, 0.1, 0.2$) impact the distribution of the output variable.

understood with a high degree of precision. One such domain is particle physics. Previous work tested adversarial FRL [22] to predict W-jets events, being at the same time invariant to pileup (i.e. noise) [33]. We note however that the original data was not published, to the best of our knowledge. We therefore suggest to use the data from the "Flavour of Physics" Kaggle challenge [31]. The task is to identify $\tau \rightarrow \mu\mu\mu$ decay events in high-energy physics data [1] and improve the detection of this rare particle decay process using machine learning techniques. Participants are provided with datasets containing particle collision data and are tasked with building models to distinguish between signal and background events. The FRL/invariance framing is provided by an agreement test which quantifies the level of invariance obtained by the model across predictions for simulation and real (i.e. detector) data. This test is necessary as for an unknown signal event there only simulation data will be available; while for the background event both simulation and real detector data are given.

To mitigate this issue most particle physics experiments have so-called "control" areas where the real- and simulation-data distributions are well-understood theoretically. These control areas are therefore employed in an agreement test.

When testing different FRL methodologies on this complex physical data, we observe phenomena which are consistent with our findings in Sections 3 and 4. In Figure 24 the output of the BinaryMI model – a stochastic model – over the control area is shown for the decay $Ds \rightarrow \varphi\Pi$ which has the same signature as the $\tau \rightarrow \mu\mu\mu$ decay [7]. When increasing the γ the two output distributions become increasingly similar, indicating that the model is invariant to data and signal.

In contrast to the BinaryMI model, Figure 25 shows the output of the DebiasClassifier. Besides the discrete output values the DebiasClassifier shows greater separation between signal and background data at higher γ values, which implies less fairness since more separation indicates a stronger bias towards certain data.

E Computing Infrastructure

All experiments were run on CPUs without involving GPUs. The system used consisted of four PCs, each equipped with 190 GB of RAM and an AMD EPYC 9254 24-Core Processor. The total runtime for all experiments was approximately one week. The primary limitation was not computational power but the required RAM to fit all models simultaneously. For reproducing the experiments, we recommend running one model over all six datasets per PC to manage memory constraints effectively.

[7] The code for this initial evaluation can be found at `https://anonymous.4open.science/r/EvalFRL/notebooks/distribution_shift.ipynb`.