Learning Fair Representations through Uniformly Distributed Sensitive Attributes

Patrik Joslin Kenfack[*], Adín Ramírez Rivera[†], Adil Mehmood Khan[*‡], and Manuel Mazzara[§]
[*]Machine Learning and Knowledge Representation Lab, Innopolis University
[§]Institute of Software Development and Engineering, Innopolis University
Innopolis 420500, Russia
Emails: p.kenfack@innopolis.university, {a.khan, m.mazzara}@innopolis.ru
[†]Digital Signal Processing and Image Analysis (DSB) group, Department of Informatics, University of Oslo
N-0373 Oslo, Norway
Email: adinr@uio.no
[‡]School of Computer Science, University of Hull
HU67RX, Hull, UK

***Abstract*—Machine Learning (ML) models trained on biased data can reproduce and even amplify these biases. Since such models are deployed to make decisions that can affect people's lives, ensuring their fairness is critical. One approach to mitigate possible unfairness of ML models is to map the input data into a less-biased new space by means of training the model on fair representations. Several methods based on adversarial learning have been proposed to learn fair representation by fooling an adversary in predicting the sensitive attribute (e.g., gender or race). However, adversarial-based learning can be too difficult to optimize in practice; besides, it penalizes the utility of the representation. Hence, in this research effort we train bias-free representations from the input data by inducing a uniform distribution over the sensitive attributes in the latent space. In particular, we propose a probabilistic framework that learns these representations by enforcing the correct reconstruction of the original data, plus the prediction of the attributes of interest while eliminating the possibility of predicting the sensitive ones. Our method leverages the inability of Deep Neural Networks (DNNs) to generalize when trained on a noisy label space to regularize the latent space. We use a network head that predicts a noisy version of the sensitive attributes in order to increase the uncertainty of their predictions at test time. Our experiments in two datasets demonstrated that the proposed model significantly improves fairness while maintaining the prediction accuracy of downstream tasks.**

***Index Terms*—Fairness, Fair representation, Bias, Decision making**

I. Introduction

The growing deployment of Machine Learning (ML) systems in business and government has shown that these systems can learn, perpetuate, and even amplify biases that that society is combating in the real world [1, 2]. The decisions provided by ML models can have a profound effect on the course of people's lives e.g., hiring, detention, college admissions Therefore, it is crucial to ensure that such decisions are sound and fair. Fairness in ML has received a lot attention during the past years.

A number of definitions of *fairness* have been proposed to quantify the unfairness of ML models. These fairness definitions can be categorized into three main categories. *Group fairness* [3, 4, 5] evaluates the model's performance within different groups; based on a given sensitive attribute (e.g., gender, race, age), it requires the model to treated groups equally. *Individual fairness* [5] requires that similar individuals with respect to given task should receive a similar outcome. *counterfactual fairness* [6] requires that the decisions provided by the model remain the same if the sensitive attribute were changed (e.g., in the case of a loan approval system, what the outcome would have been if a female applicant had been born a male). Our study focuses on achieving group-based fairness notion such as *statistical parity*, *equalize odds*, and *equal opportunity*, with an emphasis on statistical parity.

One way to mitigate unfairness is to map the input data into a less-biased new space by means of learning *fair representations*. The new representation of the data is therefore likely to produce more fair outcomes when used for any downstream tasks such as classification. Several approaches based on adversarial learning have been proposed to learn a fair representation [7, 8, 9, 10] by removing all dependencies on the sensitive attributes from the data while preserving as much of the other information as possible.

In the adversarial training, an adversary is trained to predict the sensitive attribute from the latent space yielded by a generator, while the generator is trained to fool the adversary in predicting the sensitive attribute. However, such an adversarial setup requires that any adversary cannot predict the protected attribute; beside, such adversaries can be difficult to optimize in practice [7, 11, 12]. Moreover, these approaches always lead to a degradation of the predictor accuracy, in particular, Moyer et al. [11] showed that adversarial training is unnecessary and sometimes counter-productive. In addition, reducing the tradeoff between fairness and accuracy is an active research question within the fair ML community, which essentially implies the task to provide models with high accuracy with as little bias as possible. Our study addresses these issues by proposing an alternative and straightforward approach for learning fair representations without an adversary while maintaining high utility on downstream tasks.

We introduce a new regularizing objective function that yields a representation that maintains the utility of predictions and mitigates bias comparably better than the state-of-the-art approaches. In essence, our proposed objective is not only to make the prediction of the sensitive attribute impossible to predict by an adversary, but instead to increase the uncertainty of predicting the sensitive attribute from the latent space, i.e., make it almost uniform (close to 0.5). To induce the uniformity over the sensitive attribute in the latent space, we use a neural network that takes the latent code as input and predict a noisy version of the sensitive attribute, i.e., a defined percentage (e.g., 50%) of individuals are randomly switched between different groups considered. The intuition here is that similarly to a counterfactual world, the target variable (here the sensitive attribute) of few samples is flipped in a way to confuse the inherent characteristics of each individual, i.e., increase the uncertainty of predicting the sensitive attribute of individuals from the latent space and thus make them unreliable. Model training on noisy labels has been shown to be a source of uncertainty for DNNs [13, 14]. Furthermore, neural networks can learn and fit data with noisy or random labels but fail to generalize at testing time [15], i.e., although the model achieves small training loss, the predictions on unseen data are almost random (unreliable). By leveraging this weakness of DNNs, our regularizer head trained on a noisy version of the sensitive attribute enforces the latent space to provide unreliable predictions (uniform) when used to predict the sensitive attributes.

We showed theoretically that our process minimizes an upper bound on the Kullback-Leibler divergence between the sensitive attributes predicted from the latent space, $p(s \mid z)$, and the uniform distribution. Because our analytical guarantees do not hold for the general conditional $p(s \mid z)$, we cannot conclude that our approach will be robust against all future adversaries. We evaluate an adaptive adversary in our work, but we hope that future work will build on our insights to formalize the guarantee provided in a more general case or disprove this guarantee empirically. We experimentally showed that our approach is effective for enforcing the independence between the learned representation and the sensitive attributes. As such, the downstream classification tasks using our representation provided better fairness performance in terms of statistical parity along with a positive impact on equalized odds and equal opportunity. Based on the obtained results, we claim this objective is easy to train, the approach yields higher fairness performance and does not penalize the utility of the model much.

II. Related Work

The objective of fair representation learning is to find a function to map the input data into a fair space, i.e., a space where the protected and non-protected attributes are indistinguishable. Zemel et al. [16] were the first to propose learning fair intermediate representations: similar to k-means, their method involves finding k prototypes in the same space as the input data. Each sample is assigned to the closest prototype while adding a constraint in the optimization objective to satisfy fairness and classification performance. Louizos et al. [17] proposed the Variational Fair Autoencoder, an adaptation of the variational autoencoder to learn a mapping function that enforces the independence of the sensitive attribute and the latent space. The authors treated the sensitive attribute as nuisance variable and factored it out in the latent space. As opposed to Louizos et al.'s [17] work, our fair mapping function focuses not only on encouraging independence to the sensitive attribute, but also on maintaining the high utility of the representation on downstream tasks such as classification.

More closely related to our work, Edwards and Storkey [7] proposed an adversarial approach to enforce statistical parity, which involves three main components: the autoencoder that yields the latent space (generator); the adversary trained to predict the sensitive attribute from the latent space; and a classifier trained to maximize the utility of the latent representation with respect to the class label. The autoencoder and classifier try to fool the adversary in predicting the sensitive attribute from the latent space while the adversary tries to maximize its accuracy of prediction. Madras et al. [8] extended the previous work and proposed new adversarial objectives that yield transferable fair representation and also considered other fairness notions like equal opportunity and equalized odds. Kenfack et al. [9] showed that applying the adversary at multiple levels of representations tightens the fairness bound of the learned representation. Feng et al. [12] used an adversary that minimizes the Wasserstein Distance between the distribution of the protected and non-protected groups. Moyer et al. [11] learn fair representations by minimizing the mutual information between the latent space and sensitive attribute while maximizing the mutual information between the latent space and loss of the task at hand.

These previous works mainly focus on improving fairness and have shown a significant drop in the accuracy of predictions compared to the unconstrained models. Our goal is to reduce the fairness/accuracy tradeoff of the learned representation, i.e., maintain a high accuracy of prediction on downstream tasks while improving fairness. We propose a generalized probabilistic model for fair representation learning and a relaxation of the adversary objective that appears to be easier to train and yields a better fairness/accuracy tradeoff than previous work. As demonstrated by Dutta et al. [18], it is possible to find data distribution such that optimal fairness and accuracy are achieved simultaneously.

III. Method

In order to learn fair representations, we propose a probabilistic hierarchical model that learns representations such that it is (ideally) impossible to use them for predicting the sensitive attributes thus achieving fairness with respect to these attributes. First, we detail the probabilistic model for a single level of hierarchy. Then, we discuss how to expand this model to several levels within a hierarchical model.

A. Probabilistic Model

Let the user data be the triplet (x, s, y) containing both nonsensitive information x and sensitive one s. Our task is to predict the dependent variable y. We introduce a latent

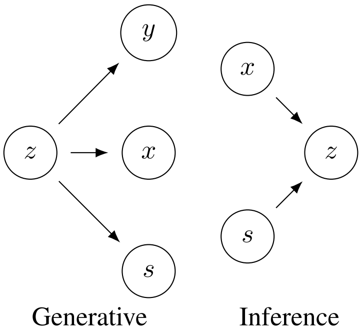

Fig. 1. Generative and inference models for our model's variables. We assume a fair generation process where the class y and features x depend only on the latent variable z. For the inference of the latent variables z, we rely on x and s, while obfuscating s.

variable z that models the user information while taking care of obfuscating the sensitive information s.

We assume a generative model (Fig. 1)

$$p(x, s, y) = \int p(y \mid z)p(x \mid z)p(s \mid z)p(z)\,\mathrm{d}z, \tag{1}$$

and we are interested in maximizing its log-probability, i.e., $\max \log p(x, s, y)$. We assume the inference model to be $q(x, s, z) = q(z|x, s)q(x, s)$ (see Fig. 1). By using a variational inference approach, we obtain the evidence lower bound of the log likelihood, that is,

$$\log p(x, s, y) \geq \mathop{\mathbb{E}}_{q(z \mid x,s)} \Bigg[\log p(y \mid z) + \log p(s \mid z) + \log p(x \mid z) + \log \frac{p(z)}{q(z \mid x, s)}\Bigg]. \tag{2}$$

This model reveals that we need to maximize the likelihood of the class prediction, $p(y \mid z)$, as well as the sensitive data prediction, $p(s \mid z)$. Simultaneously, we need to guarantee the generation of the original data, $p(x \mid z)$. Moreover, we need to minimize the Kullback-Leibler divergence of the prior and our proposed encoder, $\mathbb{E}_{q(z \mid x,s)} \log q(z \mid x, s)/p(z)$.

For our objective, we simplify this model. First, we approximate the expectation over the latent $q(z \mid x)$ by using sampling. Besides, in its simplest form, we use a single sample. Moreover, we do not work with the divergence but use implicit distributions instead. Our simplified optimization problem is

$$\max \log p(y \mid z) + \log p(s \mid z) + \log p(x \mid z), \tag{3}$$

where the sampling $z \sim q(z \mid x)$ is replaced with a deterministic function that substitutes the distribution (implemented through a neural network). However, this problem yields unfair representations since it maximizes the log-likelihood of the sensitive attribute as well. To solve this problem, we propose to make the sensitive distribution $p(s \mid x)$ close to a uniform distribution instead. The idea is that, instead of maximizing the prediction of the sensitive attribute, we make the prediction unreliable (thus, close to a uniform distribution). For example, if the sensitive attribute is binary, instead of maximizing or minimizing the likelihood (which makes it predictable), we try to maintain the likelihood close to 0.5.

Algorithm 1 Fair representation learning.

```
Require: $t \in [0, 1]$ a flipping probability (default 0.5), $N$ training epochs,
  $K$ levels, and $\Theta = \{\theta_i\}_{i=0}^{K-1}$ set of initialized network parameters.
  for $n \leftarrow 1, N$ do
    $X, Y, S \leftarrow$ random mini-batch from dataset
    $Z_1 \leftarrow [XS]$
    for $i \leftarrow 1, K$ do                                  ▷ Iterate over levels
      $Z_{i+1} \leftarrow \mathrm{Enc}_i(Z_i)$                   ▷ $q(z_{i+1} \mid z_i)$
      $\hat{Z}_i \leftarrow \mathrm{Dec}_i(Z_{i+1})$             ▷ $p(z_i \mid z_{i+1})$
    end for
    $\hat{Y} \leftarrow \mathrm{Pred}_y(Z_K)$                    ▷ $p(y \mid z_K)$
    $\hat{S} \leftarrow \mathrm{Pred}_s(Z_K)$                    ▷ $p(s \mid z_K)$
    $S_r \leftarrow \mathrm{flip}(S, t)$                         ▷ Randomly flip $S$ with prob. $t$
    $\mathcal{L}_x \leftarrow \mathrm{MSE}(\hat{Z}_K, Z_K)$      ▷ Reconstruction loss
    $\mathcal{L}_y \leftarrow \mathrm{CE}(\hat{Y}, Y)$           ▷ Prediction loss
    $\mathcal{L}_s \leftarrow \mathrm{CE}(\hat{S}, S_r)$         ▷ Sensitive attribute loss
    $\mathcal{L} = \alpha\mathcal{L}_x + \beta\mathcal{L}_y + \lambda\mathcal{L}_s$
    $\Theta \leftarrow \Theta - \alpha\nabla_\Theta\mathcal{L}$  ▷ Update network param.
  end for
```

For the implementation, we approximate the log-likelihood $\log p(s \mid z)$ with the cross-entropy

$$\mathcal{L}_s = \mathrm{CE}(\hat{s}, s_r), \tag{4}$$

where $\hat{s}$ is our prediction, and rather than the ground truth, we use a stochastic process, s_r, that randomly flips between the possible classes for s. This last part is the key transformation that obfuscates the sensitive attribute. The other losses approximate the other two log-likelihoods. Namely, they are a mean-squared error minimization between the reconstruction $\hat{x}$ and the data x

$$\mathcal{L}_x = \mathrm{MSE}(\hat{x}, x), \tag{5}$$

and the cross-entropy for the downstream task

$$\mathcal{L}_y = \mathrm{CE}(\hat{y}, y). \tag{6}$$

Thus, our optimization problem (3) becomes the minimization of

$$\mathcal{L} = \alpha\mathcal{L}_x + \beta\mathcal{L}_y + \lambda\mathcal{L}_s, \tag{7}$$

where α, β and λ are weights associated to each loss.

B. Hierarchical Representations

To produce even fairer representations, we explore a hierarchical structure of K levels of representations. The overall idea is to construct representations following the probabilistic model presented before (see Section III-A). The ith level posses a latent variable z_i that follows

$$p(z_i, s, y) = \int p(z_i, z_{i+1}, s, y)\,\mathrm{d}z_{i+1} \tag{8}$$

as in our original joint (1), such that z_{i+1} is the latent representation for the z_i variable. In this way, it is possible to construct a set of K joints and learn them simultaneously by constructing a cascade of encoders and decoders (following Fig. 2). We regularize them with our loss function (7) on the highest level K. In our implementation, without the loss of generality, the first level is the input data, i.e., $z_1 = [x\ s]$.

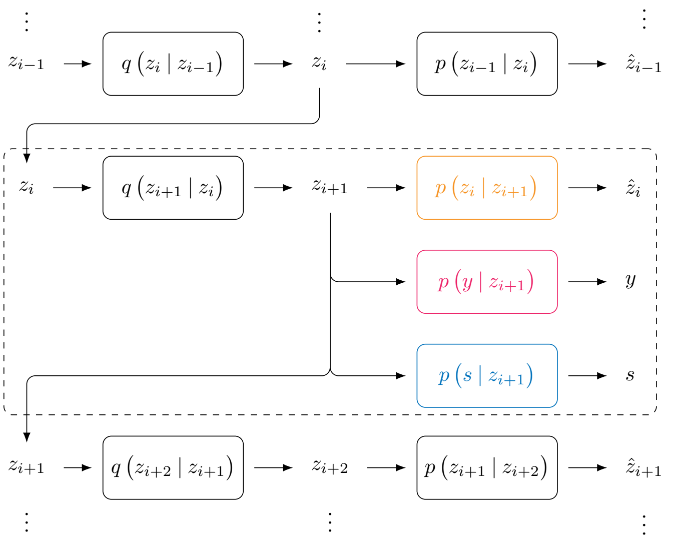

Fig. 2. Fair hierarchical representations learning. The data z_i of the ith level of the hierarchy is encoded into a latent variable z_{i+1} and then reconstructed $\hat{z}_i$. Its class y is predicted from the latent variable. Conversely, the corresponding sensitive variable s is trained such that its likelihood is uniform. These three tasks correspond to the likelihoods from our model (3), and are imposed over every level (not shown in the figure for simplicity).

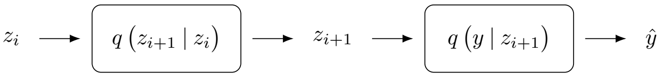

Fig. 3. Fair prediction learning. We transform the data z_i into its fair representation z_{i+1} through the learned encoder $q(z_{i+1} \mid z_i)$. Afterwards, we learn the predictor $q(y \mid z_{i+1})$ based on the learned fair representations. Note that the sensitive information is not used in this stage (cf. Fig. 2).

C. How the uniformity of the sensitive attribute implies fairness

Our method aims to enforce statistical parity on any downstream tasks that use the representation learned by the encoder part of our method, i.e., mapping of the input data into a fairer space. Recall that Statistical parity promotes the independence between the sensitive attribute (S) and the classifier outcome ($\hat{Y}$), i.e., $\hat{Y} \perp S$. Given the input data X, the encoder part of our model maps it into the latent space Z. We regularize the mapping by enforcing the uniformity of predicting S given Z, we do this by optimizing the regularizer network over a noisy version of S, in which the sensitive attribute of some individual is flipped. Our experiments showed in final representation individuals are equally likely to be assigned a given group label, i.e., $p(S \mid Z) \approx 0.5$. Moreover, Avigad [19] showed how randomness in a sequence of binary numbers can induce uniform distribution. When S is binary, enforcing the uniformity implies that $p(S = 1 \mid Z) = p(S = 0 \mid Z) \approx 0.5$, therefore $S \perp Z$. Any classifier f trained from Z will also be independent to S, i.e.,

$$\hat{Y} = f(Z) \perp S \to P(Y \mid S = 1) = P(Y \mid S = 0),$$

thus, achieving statistical parity.

To enforce the prediction of sensitive attribute from learned representation to be uniform (unreliable), our method leverages the weakness of DNNs to generalize when trained on noisy label [13]. In particular, Zhang et al. [20] show that deep neural networks can easily fit data with partially or completely noisy labels. In the case of partially noisy labels, the network still receives a signal from clean samples but also learns noise. As a result, the model does not generalize well on testing data, i.e., yields small training error but large error on the test set. As we want the latent variable to exhibit no information about the sensitive attributes, we regularize the latent space with a network head that predicts a noisy version of the sensitive attributes. As such, the latent space will degenerate with respect to the sensitive attribute and at test time the prediction of the sensitive attribute from the latent space will be unreliable, i.e., almost uniform.

Assuming that the conditional probability of the sensitive attribute distributes as a Bernoulli with probability $u(z)$, $p(s \mid z) = \mathcal{B}(u(z))$—hereinafter, we refer to $u(z)$ as u for brevity. Given that distribution of flipping the ground truth is a Bernoulli distribution $\mathcal{B}(r)$, we obtain that our flipping process (described in Section III-A) is a Bernoulli $\mathcal{B}(u - 2ur + r)$ (see the proof in the Appendix A). We show, by Theorem 1, that our cross-entropy loss (4) is an upper bound of the Kullback-Leibler (KL) divergence between the conditional $p(s \mid z)$ and a uniform distribution (when $\mathcal{B}(r = 0.5)$). Thus, by minimizing this loss (4), we enforce the conditional distribution to be the uniform distribution.

Theorem 1. *Let two independent Bernoulli distributions be $\mathcal{B}(u)$ and $\mathcal{B}(r)$, and a third related to the parameters of the previous ones be $\mathcal{B}(u - 2ur + r)$. The cross-entropy between $\mathcal{B}(u)$ and $\mathcal{B}(u - 2ur + r)$ is an upper bound of the Kullback-Leibler divergence between the two original distributions, i.e.,*

$$\mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(r)) < H(\mathcal{B}(r), \mathcal{B}(u - 2ur + r)). \qquad (9)$$

Proof. See Appendix B. □

D. Model Training

We present the pseudo-code of our learning process in Algorithm 1. We generalize the algorithm for layer-wise training. We trained the model components simultaneously to minimize the loss function (7). To enforce the uniformity of the sensitive attributes, a key step in the training process is flipping of the sensitive attribute with a probability t. In Feng et al.'s [12] work, the adversary is trained within each minibatch until convergence is reached for one training step of the generator, while Edwards and Storkey [7] alternate the gradient within each minibatch to update the generator or adversary. Adversarial setting is hard to optimize in practice and provides no convergence guarantees. In contrast to adversarial training, our method is easier to train (more stable) and straightforward. It is more efficient in terms of accuracy for similar levels of fairness (cf. Section IV).

IV. Experiments and Results

In this section, we present the experimental setup, empirical results, and the comparison of our approach, namely LFR-U, with other fair-representation learning methods. We compare

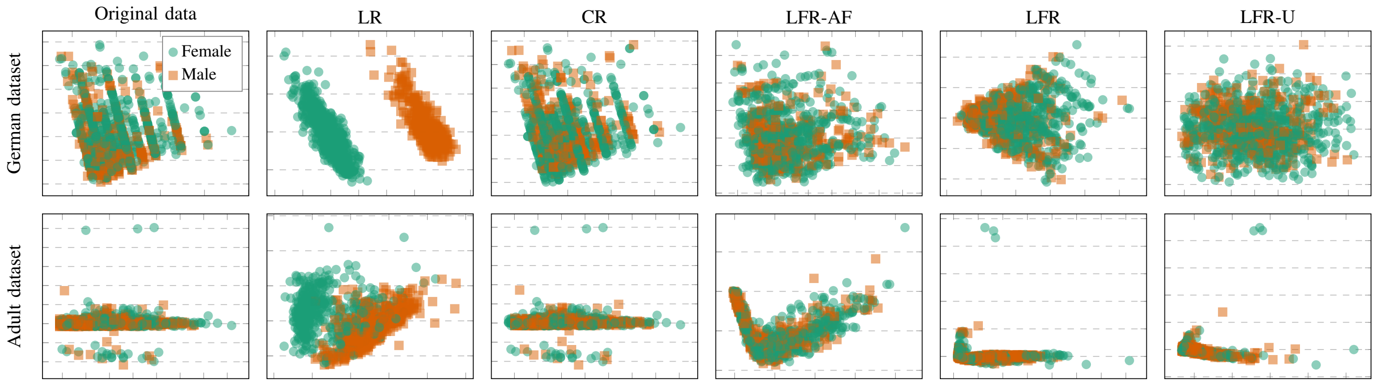

Fig. 4. Projection of the representations of 1000 random samples from the German and Adult datasets using PCA. The original data mixture is separated when learned with a non-uniform prediction on the sensitive attribute (LR). In contrast, enforcing uniformity on the sensitive attribute distribution yields an entangled space (LFR-U).

all the models in terms of accuracy, fairness, and their ability to learn representations that cannot predict the sensitive attribute[1].

A. Experimental setup

a) Baseline methods: We consider as baseline a simple logistic regression (LR) model trained on the original data with no fairness constraints. For the comparison, we consider Correlation Remover (CR) [21], which applies a linear transformation to the non-sensitive feature columns in order to remove their correlation with the sensitive feature columns. We also consider adversarial-based approaches such as work done by Edwards and Storkey [7], Madras et al. [8], who trained an adversary to distinguish between samples from the protected and unprotected groups, namely LFR. We consider statistical parity and, since we are not interested in the transferability of the learned representation, we used the binary cross-entropy loss for the adversary and training process proposed by Edwards and Storkey [7]. Furthermore, we consider the work of Feng et al. [12], namely LFR-AF, where they trained an adversary to minimize the Wasserstein distance between the distribution of protected and unprotected groups. We used the implementation provided by the authors. We also consider a simple Autoencoder (AE) model that minimizes the loss of reconstructing the input data from the latent space, and then use encoder part to map (compress) the training data logistic regression model. The goal of using an autoencoder is to analyze the effect of the compression over the fairness on downstream task.

b) Data: We experimented on two datasets: German [22] and UCI Adult Income [23] datasets. The German dataset comprises 1000 samples of bank account information described with 21 features. The target variable y predicts whether an account is good or bad. The adult income dataset contains $48\,843$ instances of demographic information of American adults, described with 14 features that are a mixture of categorical, ordinal, and numerical data types. The target indicates whether personal income levels are above or below USD $50\,000$ per year. We used sex (Male or Female) as the sensitive attribute in both datasets.

c) Metrics: In addition to the typical accuracy, we also consider the following fairness metrics used in the literature. Let $X = \{x_i\}_i^N$; $x_i \in \mathbb{R}^d$ be our data, $S = \{s_i\}_i^N$ our binary sensitive attribute such that s_i is contained within x_i, and $\eta(\cdot)$ a classifier that maps a given sample x_i to a class label $\hat{y}_i$. *Statistical parity* [5] is a fairness notion that promotes the independence between the positive prediction and sensitive attribute through

$$P\left(\eta(X) = 1 \mid S = 1\right) = P\left(\eta(X) = 1 \mid S = 0\right). \tag{10}$$

We measure statistical parity as

$$\Delta_{\text{DP}} = \left| \frac{\sum\limits_{i:s_i=1} \eta(x_i)}{N_1} - \frac{\sum\limits_{x:s_i=0} \eta(x_i)}{N_0} \right|, \tag{11}$$

where N_s is the number of samples with the sensitive attribute set to s.

We also consider *equalized odds* [3, 4], which promotes the conditional independence between the prediction outcome and the sensitive attribute given the class label. That is

$$P\left(\eta(X) = 1 \mid S = 1, Y = y\right) = \\ P\left(\eta(X) = 1 \mid S = 0, Y = y\right), \tag{12}$$

for $y \in \{0, 1\}$. We measure it as

$$\Delta_{\text{EOD}} = \sum_{y \in \{0,1\}} \left| \frac{\sum\limits_{i:s_i=1,y_i=y} \eta(x_i)}{N_1^{(y)}} - \frac{\sum\limits_{i:s_i=0,y_i=y} \eta(x_i)}{N_0^{(y)}} \right|, \tag{13}$$

where $N_s^{(y)}$ is the number of samples with the sensitive attribute set to s and the class label set to y.

Equal opportunity is similar to equalized odds (13) [3, 4]. However, it only considers the case where $y = 1$. We refer to it as Δ_{EOP}.

[1] Source code: https://github.com/patrikken/lfr-u

d) Model: All the components of models are defined as multi-layer perceptron (MLP). The autoencoder part is defined with model with two levels of representation (layers). The first hidden layer is 15 units (z_1), while the second one (i.e., latent space) is 10 and 8 units (z_2) for the Adult and German datasets, respectively. Classifiers and regularizer networks are defined as single-layer MPL with the number of input units equal to the dimension of the latent space (z_2). LFR-AF, AE, and LFR models are built with the same autoencoder architecture. For a fairer comparison, we enforced uniformity only at highest hierarchy (z_2).

We used Adam optimizer [24] with step size of 0.001 for 1000 steps and a batch size of 64. We, then, trained a logistic regression model on the leaned representations to predict the class labels or the sensitive attribute to assess the level of dependency between the given representation and the learned representation. We split the dataset into train and test sets. We used the training set to learn the representations. We map the test set into the learned space and used it to train the logistic regressor using 10-fold cross-validation. We run the experiment seven times and averaged the results shown in Figs. 5 and 6. This evaluation regressor is different from the classifier, $p(y|z)$, used at training time to regularize the model—cf. $\mathcal{L}_y$ (6).

B. Learned Representations

To examine how the groups are distributed in the latent space, we learned representations using the different methods considered and projected them into two dimensions using PCA. Figure 4 showcases the distribution of groups (Male and Female) in the latent space. We trained representation using our approach without inducing the uniformity of the sensitive attribute in the latent space, i.e., without the noisy prediction of the sensitive. Therefore, the learned representation (Fig. 4 second column–LR) is enforced to discriminate between the group label and the class label. As a result, the second column of Fig. 4 shows that without the uniformity, groups are linearly separable in the latent space. In contrast, we enforce the uniformity in latent space, the learned representations are entangled and therefore hard to distinguish (or predict).

Figure 4 also shows that similarly to other fair representation learning methods, our methods is able to entangle different groups in the latent space. This suggests that our method of making the prediction of the sensitive attribute unreliable by enforcing its uniformity is as good as defeating an adversary (LFR and LFR-U). The advantage of our approach for obtaining this representation is that it is easier to train and does not suffer from the instability of the adversarial approach, which requires to defeat an adversary and can sometimes be counterproductive.

C. Fairness Analysis

Figure 5 presents the fairness performances of logistic regressions models trained with data representations of each particular method. Overall, the representation yielded by our method (LFR-U) outperforms the baseline models in terms of accuracy and provides better performance in terms of statistical parity, equalized odds, and equal opportunity compared on the considered datasets. Whilst our method is designed to enforce fairness in terms of statistical parity by inducing uniformity of the sensitive attribute, our empirical results show that this has a positive impact on other fairness notions (equalized odds and equal opportunity).

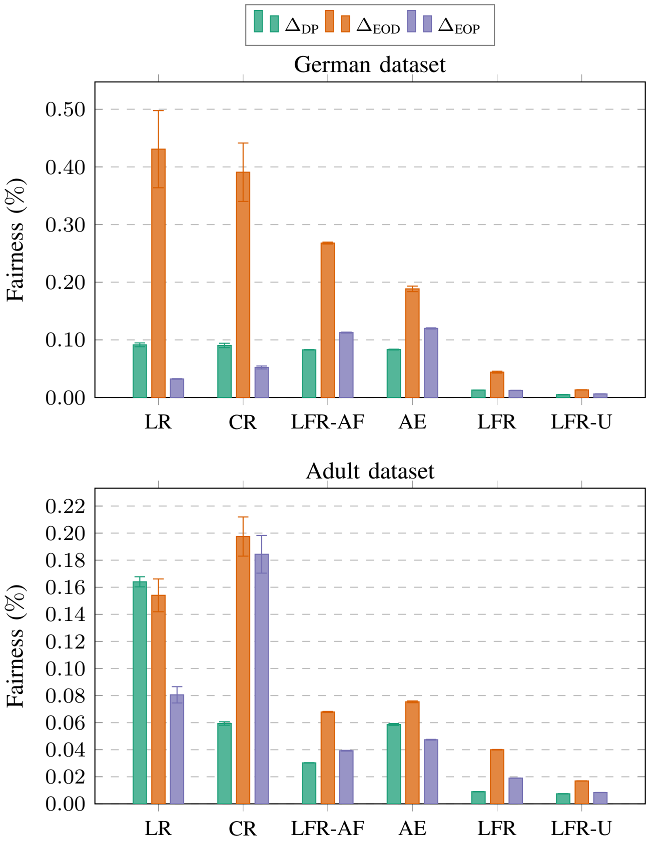

Fig. 5. Fairness performance (↓) of classifiers trained with different representations on the German and Adult datasets. Our model provides better fairness performance compared to the baseline models considered while giving similar accuracy compared to other fair representation techniques(cf. Fig. 6). The symbol ↑ means higher values is better, and ↓ lower is better.

While other methods learned representations that decreased the accuracy compared to the baseline methods, ours impacted it less (Fig. 4). The goal of the adversarial approach for fair representation learning is to completely remove the dependence on the sensitive attribute by defeating an adversary. However, this optimization process does not guarantee that the trained adversary is the optimal one and that there is no other adversary that can predict the sensitive attribute from the learned representation. In contrast, we enforce the uniformity of the sensitive attribute in the latent space by leveraging a noisy prediction of the sensitive attribute from the latent space, such that each data point has the same probability (0.5) to be assigned to a given group. Thus, instead of focusing on learning optimal features that fool a classifier, we focus on obtaining features that are equally likely to produce every class in the sensitive attribute. The results (Fig. 5) also show that in an unsupervised setting, a simple autoencoder (AE) reduces the dependence on the sensitive attribute in the latent space and thus improves fairness. This shows that sensitive information can be cancelled out by compression, which can be useful in the case where sensitive attributes are not observed.

D. Sensitive Attribute Obfuscation

We analyzed the amount of information of the sensitive attribute that is still encoded in the latent space after the learning by training a classifier to predict it from the learned representation. We expected (ideally) the accuracy of this classifier to be very low, i.e., the dependence on the sensitive attribute has been removed from the latent space. A classifier trained on the original data is quite accurate in predicting the sensitive attribute s. As shown in Fig. 6, a logistic regression model trained to predict the sensitive attribute in the original dataset achieved more than $83\,\%$ and $75\,\%$ of accuracy in the Adult and German datasets respectively. While our method and adversarial-based methods provided less than $68\,\%$ accuracy. This shows that all approaches try to obscure information about s in the latent space, while ours also maintains better utility.

From privacy perspectives, a logistic regression model can be seen as a *weak adversary*, i.e., a stronger adversary such as a neural network can archive better performance on predicting the sensitive attributes from the learned representation. However, although the objectives (fairness and privacy) converge, the primary goal is not to make the latent space robust against an adversary wanting to reconstruct the sensible attributes. Thus, the use of logistic regression in this case is not to evaluate the robustness against an adversary but to evaluate the level of obfuscation of the sensitive attribute, which is necessary for fairness.

We also tested how much information about the sensitive attribute is retained in the representation learned by our method using a stronger adversary, i.e., a neural network. The adversary is a two-layer MLP with 25 and 15 units in the first and the second hidden layers respectively. The network was trained seven times and the accuracy was averaged across runs. The optimizer and the used parameters were the same as in the previous experiment. Table I shows the accuracy of predicting the sensitive attribute and class label using the original representation and the representation provided by our methods in the Adult and German datasets. These results show that even a stronger adversary such as a neural network is not accurate in reconstructing the sensitive attribute from the representation regularized with our method. For instance, on the Adult dataset, the MLP model achieved 84.5% accuracy in predicting the sensitive attribute vs. 68.6% with our representation, while achieving 82.6% in predicting the class label compared to 85% with the original representation. The representation used to train the adversary was optimized using both the classifier and the regularizer heads, i.e., $\beta = 1$ and $\lambda = 1$ (7), which shows that the representation preserved the useful information for target downstream task while obfuscating information about sensitive attributes.

TABLE I
COMPARISON OF THE ACCURACY OF PREDICTING y (↑) AND THE SENSITIVE ATTRIBUTE s (↓) USING A MLP AS ADVERSARY ON ADULT AND GERMAN DATASET.

Dataset	Representation	Accuracy $s(\downarrow)$	Accuracy $y(\uparrow)$
Adult	Original	0.845	0.856
	Ours	0.686	0.828
German	Original	0.690	0.783
	Ours	0.592	0.701

TABLE II
COMPARISON OF ACCURACY OF PREDICTING y (↑) AND FAIRNESS (↓) METRICS OF OUR METHOD WHEN REGULARIZING THE HIGHEST HIERARCHY (z_2), OR BOTH (z_1 AND z_2) ON THE ADULT DATASET.

Regul.	Acc.	Δ_{DP}	Δ_{EOD}	Δ_{EOP}
Single	0.868	0.090	0.120	0.093
Both	0.876	0.089	0.085	0.024

E. Hierarchical Regularization

We analyzed the effect of regularizing our proposed model in the hierarchical manner compared to only regularizing the highest level. To evaluate the effect of this full regularization, we trained (using the Adult dataset) our method (LFR-U) enforcing uniformity at the highest level (z_2) and at both levels (z_1 and z_2). The results show that regularizing at multiple levels further improves fairness with respect to equalized odds and equal opportunity (Table II). We highlight that existing models only regularize a single level. Thus, all of our other experiments are with a single level regularization for a fair comparison. For this set of experiments, we trained the representation with higher weight ($\beta = 1$) on classifier head without reconstruction ($\lambda = 0$), which led to better accuracy on predicting the class label and lower fairness.

F. Ablation Study of Uniformity

We analyzed how the flipping probability t affects the learned representation. The probability t is a hyper-parameter that can be tuned to improve the accuracy or fairness performance of the learned representations. The best choice of the flipping ratio depends on how balanced the different groups in the dataset are. When groups are not well-balanced in the dataset, samples from well-represented group will have higher probability to be flipped compared to samples from underrepresented groups, which might restore the group balance during the training. In previous experiments, we used $t = 0.5$. Figures 8 and 7 show accuracy and fairness performance across a range of different values of t for the German and the Adult datasets respectively.

When we set $t = 0$, it is equivalent to the use the original dataset. Thus, the model learns to separate different groups in the latent space. Therefore, the non-uniformity of the sensitive attribute is enforced. Similarly, when $t = 1$, all the sensitive attributes are flipped for the samples; and, again, the non-

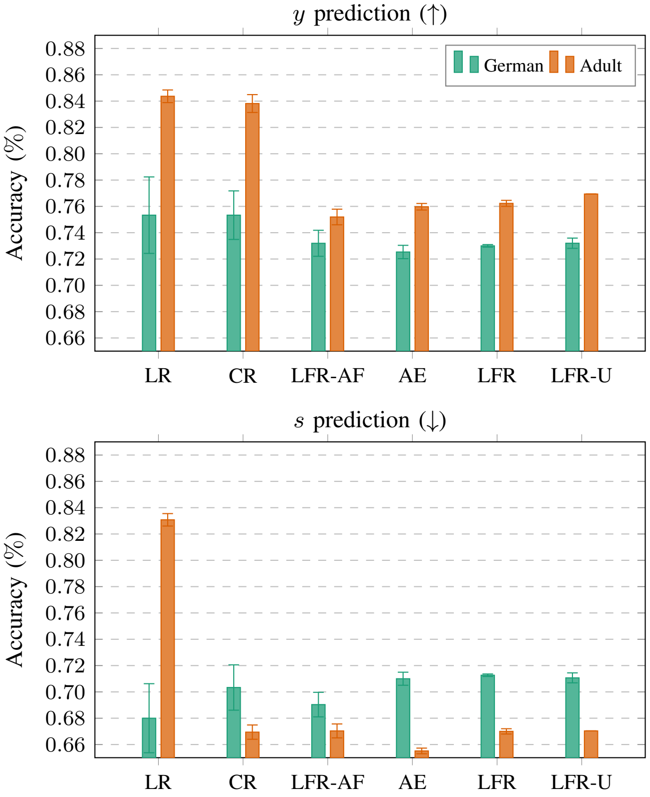

Fig. 6. Prediction accuracy of class labels y (↑) and sensitive attribute s (↓) from the learned representation of different models. Our model (LFR-U) provides comparative accuracy for predicting the class label and the sensitive attribute s. Lower accuracy of predicting s shows that our method attempts to obfuscate the sensitive attribute information in the data as adversarial-based models. The symbol ↑ means higher values is better, and ↓ lower is better. All evaluations are performed on the hold-out test sets.

uniformity is enforced but in a reversed way. Figure 7 shows in the Adult dataset, the value of t that provides the best fairness performances is 0.5, the figure shows that the level of unfairness increases as the value t moves away from 0.5. Similarly, for the German dataset the value of t that provides better fairness-accuracy trade-off is between 0.5 and 0.6 (Fig. 8).

V. Conclusion and Future Work

In this work, we introduced the idea that making a model's learned representations unreliable w.r.t. the sensitive attributes enforces fairness. Our experiments showed that even while using full data samples (i.e., including the sensitive attribute) for training, we could obfuscate the sensitive attributes while maintaining the prediction accuracy of the related tasks. Moreover, we demonstrated experimentally on two datasets the advantages of our proposal with respect to the accuracy and fairness metrics. Interestingly, though our method is designed to improve fairness in terms of statistical parity, our results showed that other fairness notions such as equalized odds and equal opportunities, got positively impacted. To summarize, our proposal is to enforce uniformity over a prediction head for the sensitive attributes, and this head is used as a regularizer for the learned representations. Moreover, we explored a hierarchical

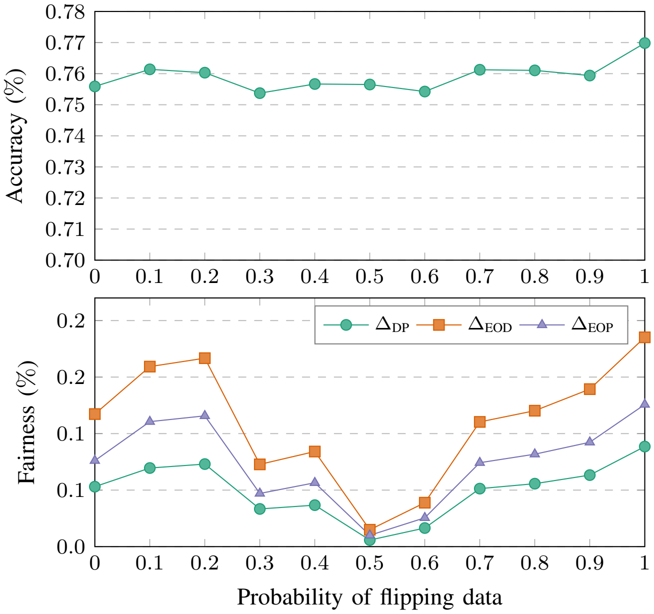

Fig. 7. Accuracy of y (↑) and fairness (↓) performance of LFR-U with different flipping probability on Adult dataset.

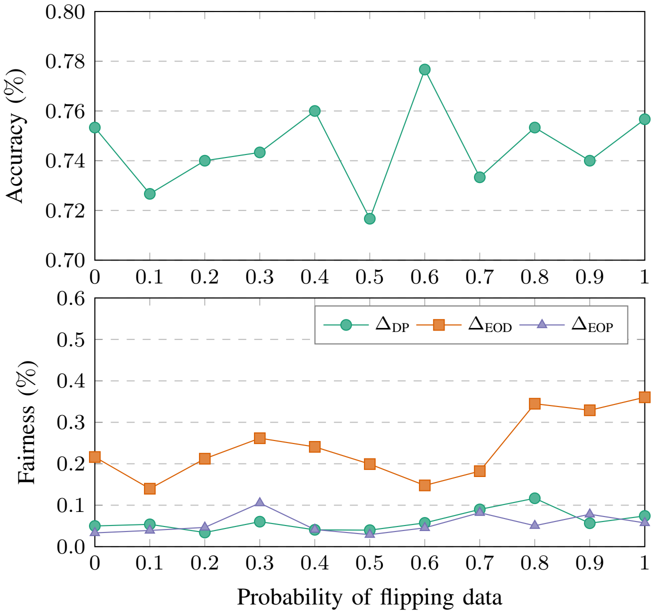

Fig. 8. Accuracy of y (↑) and fairness (↓) performance of LFR-U with different flipping probability on the German dataset.

model of representations that helps in enforcing the fairness while providing a better prediction performance on class labels. However, we observed a reduced accuracy compared to the model trained without fairness constrains. Although this decrease is less than that for other fair representation learning techniques, reducing the tradeoff between fairness and accuracy remains an important issue within the fair ML community. Another possible research direction could be provide theoretical guarantees for the representation learned in Algorithm 1, i.e., given the participants in a decision-making process, what fairness guarantees the proposed method can provide.

References

[1] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[2] P. J. Kenfack, A. M. Khan, S. A. Kazmi, R. Hussain, A. Oracevic, and A. M. Khattak, "Impact of model ensemble on the fairness of classifiers in machine learning," in *2021 International Conference on Applied Artificial Intelligence (ICAPAI)*. IEEE, 2021, pp. 1–6.

[3] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," *arXiv preprint arXiv:1610.02413*, 2016.

[4] S. Verma and J. Rubin, "Fairness definitions explained," in *2018 ieee/acm international workshop on software fairness (fairware)*. IEEE, 2018, pp. 1–7.

[5] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012, pp. 214–226.

[6] M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," *arXiv preprint arXiv:1703.06856*, 2017.

[7] H. Edwards and A. Storkey, "Censoring representations with an adversary," *arXiv preprint arXiv:1511.05897*, 2015.

[8] D. Madras, E. Creager, T. Pitassi, and R. Zemel, "Learning adversarially fair and transferable representations," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3384–3393.

[9] P. J. Kenfack, A. M. Khan, R. Hussain, and S. Kazmi, "Adversarial stacked auto-encoders for fair representation learning," *arXiv preprint arXiv:2107.12826*, 2021.

[10] E. Creager, D. Madras, J.-H. Jacobsen, M. Weis, K. Swersky, T. Pitassi, and R. Zemel, "Flexibly fair representation learning by disentanglement," in *International conference on machine learning*. PMLR, 2019, pp. 1436–1445.

[11] D. Moyer, S. Gao, R. Brekelmans, G. V. Steeg, and A. Galstyan, "Invariant representations without adversarial training," *arXiv preprint arXiv:1805.09458*, 2018.

[12] R. Feng, Y. Yang, Y. Lyu, C. Tan, Y. Sun, and C. Wang, "Learning fair representations via an adversarial framework," *arXiv preprint arXiv:1904.13341*, 2019.

[13] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[14] Y. Bai, E. Yang, B. Han, Y. Yang, J. Li, Y. Mao, G. Niu, and T. Liu, "Understanding and improving early stopping for learning with noisy labels," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 392–24 403, 2021.

[15] P. Chen, B. B. Liao, G. Chen, and S. Zhang, "Understanding and utilizing deep neural networks trained with noisy labels," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1062–1070.

[16] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International conference on machine learning*. PMLR, 2013, pp. 325–333.

[17] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The variational fair autoencoder," in *International conference on learning representations*, 2016.

[18] S. Dutta, D. Wei, H. Yueksel, P.-Y. Chen, S. Liu, and K. Varshney, "Is there a trade-off between fairness and accuracy? a perspective using mismatched hypothesis testing," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2803–2813.

[19] J. Avigad, "Uniform distribution and algorithmic randomness," *The Journal of Symbolic Logic*, vol. 78, no. 1, pp. 334–344, 2013.

[20] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[21] S. Bird, M. Dudík, R. Edgar, B. Horn, R. Lutz, V. Milan, M. Sameki, H. Wallach, and K. Walker, "Fairlearn: A toolkit for assessing and improving fairness in ai," *Microsoft, Tech. Rep. MSR-TR-2020-32*, 2020.

[22] L. Jeff, M. Surya, K. Lauren, and A. Julia, "How we analyzed the compas recidivism algorithm," 2016.

[23] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

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

Appendix A
Flipping process

The flipping process (cf. Section III-A) takes a Bernoulli over the sensitive attribute, $\mathcal{B}(u)$, and flips it using another independent Bernoulli with probability r, $\mathcal{B}(r)$. Given $s \sim \mathcal{B}(u)$, $f \sim \mathcal{B}(r)$, we compute the resulting sensitive value, s_r, from their joint $p(s, f) = p(s)p(f)$, where there are four possibilities:

$$p(s=1, f=1) = ur, \qquad \text{(false fail)} \tag{14}$$

$$p(s=1, f=0) = u(1-r), \qquad \text{(true success)} \tag{15}$$

$$p(s=0, f=1) = (1-u)r, \qquad \text{(false success)} \tag{16}$$

$$p(s=0, f=0) = (1-u)(1-r). \qquad \text{(true fail)} \tag{17}$$

It follows that,

$$\begin{aligned} p(s_r=1) =& p(s=0, f=1) + p(s=1, f=0), && \text{(18a)}\\ =&(1-u)r + u(1-r), && \text{(18b)}\\ =&r - ur + u - ur, && \text{(18c)}\\ =&u - 2ur + r, && \text{(18d)} \end{aligned}$$

and, similarly we find

$$\begin{aligned} p(s_r=0) =& p(s=1, f=1) + p(s=0, f=0) && \text{(19a)}\\ =&1 - u + 2ur - r && \text{(19b)} \end{aligned}$$

Consequently, the randomized sample $p(s_r) \sim \mathcal{B}(u - 2ur + r)$.

Appendix B
Proof of Theorem 1

(Restated) Theorem 1: Let two independent Bernoulli distributions be $\mathcal{B}(u)$ and $\mathcal{B}(r)$, and a third related to the parameters of the previous ones be $\mathcal{B}(u-2ur+r)$. The cross-entropy between $\mathcal{B}(u)$ and $\mathcal{B}(u-2ur+r)$ is an upper bound of the Kullback-Leibler divergence between the two original distributions, i.e.,

$$\mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(r)) < H(\mathcal{B}(r), \mathcal{B}(u-2ur+r)). \tag{20}$$

Proof. The KL divergence between the Bernoullis is

$$\begin{aligned}&\mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(u-2ur+r))\\ &= u\log\frac{u}{u-2ur+r} + (1-u)\log\frac{1-u}{1-u+2ur-r},\end{aligned} \tag{21}$$

and similarly

$$\mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(r)) = u\log\frac{u}{r} + (1-u)\log\frac{1-u}{1-r}. \tag{22}$$

By manipulating the former KL (21), we obtain

$$\begin{aligned}&\mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(u-2ur+r))\\ &= u\log\frac{u}{r}\frac{r}{u-2ur+r} + (1-u)\log\frac{1-u}{1-r}\frac{1-r}{1-u+2ur-r}, &\text{(23a)}\\ &= \mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(r))\\ &+ u\log\frac{r}{u-2ur+r} + (1-u)\log\frac{1-r}{1-u+2ur-r} &\text{(23b)}\\ &= \mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(r)) + C(u,r), &\text{(23c)}\end{aligned}$$

where $C(u,r)$ is a function of u and r.

Then, we can expand the LHS

$$\begin{aligned}\mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(u-2ur+r)) =& \mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(r))\\ &+ C(u,r), &\text{(24)}\\ H(\mathcal{B}(u), \mathcal{B}(u-2ur+r)) =& \mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(r))\\ &+ H(\mathcal{B}(u)) + C(u,r), &\text{(25)}\end{aligned}$$

and we can bound it by

$$H(\mathcal{B}(u), \mathcal{B}(u-2ur+r)) > \mathrm{KL}(\mathcal{B}(u) \parallel \mathcal{B}(r)). \tag{26}$$

□