Information Removal at the bottleneck in Deep Neural Networks

Enzo Tartaglione
enzo.tartaglione@telecom-paris.fr

LTCI, Télécom Paris,
Institut Polytechnique de Paris

Abstract

Deep learning models are nowadays broadly deployed to solve an incredibly large variety of tasks. Commonly, leveraging over the availability of "big data", deep neural networks are trained as black boxes, minimizing an objective function at its output. This however does not allow control over the propagation of some specific features through the model, like gender or race, for solving some uncorrelated task. This raises issues either in the privacy domain (considering the propagation of unwanted information) or bias (considering that these features are potentially used to solve the given task). In these contexts, the development of a strategy specifically purposed to remove some part of the information in these models is critical.

In this work, we propose IRENE, a method to achieve information removal at the bottleneck of deep neural networks, which explicitly minimizes the estimated mutual information between the features to be kept "private" and the target. Experiments on a synthetic dataset and on CelebA validate the effectiveness of the proposed approach, and open the road toward the development of approaches guaranteeing information removal in deep neural networks.

1 Introduction

Currently, a significantly large portion of problems is being solved through the deployment of deep learning models, considered by most the "universal problem-solving tool" [26]. For instance, these are being deployed in high-stakes applications, ranging from candidate job hiring to facial recognition systems. It is a known problem that deep models, when trained, take advantage of "spurious" correlations from their training data, which lead to significant performance variance across sub-populations, sometimes across sensitive attributes like race and gender. This causes what is known, in the literature, as *bias* of the deep model.

Learning these spurious correlations has several effects, of which the most evident one is poor performance on under-represented dataset sub-populations and out-of-distribution test data [15]. Finding a solution to the problem of biases in the deep models is currently a topic of broad interest from the community [12, 27]. Many works have tried to tackle this problem from many different perspectives: for example, complex architecture like transformers could solve the data biasing problem with proper priors [27], or methodologically, looking at unsupervised scenarios it will be possible to prevent the extraction of biases [15]. A large part of approaches intrinsically propose a re-weighting over the biased information, but removing this information is a complex matter.

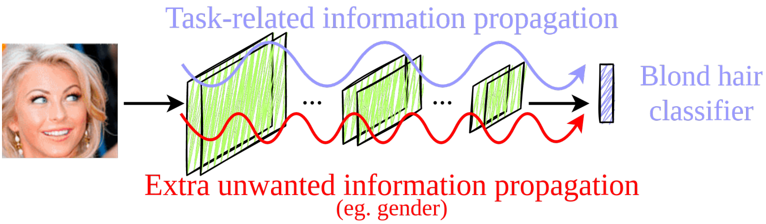

Task-related information propagation
...
...
Blond hair
classifier
Extra unwanted information propagation
(eg. gender)

Figure 1: From the latent space of ANN models there might be information leakage, allowing an attacker to recover sensitive/private data. Our goal is to prevent this, hiding the private information.

A lot of interest around debiasing is indeed devoted because it would be irresponsible and unethical to promote algorithms that could possibly create damages or lean towards discrimination of people whose data are used by the Artificial Intelligence (AI) [17, 22]. The European Union has already obtained the role of legal influencer as far as the discipline of data protection is concerned about drafting the General Data Protection Regulation (GDPR) [1]: since 2018 it is being working in trying to create the conditions to regulate the AI. Towards this end, re-weighting the information in the learning process might not be sufficient to guarantee complete protection towards the propagation (and for instance, the use) of some sensitive, or discriminatory information, by the AI model [5]. Figure 1 depicts the scenario in which, besides the task-related information (in this case, the color of the hairs), the information of the gender is also propagated. This behavior of deep models poses issues from ethical, security, and legal perspectives.

In this work, we propose IRENE, a method to remove specific information at the bottleneck of deep neural networks. This method relies on the estimation of the information we desire to maintain "private", with the employment of an auxiliary classifier. This enables the estimation of the information (at the bottleneck) we wish to remove. Then, we achieve the information removal through the minimization of a differentiable proxy of the mutual information. To the best of our knowledge, IRENE is the first work proposing an information removal at the bottleneck of deep neural networks.

In short, the main contributions from this work are listed here below.

- To the best of our knowledge, IRENE is the first work directly tackling the problem of information removal *inside* neural networks, although the theory on the information propagation at the bottleneck in deep neural networks is already known [31].

- IRENE *really* removes information. Some works in the debiasing literature try to remove it explicitly by modeling features correlations [29] or using adversarial setups [4]; but their real aim is not to really remove information related to bias: they *re-weight* it, in order to achieve higher performance on the trained task [4, 5, 6].

- IRENE removes information thanks to a differentiable formulation of the mutual information *in a simple yet efficient way*. A similar concept is achievable using a variational approach, but in our case, we do not require to pre-train the encoder or to impose Gaussian priors at the bottleneck. We do not require as well adversarial frameworks, which are more difficult and expensive to optimize [20, 33].

The rest of the paper is organized as follows. In Section 2 an overview on the close, related literature is provided; Section 3 presents IRENE, then Section 4 shows and discusses the experiments conducted. Finally, Section 5 draws the conclusions.

2 Related works

In this section we discuss the literature related to information removal in deep learning. We can group the related literature in two large families: privacy preservation approaches and debiasing techniques.

Privacy preservation. Very recently, thanks to the increase in computational capabilities, many works have been proposed on privacy-preserving in computational frameworks. A work by Dwork *et al.* [11] studied how much noise is required to guarantee "differential privacy" [10] from data. Duchi *et al.* [9] formalized convergence boundaries for training and the trade-off between privacy guarantees and the utility of the resulting statistical estimators. This knowledge has been also recently applied to deep learning frameworks, with a work by Abadi *et al.* [2], by introducing some tuned noise in the update rule. A different approach to preserve data privacy is the so-called "federated learning". In general, private datasets are held by the proprietary of the data, who is directly training a local neural network model. Now, the parameters of the models are sent to a master node, which is then propagating to all the private computational nodes the general configuration of the parameters. This approach has been proposed by Shokri and Shmatikov [24], and allows parallel and private computation. However, it does not take into account any ethical bias, like gender or race: what it guarantees is that the original data are not directly shared, but some sensible information actually is.

Debiasing. It is known that datasets are typically affected by biases. In their work, Torralba and Efros [32] showed how biases affect some of the most commonly used datasets, drawing considerations on the generalization performance and classification capability of the trained ANN models. Following up on a similar idea, the effectiveness of content-style disentanglement has been explored, besides an estimation of biased content, by Liu *et al.* [18]. Working at the dataset level is in general a critical aspect, and greatly helps in understanding the data and its structure [8]. Some works suggest the use of GANs to entirely clean up the dataset with the aim of providing fairness [23, 33], others like Mandras *et al.* [20] insert a GAN in the middle of the architecture to clean-up the internal representation of data. In general, training such an architecture is a very delicate and complex process, and it does not provide explicit fairness at inference time, as generative adversarial networks are used to generate training data. Another interesting possibility has been proposed by Kim *et al.*, with the use of adversarial learning and gradient inversion to eliminate the information related to the biases in the model [13]. Bahng *et al.* [4] develop an ensembling-based technique, called *ReBias*: this consists in solving a min-max problem where the target is to promote the independence between the network prediction and all biased predictions. Identifying the "known unknowns" [3] and optimizing on those (using a neural networks ensemble) is the approach proposed by Nam *et al.* [21], or with unsupervised surface exploration, by Khrisnakumar *et al.* [14]. A similar approach is followed by Clark *et al.* in their LearnedMixin [7]. The exploration of the embedding space, looking for biases and addressing the bias mitigation problem, and discouraging the optimization directions which favor the classifier to be biased, is proposed by Thong *et al.* [30]. The inclusion of a regularization term, addressing a similar concern but with no memory overhead, has also been recently proposed by Tartaglione *et al.* [29].

Considering the specific problem IRENE addresses, despite having similar objectives to privacy preservation approaches, the setup of the problem makes IRENE closer to debiasing approaches, despite its different goals. A recent work by Song *et al.* [25] reported that using standard training strategies to train some state-of-the-art models, allows information not relevant to the learning task to be stored inside the network. Such behavior is possible because of the typically oversized ANNs trained to solve a task [28]. In their experiments, Song *et al.* show how accurately they can recover some non-directly related to training information, showing the potential, unwanted information propagation. In the next section, IRENE will be presented and discussed.

3 IRENE: information removal at the bottleneck

In this section, we are going to introduce IRENE, our method to remove information at the bottleneck in deep neural networks. As we will see, for a given learning task, there can naturally be some mutual information between the private information ($\hat{v}$) and the target ($\hat{y}$). Typical debiasing approaches mitigate, or rather, balance these two, towards an improvement of the performance on some target task. In our case, we target the pure minimization of such a term. Here follows a general presentation of the learning framework and the proposed approach.

3.1 Private features and target features

Let us train a given deep neural network such that, given an input x, it produces an output y: the vanilla training ambition is to make it as close as possible to the ground truth $\hat{y}$. Towards this end, a loss function $\mathcal{L}(y,\hat{y})$ is minimized (eventually, besides some additional regularization constraint). Hence, the only control over the information being learned lies in the ground truth label $\hat{y}$.

Let us define here, for every input sample x, a companion ground truth label $\hat{v}$, marking a piece of different information from the task-related one: this one we wish *not* to be propagated in the model, or rather, not to be propagated from a certain layer (we define this *bottleneck layer*) onward. Naturally, there exists some correlation between the i-th task-related ground truth $\hat{y}$ and the j-th $\hat{v}$, which can be modeled by the joint probability $p_{\hat{y}\hat{v}}$. Having this, we know the mutual information $\mathcal{I}(\hat{y},\hat{v})$ being

$$\mathcal{I}(\hat{y},\hat{v}) = \sum_i \sum_j p_{\hat{y}\hat{v}}(i,j) \log\left(\frac{p_{\hat{y}\hat{v}}(i,j)}{p_{\hat{y}}(i)p_{\hat{v}}(j)}\right). \tag{1}$$

In the case of a perfect learner (ie. a model making no errors on the training set), since $y^{\mu} = \hat{y}^{\mu}\ \forall\mu$ (where μ is then input sample index), necessarily we will have $\mathcal{I}(y,\hat{v}) = \mathcal{I}(\hat{y},\hat{v})$. However, from the debiasing literature, in the cases when there exist certain $p_{\hat{y}\hat{v}}(i,j) \gg p_{\hat{y}\hat{v}}(i,k)$, the learning is imperfect as there are attractors in the learning process, making $\mathcal{I}(y,\hat{v}) > \mathcal{I}(\hat{y},\hat{v})$. In such a context, the debiasing literature leverages over such a gap to extract the information over $p_{\hat{y}\hat{v}}$ imbalances, correcting them. The purpose of debiasing algorithms; however, is not to achieve a "perfect learner" on the (biased) training set, but to improve the performance on the (unbiased) validation set, where $p_{\hat{y}\hat{v}}(i,j) \approx K \forall i,j$ (hence, uniformly distributed).

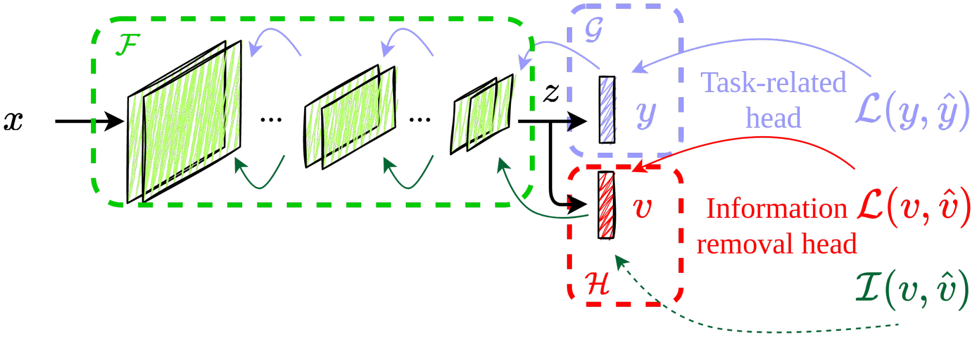


Figure 2: Sketch of IRENE. Backward lines display the back-propagation of the single contributions, and dashed lines imply the non-computation of the gradients for the target layer.

Let us divide our model into an encoder $\mathcal{F}$ and a classifier $\mathcal{G}$: we name the output of the encoder *bottleneck* z, which is also the input of $\mathcal{G}$. In order to improve the pure performance of the classifier model, it will be a necessary condition to have $\mathcal{I}(\hat{y}, \hat{v}) = \mathcal{I}(y, \hat{v})$ but this is not sufficient to guarantee that $\mathcal{I}(\hat{y}, \hat{v}) = \mathcal{I}(z, \hat{v})$. Indeed, debiasing methods target a balanced extraction (and eventually, balanced use) for $\hat{v}$: the removal/decorrelation concerning the target task is not necessarily addressed to achieve their purposes. The task we want to achieve is to explicitly minimize $\mathcal{I}(z, \hat{v})$: here follows our method designed to address such a task.

3.2 How to remove information at the bottleneck

Algorithm 1 Training with IRENE.

```
procedure ONE_ITERATION_IRENE($x$, $\hat{y}$, $\hat{v}$, $\theta_{\mathcal{F}}$, $\theta_{\mathcal{G}}$, $\theta_{\mathcal{H}}$)
    $z \leftarrow \mathcal{F}(x, \theta_{\mathcal{F}})$
    $y \leftarrow \mathcal{G}(z, \theta_{\mathcal{G}})$
    $v \leftarrow \mathcal{H}(z, \theta_{\mathcal{H}})$
    $\text{grad}(\theta_{\mathcal{G}}) \leftarrow \text{backward}(\alpha \mathcal{L}(y, \hat{y}))$
    $\text{grad}(\theta_{\mathcal{H}}) \leftarrow \text{backward}(\mathcal{L}(v, \hat{v}))$
    $\text{grad}(\theta_{\mathcal{F}}) \leftarrow \text{backward}(\alpha \mathcal{L}(y, \hat{y}) + \gamma \mathcal{I}(v, \hat{v}))$
    Update with chosen optimizer
end procedure
```

Directly addressing the problem of minimizing $\mathcal{I}(z, \hat{v})$ is either computationally unfeasible or it requires imposing constraints on the architecture of the model itself. We choose to follow neither of these two roads, but instead to distill from z how much information related from $\hat{v}$ is filtering. Towards this end, we need to train, besides the task-related head $\mathcal{G}(z, \theta_{\mathcal{G}})$, another head $\mathcal{H}(z, \theta_{\mathcal{H}})$, which is trained to extract $\hat{v}$ from z, where $\theta_{\mathcal{G}}$ and $\theta_{\mathcal{H}}$ are the parameters associated to $\mathcal{G}$ and $\mathcal{H}$, respectively.

The output of $\mathcal{H}$ is the prediction over the feature willing to maintain private, thanks to which it is possible to estimate how much information it is possible to extract at the bottleneck z of the model through $\mathcal{I}(v, \hat{v})$. The overall gradient computation strategy for one iteration is presented in Algorithm 1. After obtaining y (line 3) and v (line 4) from the forward propagation, we can compute the three quantities of interest (to be minimized), namely:

- The loss $\mathcal{L}(y,\hat{y})$, to be minimized to train the model to learn the target task. This term is scaled by a positive hyper-parameter α and back-propagated through $\mathcal{F}(x,\theta_{\mathcal{F}})$ and $\mathcal{G}(z,\theta_{\mathcal{G}})$; hence, both the groups of parameters $\theta_{\mathcal{F}}$ (line 7) and $\theta_{\mathcal{G}}$ (line 5) will be updated to contribute towards the minimization of this term.
- The loss $\mathcal{L}(v,\hat{v})$, to be minimized to train the information removal head $\mathcal{H}$ to extract all the information about $\hat{v}$ from the bottleneck z. This term is back-propagated through $\mathcal{H}$ only (line 6): indeed, allowing the back-propagation also through $\mathcal{F}$ would favor the leakage of the information of $\hat{v}$ to z countering our whole purpose. On the contrary, the purpose of $\mathcal{H}$ is indeed to act as an estimator over the extractable information, which will be crucial for the correct estimation of the next term.
- The mutual information $\mathcal{I}(v,\hat{v})$, to be minimized to accomplish our purpose of erasing the information from the bottleneck z of the model. This term is scaled by a positive hyper-parameter γ, back-propagated through $\mathcal{H}$ and $\mathcal{F}$, but the gradients computed for $\mathcal{F}$ will be the only ones kept and maintained for the update step (line 7). It is of crucial importance not to use in the update step the gradient values in $\mathcal{H}$ as it will be in contrast with the learning problem as in the previous point (extracting the information about $\hat{v}$ from the bottleneck z).

A graphical representation of the working principles for IRENE is also visualized in Figure 2, where the colored arrows represent the back-propagation for the individual terms to be minimized, and the dashed arrow represents back-propagation without gradient computation.

3.3 Differentiable mutual information proxy

In the previous sub-section, we have presented our strategy to address the problem of minimizing the information of a target attribute $\hat{v}$ from the input x at the bottleneck z. This includes as well the minimization of the mutual information $\mathcal{I}(v,\hat{v})$.

According to the definition of mutual information, in order to estimate the joint probability, we should necessarily extract the predicted label $\tilde{v}$ with $\tilde{v} = \text{argmax}(v_i)$, and from this we can compute the joint $p_{\tilde{v}\hat{v}}$. Unfortunately, this operation is non-differentiable; hence, we need to provide a smooth, differentiable operator in its place. Towards this end, similarly to what is done for minimizing the cross-entropy loss, we substitute it with the softmax $\sigma(\cdot)$ which assigns a normalized score to the outputs of $\mathcal{H}$, compatible with our setup. At such a point, we can easily compute the differentiable proxy of the mutual information

$$\mathcal{I}(v,\hat{v}) = \sum_i \sum_j p_{\sigma(v)\hat{v}}(i,j) \log\left(\frac{p_{\sigma(v)\hat{v}}(i,j)}{p_{\sigma(v)}(i)p_{\hat{v}}(j)}\right). \tag{2}$$

Minimizing (2) with the update strategy described in Section 3.2 drives the output of $\mathcal{H}$ towards maximum confusion, making $\sigma(v)_i \rightarrow \frac{1}{C}\forall i$, where C is the number of classes for the information removal task.

It is worth noticing that, for our specific task, (2) can not be substituted with the maximization of the cross-entropy loss $\mathcal{L}(v,\hat{v})$. Let us assume we have $C=2$: under the assumption that the training is completed with success, the classifier will always make the wrong

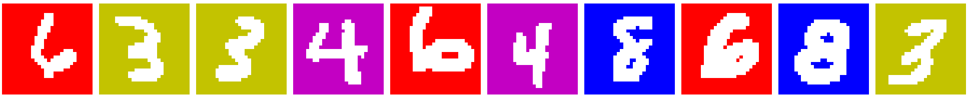


Figure 3: Example of images in the Biased-MNIST dataset. The background color correlates to a specific digit depending on the chosen value of ρ.

prediction, which in the binary case results in the maximum mutual information. This effect needs to be taken into account when applying debiasing strategies to pure biased (or in our case, private) feature removal [4, 7, 13, 30]. In the next section, we present our empirical evaluation.

4 Experiments

In this section, we present the experiments we conducted to test IRENE on-the-field. We perform our experiments on two datasets: Biased-MNIST (synthetic) and CelebA (a celebrity faces dataset). In all the presented experiments we will train, besides the main model composed of $\mathcal{F}$ and $\mathcal{G}$, the auxiliary head $\mathcal{H}$, attempting to extract the information willing to remove at the bottleneck z (which in our experiments we define as the layer before the output layer). Our training and inference algorithms are implemented in Python, using PyTorch 1.12 and a RTX3090 Ti NVIDIA GPU with 24GB of memory has been used for training and inference.[1]

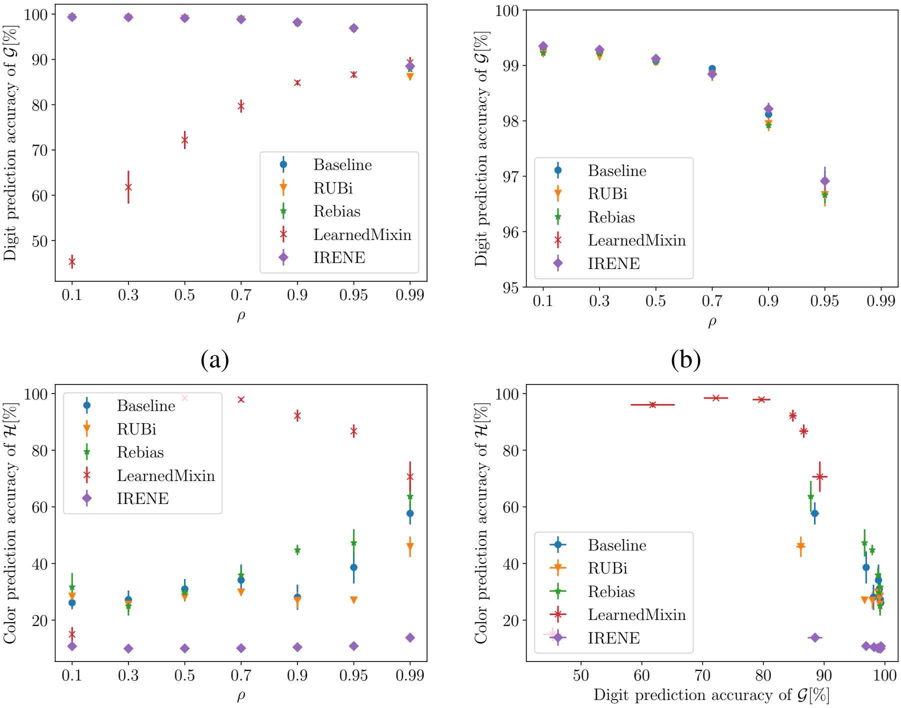
4.1 Experiments on BiasedMNIST

As a first benchmark, we employ the synthetic dataset Biased-MNIST, recently proposed by Bahng *et al.* [4]. This dataset is built on top of the broadly-known MNIST dataset [16], adding a background color as displayed in Figure 3. Its peculiarity relies on the possibility of correlating the background color to a specific digit with a hyper-parameter $\rho \in [0.1; 1.0]$ (where 0.1 corresponds to uncorrelated and 1.0 is a total correlation), which can be freely modified. Despite this dataset is mainly employed by the debiasing literature with the purpose of improving the digit classification performance on an uncorrelated test set, we will use it to test IRENE and its effect on both the information removal task (information about the background color) and the digit classification performance (target task), measured on the test set.

In order to investigate potential similarities and differences with some debiasing algorithms, we report as well results obtained with three debiasing techniques: RUBi [6], Rebias [4] and LearnedMixin [7].

Following [4], we train a convolutional architecture consisting of four 7×7 kernels, with ReLU activation and batch norm layers between convolution and activation. We have tested $\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9, 0.95, 0.99\}$ and averaged the results over 10 different runs. For all the experiments we have used SGD optimization for 80 epochs with a learning rate 0.1, decayed by a factor 0.1 at the milestones [40,60], weight decay 10^{-4}, and batch size 100. For IRENE we have set $\alpha = 0.5$ and $\gamma = 0.5$.

[1] The source code is available at https://github.com/enzotarta/irene.



(a)

(b)

(c)

(d)

Figure 4: Results on the BiasedMNIST dataset for different values of ρ. We report the performance of $\mathcal{G}$ (target task) as a function of ρ (the higher, the better) (a) and its zooming (b), the performance of $\mathcal{H}$ (the lower, the better) (c) and the performance of $\mathcal{H}$ as a function of the performance achieved by $\mathcal{G}$ on the target task (lower right corner, the better) (d).

Figure 4 reports the results of our experiments. Looking at the baseline result, we observe a physiological drop in the digit prediction accuracy as ρ increases (and consequently the increment of the color prediction accuracy), which is a marker of the use of the background color for the prediction. Interestingly, for the uncorrelated training set scenario ($\rho = 0.1$), the color prediction accuracy is above random guess (10%), meaning that there is, in any case, information leakage. IRENE performs very well, maintaining the baseline performances on the target task for all the tested values of ρ and maintaining the performance of $\mathcal{H}$ close to random guess. Similarly to what was observed by Barbano *et al.* [5] (despite in a different range of values for ρ), debiasing strategies allow information leakage, which is however employed in order to improve the performance of $\mathcal{G}$. In these experiments we observe, for the debiasing algorithms, performance close to the baseline as these algorithms focus on more extreme regimes (in these cases, higher ρ).

4.2 Experiments on CelebA

The CelebA dataset [19] has been designed for face-recognition tasks, providing 40 attributes for every image. The dataset contains a total of 202.6k images and, following the official train-validation split, we obtain 162.7k images for the training set, 19.9k images for the

Table 1: Results on the CelebA dataset. Here the gender is the information to erase.

Target	**Method**	**Prediction accuracy of $\mathcal{G}$ (trained task) [%](↑)**	**Gender prediction accuracy of $\mathcal{H}$ (information to remove) [%](↓)**
Blond hair	Baseline	95.34±0.07	84.32±2.76
	RUBi	95.29±0.14	88.55±1.22
	Rebias	**95.59±0.11**	88.50±3.78
	LearnedMixin	90.01±2.66	74.09±2.66
	IRENE ($\gamma = 0.1$)	95.37±0.10	55.47±8.18
	IRENE ($\gamma = 0.5$)	95.28±0.09	53.64±10.69
	IRENE ($\gamma = 1$)	95.24±0.29	**53.58±10.71**
Heavy makeup	Baseline	**90.58±0.14**	92.89±0.36
	RUBi	90.40±0.08	95.17±1.11
	Rebias	90.28±0.34	93.78±2.55
	LearnedMixin	84.88±3.28	68.09±10.55
	IRENE ($\gamma = 0.1$)	90.32±0.97	65.13±11.08
	IRENE ($\gamma = 0.5$)	85.66±2.80	56.45±9.46
	IRENE ($\gamma = 1$)	83.31±3.41	**51.98±9.56**
Eyeglasses	Baseline	99.67±0.02	69.51±4.33
	RUBi	99.64±0.01	59.21±5.22
	Rebias	99.65±0.01	76.61±8.21
	LearnedMixin	93.54±2.94	61.35± 3.67
	IRENE ($\gamma = 0.1$)	99.68±0.01	64.08±1.08
	IRENE ($\gamma = 0.5$)	**99.69±0.02**	61.57±6.95
	IRENE ($\gamma = 1$)	99.68±0.01	**54.66±12.58**

validation set, and 19.9k images for testing our models. For our training purposes, we use a ResNet-18 model backbone as $\mathcal{F}$.

The training has been performed using SGD, with an initial learning rate of 0.1, decayed by a factor of 10 after no improvement over the validation set loss has been detected for 10 consecutive epochs. The training stops when the learning rate drops below 10^{-3}. We use batch size 100 with a momentum of 0.9 and weight decay of 10^{-5}. Images are here re-scaled to 224×224 pixels. For IRENE, we use $\alpha = 0.5$ for all the experiments while $\gamma \in \{0.1, 0.5, 1\}$.

Differently from the debiasing literature, we are here not interested in testing the performance of a model trained over an unbalanced dataset, but we are here interested in observing the impact of some specific features we wish to keep private over the target task. Towards this end, we perform training over a *balanced* training set. We select here as classification target the recognition of the *eyeglasses*, *blond hair*, and *heavy makeup* attribute, while the gender is the information we wish to remove.

Results are reported in Table 1 and are averaged over 10 seeds. Here we observe different behaviors depending on the target task. Intuitively, the eyeglasses task (identifying the wearing of lenses) should be the one more “disentangled” by the gender information, and indeed, even with a high emphasis on the private feature removal (with the highest γ) the trained task performance remains close to the baseline. For the *blond hair* task (identifying people with blond hair), we observe that the performance on the target task remains close to the baseline but the information leakage on the gender is higher than for eyeglasses; however, even with a low γ, this information can be successfully removed. Finally, with *heavy* *makeup* (identifying people with heavy makeup), we observe the highest performance of $\mathcal{H}$ over all the tasks (in this case, even higher than $\mathcal{G}$), and removing this information worsens the performance on the target task. The information on gender is crucial to identify different makeup styles. From these three cases, we conclude that it is in general possible to remove information on a specific feature at the bottleneck of a deep model, but depending on the target task the performance might be impacted.

5 Conclusion and future work

In this work, we have proposed IRENE, a method to remove information at the bottleneck in deep neural networks. In particular, we train an auxiliary head classifier $\mathcal{H}$ which extracts the information on the "private" classes at the bottleneck z, and this is used to minimize directly at the bottleneck the mutual information between the ground truth of the information to remove and z.

We have tested IRENE on a synthetic dataset, BiasedMNIST, on 7 different regimes, and on CelebA, a dataset of faces of celebrities, with three different target tasks. For all the experiments IRENE can remove the information being asked to, but depending on the specific target task the performance might be affected. Interestingly, IRENE performs similarly to the tested debiasing algorithms in terms of target accuracy while removing the information to keep private. This suggests that extending IRENE to debiasing applications is a promising future research direction.

References

[1] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). *OJ*, L 119:1–88, 4.5.2016.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[3] Joshua Attenberg, Panos Ipeirotis, and Foster Provost. Beat the machine: Challenging humans to find a predictive model's "unknown unknowns". *Journal of Data and Information Quality (JDIQ)*, 6(1):1–17, 2015.

[4] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning (ICML)*, 2020.

[5] Carlo Alberto Barbano, Enzo Tartaglione, and Marco Grangetto. Bridging the gap between debiasing and privacy for deep learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3806–3815, 2021.

[6] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *Advances in neural information processing systems*, pages 841–852, 2019.

[7] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4067–4080. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1418. URL https://doi.org/10.18653/v1/D19-1418.

[8] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[9] John C Duchi, Michael I Jordan, and Martin J Wainwright. Privacy aware learning. *Journal of the ACM (JACM)*, 61(6):1–57, 2014.

[10] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, 2009.

[11] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3):17–51, 2016.

[12] P. Henriksen, Kerstin Hammernik, Daniel Rueckert, and Alessio Lomuscio. Bias field robustness verification of large neural image classifiers. In *BMVC*, 2021.

[13] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Arvindkumar Krishnakumar, Viraj Prabhu, Sruthi Sudhakar, and Judy Hoffman. Udis: Unsupervised discovery of bias in deep visual recognition models. In *BMVC*, 2021.

[15] Arvindkumar Krishnakumar, Viraj Prabhu, Sruthi Sudhakar, and Judy Hoffman. Udis: Unsupervised discovery of bias in deep visual recognition models. In *British Machine Vision Conference (BMVC)*, volume 1, page 3, 2021.

[16] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[17] Ren Li, Meng Zheng, Srikrishna Karanam, Terrence Chen, and Ziyan Wu. Everybody is unique: Towards unbiased human mesh recovery. In *BMVC*, 2021.

[18] Xiao Liu, Spyridon Thermos, Gabriele Valvano, Agisilaos Chartsias, Alison Q. O'Neil, and Sotirios A. Tsaftaris. Measuring the biases and effectiveness of content-style disentanglement. In *BMVC*, 2021.

[19] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[20] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. *arXiv preprint arXiv:1802.06309*, 2018.

[21] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. In *Advances in Neural Information Processing Systems*, 2020.

[22] Seyed Ali Osia, Ali Shahin Shamsabadi, Sina Sajadmanesh, A. Taheri, Kleomenis Katevas, H. Rabiee, N. Lane, and H. Haddadi. A hybrid deep learning architecture for privacy-preserving mobile analytics. *IEEE Internet of Things Journal*, 7:4505–4518, 2020.

[23] Prasanna Sattigeri, Samuel C Hoffman, Vijil Chenthamarakshan, and Kush R Varshney. Fairness gan. *arXiv preprint arXiv:1805.09910*, 2018.

[24] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

[25] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 587–601. ACM, 2017.

[26] Sho Sonoda and Noboru Murata. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268, 2017.

[27] Sruthi Sudhakar, Viraj Prabhu, Arvindkumar Krishnakumar, and Judy Hoffman. Mitigating bias in visual transformers via targeted alignment. In *BMVC*, 2021.

[28] Enzo Tartaglione and Marco Grangetto. Take a ramble into solution spaces for classification problems in neural networks. In *International conference on image analysis and processing*, pages 345–355. Springer, 2019.

[29] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13508–13517, 2021.

[30] William Thong and Cees G. M. Snoek. Feature and label embedding spaces matter in addressing image classifier bias. In *BMVC*, 2021.

[31] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015.

[32] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[33] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE, 2018.