Visual Recognition with Deep Learning from Biased Image Datasets

Robin Vogel  
School of Informatics  
University of Edinburgh  
Edinburgh, United Kingdom  
robin.vogel@ed.ac.uk

Stephan Clémençon  
LTCI, Institut Polytechnique de Paris  
Telecom Paris  
Palaiseau, France  
stephan.clemenccon@telecom-paris.fr

Pierre Laforgue  
Department of Computer Science  
Università degli Studi di Milano  
Milano, Italy  
pierre.laforgue@unimi.it

Abstract—In practice, and more especially when training deep neural networks, visual recognition rules are often learned based on various sources of information. On the other hand, the recent deployment of facial recognition systems with uneven predictive performances on different population segments highlights the representativeness issues possibly induced by a naive aggregation of image datasets. Indeed, sampling bias does not vanish simply by considering larger datasets, and ignoring its impact may completely jeopardize the generalization capacity of the learned prediction rules. In this paper, we show how biasing models, originally introduced for nonparametric estimation in [1], and recently revisited from the perspective of statistical learning theory in [2], can be applied to remedy these problems in the context of visual recognition. Based on the (approximate) knowledge of the biasing mechanisms at work, our approach consists in reweighting the observations, so as to form a nearly debiased estimator of the target distribution. One key condition for our method to be theoretically valid is that the supports of the distributions generating the biased datasets at disposal must overlap, and cover the support of the target distribution. In order to meet this requirement in practice, we propose to use a low dimensional image representation, shared across the image databases. Finally, we provide numerical experiments highlighting the relevance of our approach whenever the biasing functions are appropriately chosen.

Index Terms—Sampling bias, selection effect, visual recognition, reliable statistical learning

I. INTRODUCTION

Besides the considerable advances in memory technology and in the computational power of machines, which now permit to implement optimization programs that operate on vast classes of rules, making deep learning feasible, the spectacular rise in performance of visual recognition algorithms is essentially due to the recent availability of massive labeled image datasets. However, due to the poor control of the data acquisition process or even the absence of any experimental design to collect the datasets, the distribution of the training examples may drastically differ from that of the data to which the predictive rule will be applied when deployed. The adverse effects of machine-learning algorithms for facial recognition when trained from data suffering from selection bias have been recently highlighted in various works, as recalled in example 1 below.

This research was partially supported by the chair “Good In Tech”, see acknowledgements for more details.

Example 1 (Bias in facial recognition): Most publicly available large face databases — such as [3] or [4] for instance — are composed of images of celebrities and do not represent appropriately the global population, with respect to ethnicity and gender in particular. In [5], face recognition systems learned from such databases are argued to suffer from an “other-race effect”, i.e. that they fail to distinguish individuals from unfamiliar ethnicities, like humans often do. Recently, industrial benchmarks for commercial face recognition algorithms highlighted the discrepancies in accuracy across race and gender and their fairness is now questioned, see [6].

In [7], a balanced dataset over ethnicities built by discarding selected observations from a larger dataset has been proposed so as to cope with the representativeness issue mentioned in example 1. However, reducing the number of training samples may strongly damage performance and generalization for most computer vision tasks, see e.g. [8].

The purpose of this paper is to investigate how to use several image datasets, which are each biased with respect to the target distribution in a possibly different manner, for learning visual recognition systems, when knowing approximately the corresponding bias functions. Whereas learning from training data that are not distributed or of the same format as the test data is a very active research topic in machine learning literature, see e.g. [9], [10] and in computer vision especially, with a plethora of transfer learning methods (e.g. [11]) or domain adaptation techniques (e.g. [12]), the approach considered in this article is of different nature and relies on general user-defined biasing functions to relate the training datasets to the testing distribution for computer vision tasks. In example 1, the biasing functions would have the gender and nationality information usually present in identification databases as arguments for instance. However, the approach promoted here is very general and applies when specific types of selection bias are present in the training databases supposedly available and are approximately known. It is based on recent theoretical results on re-weighting techniques for learning from biased training samples, documented in [2] (see also [11]) and [13].

The notion of bias is used with various different meanings. For instance in example 1, it is understood as a racial prejudice, due to the possibly great disparity between the
Debiasing machinery can be used for visual recognition tasks and illustrates its relevance through various experiments on the well-known datasets CIFAR10 and CIFAR100.

The rest of the paper is organized as follows. Related works are collected in section II, while the methodology considered here for de-biasing image datasets is described at length in section III. In section IV, experimental results are displayed and discussed, providing empirical evidence of the advantages of the approach promoted here. Finally, some concluding remarks are gathered in section V.

II. RELATED WORK - STATE OF THE ART

Learning visual recognition models in presence of bias has been recently the subject of attention in the computer vision literature. Recently, authors explicitly referred to the notion of bias. For example, [18] proposed to correct for gender bias in captioning models by using masks that hide the gender information, or [19] learned to dissociate an objective classification task to some associated confounding bias using a known prior on the bias.

Bias and Transfer Learning. However, since we considered bias as any significant difference between the training and testing distributions, learning with biased datasets is related to transfer learning. We introduce by \( \mathcal{X} \) (resp. \( \mathcal{Y} \)) the input (resp. output) space associated to a computer vision task. [20] define domain adaptation as the specific transfer learning setting where both the task — e.g. object detection or visual recognition — and the input space \( \mathcal{X} \) are the same for both training and testing data, but the train and test distribution are different, i.e. \( p_{\text{train}}(x, y) \neq p_{\text{test}}(x, y) \) for some \( (x, y) \in \mathcal{X} \times \mathcal{Y} \). We denote by \( z \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y} \) any tuple \( (x, y) \in \mathcal{X} \times \mathcal{Y} \). Our work makes similar assumptions as domain adaptation, but considers several training sets.

Transferable Features. [21], [12] address domain adaptation by learning a feature extractor \( g \) that is invariant with respect to the shift between the domains, which makes the approach well suited to problems where only the marginal distribution \( p(x) \) over \( \mathcal{X} \) is different between training and testing. Formally, under this assumption the posterior distributions satisfy \( p_{\text{train}}(y \mid g(x)) = p_{\text{test}}(y \mid g(x)) \) for any \( (x, y) \in \mathcal{X} \times \mathcal{Y} \) and we have \( p_{\text{train}}(x) \neq p_{\text{test}}(x) \) for some \( x \in \mathcal{X} \). Consequently, their approach is well-suited when the difference is reduced to that of the “low-level” visual appearance of the images (see fig. 1). However, [22] considers domains as types of images (e.g. medical, satellite), which makes learning invariant deep features undesirable. In the context of example I, learning deep invariant features would discard information that is relevant for identification. For those reasons, [22] as well as [23] considers all networks layers as possible candidates to adaptation. Taking a different approach, [11], [14], [24] all explicitly model different train and test posterior probabilities \( p(y \mid x) \) for the output, for example through specialized classifiers for both the target and source dataset, see e.g. [24]. Most of the work on deep transfer focuses on adapting feature representations, but...
another popular approach in transfer learning is to transfer knowledge from instances [20].

**Instance Selection.** Recent computer vision papers proposed to select specific observations from a large scale training dataset to improve performance on the testing data. Precisely, [25] selects observations that are close to the target data, as measured by a generic image encoder, while [26] uses a similarity based on low-level characteristics. More similar to our work, [27], [24] use importance weights for training instances, that are computed from a small set of observations that follow the target distribution. Precisely, [27] transfers information from the predicted distribution over labels of the adapted network, while [24] learns with a reinforcement learning selection policy guided by a performance metric on a validation set. The objective of the importance function is to reweight any training instance \((x, y)\) by a quantity proportional to the likelihood ratio \(\Phi(x, y) := \frac{p_{\text{test}}(x, y)}{p_{\text{train}}(x, y)}\). While most papers assume the availability of a sample that follows \(p_{\text{test}}\) to approximate the likelihood ratio \(\Phi\), recent work in statistical machine learning proposes to introduce simple assumptions on the relationship between \(p_{\text{train}}\) and \(p_{\text{test}}\) [13], [2]. These assumptions require the machine learning practitioner to only have some *a priori* knowledge on the testing data, which happens naturally in many situations, notably in industrial benchmarks. In the context of example 1, facial recognition providers often have accurate knowledge of some characteristics (e.g. nationality) of the target distribution but no access to the target data. [13] assumed that a single training sample covers the whole testing distribution (with a dominated measure assumption), but [2] relaxes that assumption by considering several source datasets.

**Multiple Source Datasets.** Learning from several source datasets has been studied in various contexts, such as domain adaptation [28], [10], or domain generalization [16]. In the latter work, authors optimize for all sources simultaneously, and find an interesting trade-off between the different competing objectives. Our approach is closer to the one developed in [2], which consists in appropriately reweighting the observations generated by some biased sources. Under several technical assumptions, the authors have extended the results of [17] and have shown that the unbiased estimate of the target distribution thus obtained can be used to learn reliable decision rules. Note however that an important limitation of these works is that the biasing functions are assumed to be exactly known. In this work, we learn a deep neural network for visual recognition with sampling weights derived from [2], using approximate expressions for the biasing functions. Note that [19] also assumes the knowledge of bias information to adjust a neural network. However, their approach is pretty incomparable to ours, insofar as it consists in penalizing the mutual information between the bias and the objective.

### III. Debiasing the Sources - Methodology

In this section, we formally introduce our methodology to learn reliable decision functions in the context of biased image-generating sources. In Section III-A, we first recall the debiased Empirical Risk Minimization framework developed in [2], which our approach builds upon. In Section III-B, we then highlight that applying this theory to image datasets raises important practical issues, and propose heuristics to bypass these difficulties. In what follows, we use \(\mathbb{I}\{\cdot\}\) to denote the indicator of an event.

#### A. Debiased Empirical Risk Minimization

Recall that \(z = (x, y)\) denotes a generic random variable with distribution \(p_{\text{test}}\) over the space \(\mathcal{Z} = \mathcal{X} \times \mathcal{Y}\). We consider the supervised learning framework, meaning that \(x\) represents some input information supposedly useful to predict the output \(y\). Given a loss function \(\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+\), and a hypothesis set \(\mathcal{H} \subset \mathcal{Y}^\mathcal{X}\) (e.g., the set of decision functions possibly generated by a fixed neural architecture), our goal is to compute

\[
h^* = \arg\min_{h \in \mathcal{H}} \mathbb{E}(x, y) - p_{\text{test}}[\ell(h(x), y)]. \tag{1}
\]

As \(p_{\text{test}}\) is unknown in general, a usual hypothesis in statistical learning consists in assuming that the learner has access to a dataset composed of \(n \in \mathbb{N}\) independent realizations generated from \(p_{\text{test}}\). However, as highlighted in the introduction section, this i.i.d. assumption is often unrealistic in practice. In this paper, we instead assume that the observations at disposal are generated by several biased sources. Formally, let \(K \in \mathbb{N}\) be the number of sources, and for \(k \in \{1, \ldots, K\}\), let \(\mathcal{D}_k = \{z_1^{(k)}, \ldots, z_{n_k}^{(k)}\}\) be a dataset of size \(n_k\) composed of independent realizations generated from biased distributions \(p_k\). We also define \(n = \sum_{k=1}^K n_k\), and \(\lambda_k = n_k/n\). The distributions \(p_k\) are said to be biased, as we assume the existence of (known) biasing functions \(\omega_k: \mathcal{Z} \rightarrow \mathbb{R}_+\) such that for any \(z \in \mathcal{Z}\) we have

\[
p_k(z) = \frac{\omega_k(z)}{\Omega_k} p_{\text{test}}(z), \tag{2}
\]

where \(\Omega_k = \int \omega_k(z) d p_{\text{test}}(z)\) is an (unknown) normalizing factor. This data generation design is referred to as Biased Sampling Models (BSMs), and was originally introduced in [17] and [1] for asymptotic nonparametric estimation of cumulative distribution functions. Note that BSMs are a strict generalization of the standard statistical learning framework, insofar as the latter can be recovered as a special case with the choice \(K = 1\) and \(\omega_1(z) = \mathbb{I}\{z \in \mathcal{Z}\}\). For general BSMs, it is important to observe that performing naïvely Empirical Risk Minimization (ERM, see e.g., [29]), which means concatenating the \(n\) observations without considering the sources of origin and computing

\[
\hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{k=1}^K \sum_{i=1}^{n_k} \ell(h(x_i^{(k)}), y_i^{(k)}), \tag{3}
\]

might completely fail. Indeed, minimizing criterion (3) instead of (1) amounts to replace \(p_{\text{test}}\) with the empirical distribution

\[
\hat{p}(z) = \frac{1}{n} \sum_{k=1}^K \sum_{i=1}^{n_k} \mathbb{I}\{z = z_i^{(k)}\} = \sum_{k=1}^K \lambda_k p_k(z),
\]
Fig. 2. Given the datasets $D_k$, it is easy to compute (unbiased) estimates of the $p_k$, through the empirical distributions $p̂_k(z) = (1/n_k)\sum_{i=1}^{n_k} \mathbb{1}\{z = z_i^{(k)}\}$. Since the $ω_k$ are assumed to be known, one can construct $p̂_k/ω_k$, normalize the distribution (to mimic the effect of $Ω_k$ which is unknown), and obtain an estimate of $p_{test}$ restricted to the support $S_k$ of $p_k$. But if the union of $S_1$ and $S_2$ is strictly included in $S_{test}$, as in case a), it is impossible to estimate $p_{test}$ on $S_{test}(S_1 ∪ S_2)$. This is expected, as the learner does not have access to any observation valued in this part of the space. If $S_{test}$ is included in $S_1 ∪ S_2$, but there is no overlap, as in case b), we have estimates of $p_{test}$ on the entire support, but it is impossible to find the right normalization to ensure that the combination of $p̂_{1test}$ and $p̂_{2test}$ is a good estimate of $p_{test}$. More precisely, $0.1p̂_{1test} + 0.9p̂_{2test}$ and $0.9p̂_{1test} + 0.1p̂_{2test}$ are two distributions defined on $S_{test}$ that are equal to $p̂_{test}$ if restricted to $S_k$, but from the data available in scenario b) there is no way to know which one is closer to $p_{test}$. However, if there is overlap, as in case c), the knowledge of the $ω_k$ and the observations present in the intersection allow to estimate the relative normalization, and to produce an almost unbiased estimate of $p_{test}$.

\[
p̂_k(z) = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbb{1}\{z = z_i^{(k)}\}.
\]

However, it can be seen from the above equations that $p̂$ is rather an approximation of distribution $p := \sum_k λ_k p_k$, which might be very different from $p_{test}$ (e.g., in the extreme case where $λ_1 = 1$ and $λ_k = 0$ for $k \neq 1$, we have $p̂ = p_1 \neq p_{test}$).

To remedy this problem, [2] has developed a debiased ERM procedure for BSMs, under the following two assumptions: (1) the union of the supports of the $p_k$ must contain the support of $p_{test}$, and (2) the supports must sufficiently overlap. See Figure 2 for an intuitive explanation of the assumptions. Formally, the support assumption ensures that the $ω_k$ cannot be null all at the same time, and thus allows to invert the following equality:

\[
\bar{p} = \sum_{k=1}^{K} λ_k p_k = \sum_{k=1}^{K} \frac{λ_k ω_k}{Ω_k} p_{test},
\]

\[
p_{test} = \left( \sum_{k=1}^{K} \frac{λ_k ω_k}{Ω_k} \right)^{-1} \bar{p}.
\]

As noticed earlier, it is straightforward to obtain an estimate of $\bar{p}$, through $\bar{p} = \sum_k λ_k p̂_k$. The $ω_k$ being assumed to be known, it is then enough to find estimates of the $Ω_k$, and to plug them into (4), to construct an (almost unbiased) estimate of $p_{test}$. As shown in [1], such estimates $W_k$ can be easily computed by solving a system of $K$ equations through a Gradient Descent approach, see the Appendix for technical details. Let $\bar{p}$ be the distribution obtained by replacing $\bar{p}$ and $Ω_k$ in (4) with $\bar{p}$ and $W_k$ respectively. Debiased ERM consists in approximating $p_{test}$ in (1) with $\bar{p}$. It can be shown that it is equivalent to compute

\[
\hat{h} = \arg\min_{h ∈ H} \sum_{k=1}^{K} \sum_{i=1}^{n_k} π_{kli} \ell(h(x_i^{(k)}), y_i^{(k)}),
\]

where

\[
π_{kli} = \left( \frac{\sum_{i=1}^{K} λ_{iwin} (z_i^{(k)})}{W_i} \right)^{-1} \left[ \frac{\sum_{j=1}^{m} λ_{jwin} (z_j^{(m)})}{W_i} \right].
\]

Hence, once the debiasing weights $π_{kli}$ are derived, this approach is not more computationally expensive than standard ERM. Note also that most modern machine learning libraries allows to weight the training observations during the learning stage, e.g., through the sample_weight keyword argument in the fit method for scikit-learn [30].

B. Application to Image-Generating Sources

Thanks to its generality, the debiased ERM procedure allows to model realistic scenarios, which make its application to image-generating sources particularly relevant. First, the biasing functions $ω_k$ apply to the entire observation $z = (x, y)$. Then, one can address biases induced by the measurement devices (cameras with different intrinsic characteristics are expected to produce photos $x$ with different film grains), from the subjects (cameras located in different areas of the world are expected to observe different classes of objects/animals $y$), or both at the same time. This rich framework generalizes Covariate Shift, see e.g., [31], [32], a popular bias assumption which only allows the marginal distribution of $x$ to vary across the datasets, the conditional probability $p(y | x)$ being assumed to remain constant equal to $p_{test}(y | x)$. A second advantage is that an individual biasing function $ω_k$ might be null on some part of $Z$. This allows for instance to consider biasing functions of the form $ω_k(z) = \mathbb{1}\{z ∈ Z_k\}$, where $Z_k$ is a subspace of $Z$, accounting for the fact that dataset $D_k$ is actually sampled from a subpopulation (e.g., regional animal species). In this case, Importance Sampling typically fails, as the importance weights are given by $p_{test}(z)/p_k(z) x 1/ω_k(z)$, which explodes for $z ≠ Z_k$. In contrast, debiased ERM combines the different datasets in a clever fashion to produce...
an estimator valid on the entire set $\mathcal{Z}$. The final asset of debiased ERM is undeniably the theoretical guarantees that come with it. Indeed, it has been shown in [2] that, up to constant factors, $\hat{h}$ has the same excess risk as an empirical risk minimizer that would have been computed on the basis of an unbiased dataset of size $n$. This remarkable property is however conditioned upon two relatively strong requirements, that we will study and relax in the context of image-generating sources.

First, the biasing functions $\omega_k$ have to be known. Recall nonetheless that this assumption is less compelling that it might look like at first sight. Indeed, only the $\omega_k$, and not the normalizing factors $\Omega_k$, are assumed to be known. Consider again the case where $\omega_k(z) = \mathbb{I}\{z \in \mathcal{Z}_k\}$, with $\mathcal{Z}_k$ a subspace of $\mathcal{Z}$. The assumption requires to know $\omega_k$, or equivalently the subpopulation $\mathcal{Z}_k$, but not to know the relative size of this subpopulation among the global one with respect to $p_{\text{test}}$, i.e., $\int_{\mathcal{Z}} dp_{\text{test}}(z) = \int_{\mathcal{Z}} \mathbb{I}\{z \in \mathcal{Z}_k\} dp_{\text{test}}(z) = \Omega_k$, which is much more involved to obtain in practice. Still, except in very particular cases such as randomized control trials, it is extremely rare to have an exact knowledge of the $\omega_k$. In the following, we conduct numerical experiments where we show that the biasing functions can actually be learned along the training process, without significantly degrading the empirical performances. Our approach thus provides an end-to-end methodology to debias image-generating sources, which only relies on the biased databases, does not use any oracle quantities (the $\omega_k$), and can therefore be fully implemented in practice.

The second restriction regards the overlapping. Indeed, an important caveat of the analysis carried out in [2] is that constant factors depend heavily on the overlapping. Formally, let

$$\kappa = \inf_{(k,l), \omega_k \text{ and } \omega_l \text{ overlap}} \int_{\mathcal{Z}} \mathbb{I}\{\omega_k(z) > 0\} \cdot \mathbb{I}\{\omega_l(z) > 0\} dp_{\text{test}}(z),$$

and

$$m = \min_{k < K} \inf_{z \in \mathcal{Z}: \omega_k(z) > 0} \omega_k(z).$$

These two parameters quantify the overlapping, as shown in Figure 3. Results in [2], see e.g., Proposition 1 therein, typically hold with probability $1 - e^{-cn}$, where $c$ is a constant proportional to $\kappa^2 m^2$. As previously discussed, we recover that theoretical guarantees are conditioned upon overlapping: when $\kappa = 0$ or $m = 0$, the bound obtained is vacuous. But the formula actually suggests an even more interesting behavior. Besides the 0 threshold, we can see that the more distributions overlap, i.e., the bigger $\kappa$ and $m$ are, the better the guarantees. On the contrary, if input observations are images, i.e., high dimensional objects whose distributions have non overlapping supports, it is likely that constants $\kappa$ and $m$ become too small to provide a meaningful bound. To remedy this practical issue, we propose to use biasing functions which take as input a low-dimensional embedding of the images (or the label $y$) in order to maximize the overlapping and thus ensure good constant factors. We also conduct several sensitivity analyses, which corroborate the fact that increasing the density overlapping brings stability to the procedure, and globally enhances the performances.

IV. EXPERIMENTS

We now present experimental results on two standard image recognition datasets, namely CIFAR10 and CIFAR100, from which we extract $K$ biased datasets, so as to recreate the framework of Section III-A in a controllable way. The great generality of this setting allows us to model realistic scenarios, such as the following. An image database is actually composed of pictures that have been collected in several goes, by cameras located in different parts of the world. One may think about animal pictures gathered by expeditions in different countries, or security cameras located in different areas for instance. In this case, the location bias then translates into a class bias, and the class proportions greatly differ from one database to the other (we do not observe the same species of animals in Africa or in Europe). We show that our method allows to efficiently approximate the biasing functions in this case, and provides a satisfactory end-to-end solution to this common bias scenario. A second typical bias scenario occurs when the images originate from cameras with different characteristics, and thus exhibit different film grains. We show that a bias model on a low representation of the images (therefore avoiding the overlapping difficulty) is able to model such phenomenon, and that our approach provides satisfactory results despite the bias mechanism at work.

**Imbalanced learning** is a single dataset image recognition problem, in which the training set is assumed to contain the classes of interest in a proportion different from the one in the validation set (usually supposed balanced). Long-tail learning, a special instance of imbalanced learning, has recently received a great deal of attention in the computer vision community, see e.g., [33]. In Section IV-B, we propose a multi-dataset of imbalanced learning, where we assume that each database $\mathcal{D}_k$ contains a different proportion of the classes of interest, this proportion being possibly equal to 0 for certain classes (which is typically not allowed in single dataset imbalanced learning). Under the assumption that the classes are balanced in the validation dataset, we show that $\tilde{\omega}_k(z) = \tilde{\omega}_k(x, y) = n_k^{(y)}$, where $n_k^{(y)}$ is the number of
observations with label \( y \) in dataset \( D_k \), approximates well the bias function. The good empirical results we obtain provide a proof of concept for our methodology in image recognition problems where the marginal distributions over classes differ between the data acquisition phase and the testing stage. This scenario is often found in practice, e.g., as soon as cameras are located in different places of the world and thus record objects in proportions which are specific to their locations (more cars in the city, more trees in the countryside, ...). Note that our goal here is not to achieve the best possible accuracy, otherwise knowing the test proportions is enough to reweight the observations accurately, but rather to show that our method achieves reasonable performances, even without knowledge of the \( \omega_k \). We also conduct a sensitivity analysis, showing that increasing the overlapping indeed improves stability and accuracy.

In section IV-C, we assume that the databases \( D_k \) are collected under different acquisition conditions (e.g., quality of the camera used, scene illumination, ...), thus generating different types of images. To approximate the bias function, we first embed the images into a small space where the different types are easily separable. Then, we use the available instances in \( D_k \) to estimate the boundaries between the different image types. This experimental design complement the previous one, as bias now apply to (a transformation of) the inputs \( x \), and not on the classes \( y \), thus highlighting the versatility of our approach in terms of bias that can be modeled. In our experiments, we consider a scenario where images have distinct backgrounds, but of course the debiasing technique could be applied to any other intrinsic property of the images, such as the illumination of the scene.

A. Experimental details

CIFAR10 and CIFAR100 are two standard image recognition datasets, that contain both 50K training images and 10K testing images, of resolution 32 \( \times \) 32 pixels. CIFAR10 is a 10-way classification dataset, while CIFAR100 is a 100-way classification dataset. Both the training sets and the testing sets are balanced in terms of the number of images per class. CIFAR10 (resp. CIFAR100) thus contains 5K (resp. 500) images per class in the training split and 1K (resp. 100) images per class in the testing split. Our experimental protocol consists in four steps:

1) we create the biased datasets \( D_1, \ldots, D_K \) by sampling observations with replacement (and according to the \( \omega_k \)) from the train split of CIFAR10/100,
2) we construct estimates \( \hat{\omega}_k \) of the biasing function using the information contained in the datasets \( D_1, \ldots, D_K \), and possibly some additional information on the testing distribution if available,
3) we compute the \( \hat{W}_k \) according to the procedure detailed in the Appendix, and derive the debiasing weights \( \pi_{ki} \), see Equation (5) where we use \( \hat{\omega}_k \) instead of \( \omega_k \),
4) we learn a network from scratch using the debiasing weights calculated in step 3.

Steps 1 and 2 are setting-dependent, as they use respectively an exact or incomplete knowledge about the bias mechanism at work. The exact expression is used in step 1 to generate the training databases from the original training split. It is however not used in the subsequent learning procedure. The approximate expression serves as a prior in step 2 to produce the estimates \( \hat{\omega}_k \). The latter are typically determined by estimating from the databases \( D_1, \ldots, D_K \) the value of an unknown parameter in the approximate expression. Choices for the approximate expression can be motivated by expert knowledge, possibly combined with a preliminary analysis of the biased databases. We provide a detailed description of steps 1 and 2 for each experiment in their respective sections.

In step 3, we compute the \( \hat{W}_k \), which according to the computations in the Appendix amounts to minimize over \( \mathbb{R}^K \) the convex function \( \hat{D}(u) \) defined in Equation (13). As \( \hat{D} \) is separable in the observations, we use the pytorch [34] implementation of Stochastic Gradient Descent (SGD) for 4000 iterations, with batch size 100, fixed learning rate \( 10^{-2} \), and momentum 0.9.

In step 4, we train the ResNet56 implementation of [35] for 205 epochs using SGD with fixed initial learning rate \( 10^{-1} \) and momentum 0.9, where we divide the learning rate by 10 at epoch 103 and 154. Our implementation is available at https://tinyurl.com/5ahadh7c.

B. Class Imbalance Scenario

In this section, we assume that the conditional probability of \( x \) given \( y \) is the same for all the distributions \( p_1, \ldots, p_K \). Formally, for all \( y \in \mathcal{Y} \), and any \( k \in \{1, \ldots, K\} \), we have \( p_k(x \mid y) = p_{\text{test}}(x \mid y) \). However, the class probabilities \( p_k(y) \) may vary from one distribution to the other, and we recover the link to imbalanced learning. This is a common scenario, already studied in [13] for instance. In this context, it can be shown that the biasing function on \( z = (x, y) \) is proportional to the likelihood ratio between the class proportion of the biased dataset \( p_k(y) \) and that of the testing distribution \( p_{\text{test}}(y) \). Indeed, using (2) we obtain

\[
\omega_k(x, y) = \Omega_k \frac{p_k(x, y)}{p_{\text{test}}(x, y)} = \Omega_k \frac{p_k(y)}{p_{\text{test}}(y)}.
\]

Since we assume the classes to be balanced in the test set, we also have \( p_{\text{test}}(y) \equiv 1/|\mathcal{Y}| \). It then follows from (6) that knowing \( p_k(y) \) gives \( \omega_k(z) \) up to a constant. Similarly, estimating \( p_k(y) \) based on \( D_k \) also provides an estimate of \( \omega_k \). Although our experiment considers the classes associated to the datapoints as subjects of the bias, the knowledge of any discrete attribute of the data can be used for debiasing as long as both: 1) realizations of that attribute are known for the training set, and 2) its marginal distribution on the testing distribution is known. Those additional discrete attributes are called strata in [13].

Generation of the biased datasets. Let \( M = |\mathcal{Y}| \), and choose \( K \) which divides \( M \) (here we take \( K = 5 \)). We split the original train dataset into \( K \) subsamples with different class proportions as follows. We first partition the classes into
K meta-classes $C_1, \ldots, C_K$, such that each $C_k$ contains $n_c := M/K$ different classes. In the case of CIFAR-10 we have $C_1 = \{0, 1\}, C_2 = \{2, 3\}, \ldots, C_5 = \{8, 9\}$. Then, we consider that each dataset $D_k$ is composed of classes in meta-classes $C_k$ and $C_{k+1}$ only, and that classes coming from the same meta-class are in the same proportion. Formally, for $k \in \{1, \ldots, K\}$, we have:

$$p_k(y) = \begin{cases} \frac{n_c}{n_k} & \text{if } y \in C_k \\ \frac{n_c}{n_k} & \text{if } y \in C_{k+1} \\ 0 & \text{otherwise} \end{cases}$$

with the convention $C_{K+1} = C_1$, and where $\gamma \in [0, 0.5]$ is an overlap parameter. Note that the concatenation of the datasets $D_1, \ldots, D_K$ has the same distribution over the classes (in expectation) as the original unbiased training dataset. A classifier learned on this concatenation will therefore serve as a reference. Note also that some observations of the original dataset may not be present in the resampled dataset, which is a consequence of sampling with replacement. For $K = 5$, we illustrate the distribution over classes of all datasets $D_1, \ldots, D_5$ for different values of $\gamma$ in Figure 4.

**Approximating the biasing functions.** Let $n_k^{(y)}$ be the number of observations with class $y$ in database $D_k$. A natural approximation of $p_k(y)$ is $n_k^{(y)}/n_k$. Furthermore, with the above construction of the biased datasets, we know that $\Omega_k$ and $n_k$ are the same for all $k \in \{1, \ldots, K\}$, and that $p_k(y) = 1/M$. Plugging these values in Equation (6), and since it is enough to know the $\omega_k$ up to a constant multiplicative factor (see Section III-A), a natural approximation is

$$\hat{\omega}_k(y) := n_k^{(y)}.$$  

We then solve for the debiasing sampling weights $\pi_{k,i}$ by plugging the approximations $\hat{\omega}_k$'s into the procedure described in section III.

**Analysis of the debiasing weights.** In Figure 5, we plot the distribution over classes of the datasets weighted by the debiasing procedure and compare it to the test distribution over classes, for 8 different runs of the debiasing procedure. We observe that the distributions are significantly further from the test distribution and more noisy as $\gamma$ decreases. If we concatenate the databases $D_1, \ldots, D_K$ into a single training dataset, i.e., set $\pi_{i,k}$ constant, then the concatenated dataset is equal to the original training dataset of CIFAR in expectation with respect to our synthetic biased generation procedure. Therefore, it is desirable to obtain uniform weights after solving for the debiasing procedure. In Figure 6, we plot the $L_2$ distance between the obtained sampling weights and the uniform weights, as a function of $\gamma$. The result shows that the distance from the optimal weights is higher when $\gamma$ is low and lower when $\gamma$ is high, confirming the overlap assumptions introduced in Section III.

**Empirical results.** We provide testing accuracies for several $\gamma$’s in table I. A slight performance decrease is observed in both cases, stronger in the case of CIFAR100. We see that the predictive performance increases as $\gamma$ grows, while our estimate over 8 runs of the variance of the final accuracy decreases. Our results confirm empirically the necessity to have a significant enough overlap between the databases for our debiasing procedure.

**C. Image Acquisition Scenario**

In many cases, the bias does not originate from class imbalances, but from an intrinsic property of the images, such as the illumination of the scene or the background. In this image acquisition scenario, we simulate the effect of learning with datasets collected on different backgrounds.

**Generation of the biased datasets.** In the datasets CIFAR10 and CIFAR100, the background varies significantly between images. Some images feature a textured background while others have an uniform saturated background. We generate the biased datasets by grouping the images into bins derived from the average parametric color value in normalized HSV representation of a 2 pixel border around the image. For an image $x$, we denote this 3-dimension representation of an image by $\bar{x} = (\bar{x}_H, \bar{x}_S, \bar{x}_V) \in [0, 1]^3$. Each coordinate in this 3-dimensional space is then split using the median values $h_H, h_S, h_V$ of the encodings $\bar{x}$ of the training dataset, giving 8 bins $B_0, \ldots, B_7 \subset [0, 1]^3$ to split the training dataset. The background is then sampled from one of these bins with a probability that decreases as the distance to the middle $\bar{x}$ increases. The bias is then added to the image. In many cases, this approach is sufficient to introduce the desired bias.

## Table I

| Dataset | $\gamma$ | Acc. |
|---------|---------|------|
| CIFAR10 | $10^{-3}$ | 89.96 (±2.15) |
|         | $10^{-2}$ | 90.51 (±1.72) |
|         | 0.10     | 90.97 (±1.64) |
|         | 0.20     | 91.02 (±1.64) |

| Dataset | $\gamma$ | Acc. |
|---------|---------|------|
| CIFAR100 | $10^{-3}$ | 62.00 (±0.73) |
|          | $10^{-2}$ | 63.66 (±3.66) |
|          | 0.10     | 65.08 (±0.74) |
|          | 0.20     | 64.96 (±1.96) |
images of CIFAR into 8 datasets $D_1, \ldots, D_8$. Precisely, for any $l \in \{0, \ldots, 7\}$, if $(b_H, b_S, b_V) \in \{0, 1\}^3$ is the binary representation of $l$, i.e., $l = 4b_H + 2b_S + b_V$, then

$$B_l = \{ \bar{x} \in [0, 1]^3 : \forall C \in \{H, S, V\}, \quad (\bar{x}_C - b_C) (2b_C - 1) \geq 0 \}.$$  

To satisfy the overlap condition, we define the biasing function $\omega_k(x)$ for each image $x$ and any $k \in \{1, \ldots, 8\}$, as

$$\omega_k(x) = \begin{cases} 1 & \text{if } \bar{x} \in B_k, \\ \max (0, 1 - \gamma^{-1} \| \bar{x} - \Pi_1(\bar{x}, B_k) \|_1) & \text{if } \bar{x} \notin B_k, \end{cases}$$

where $\Pi_1(\bar{x}, B_k)$ is the projection of $\bar{x}$ on $B_k$ with respect to the $\ell_1$ norm. We sample a fixed number of elements per dataset $D_k$ with possible redundancy (due to sampling with replacement) using the biased functions $\omega_k$ as sampling weights.

**Approximating the biasing functions.** We first compute the minimum $m^{(k)} \in [0, 1]^3$ and maximum $M^{(k)} \in [0, 1]^3$ of each dataset $D_k$ for the average border values $\bar{x}$, with $m^{(k)} = (m_H^{(k)}, m_S^{(k)}, m_V^{(k)})$ and $M^{(k)} = (M_H^{(k)}, M_S^{(k)}, M_V^{(k)})$.

Then, we approximate the biasing functions by $\tilde{\omega}(x) = \mathbb{I} \{ \bar{x}_C \in [m_C^{(k)}, M_C^{(k)}] \} \text{ for all } C \in \{H, S, V\}$, (7) to solve for the sampling weights.

**Balanced datasets scenario.** We first considered a balanced scenario, where each dataset has the same number of instances, i.e., $n_k = 50,000/K$ for all $k \in \{1, \ldots, K\}$. In this scenario, the debiased dataset is still biased in some sense, since (7) does not take into account the natural distribution of the CIFAR training dataset onto the bins $B_0, \ldots, B_7$ (see top figure of Figure 8). Although the results should improve with the debiasing procedure, they cannot be compared directly to the classification accuracy with the original training dataset of CIFAR. Following the outline of Section IV-B, we provide in Figure 8 the marginal distribution of the bias-corrected samples over the HSV dimensions of the average 2-pixel borders and compare them to the test marginal distributions. We observe that the corrected distributions are closer to the true distributions and less random when $\gamma$ is high, although some expected bias remains. We provide the corresponding accuracies as well as an estimator of their variance for the learned models in Table II. Our results show that the accuracy of the debiasing procedure increases with the overlap.
Imbalanced datasets scenario. Dividing the data over HSV bins can split CIFAR into rather balanced bins, see Figure 9. However, in many practical scenarios, the different sources of data do not provide the same number of observations. In that case, concatenating the databases can lead to very bad results. We propose to study the relative performance of our debiasing methodology against the concatenation of the databases, for a long-tail distribution of the number of elements per dataset, as illustrated in Figure 9. Precisely, we fix $n_k \propto \alpha^k$ where $\alpha = 0.75$ and $n = \sum_{k=1}^{K} n_k = 50,000$. In this experiment, we take into account the size of the datasets relatively to the size of the bins in the testing data using an additional multiplicative term in $\bar{\omega}_k$. Results are provided in table III, highlighting the great impact of overlapping.
V. CONCLUSION

In this work, we applied to visual recognition problems the reweighting procedure of [2] for learning with several biased datasets with known biasing functions $\omega_k$’s. Their approach is not readily applicable to image data, since images datasets typically do not occupy the same portions of the space of all possible images. To circumvent this problem, we propose to express the biasing functions either 1) by using additional information on the images, or 2) on smaller spaces by transforming the information contained in the images. While our work demonstrates the effectiveness of a general methodology to learn with biased data, the choice of the biasing functions is deferred to the practitioner, and we cannot provide precise information on the expected effects of reweighting the data. For these reasons, we consider the two most promising directions for future work to consist in: 1) finding a general methodology for choosing relevant approximations to biasing functions, and 2) predicting the impact of the reweighting procedure before training, which is especially important when dealing with large models.

APPENDIX

In this appendix, we provide the technical details about the computation of the $\hat{W}_k$. Recall that the $\hat{W}_k$ are empirical estimators of the $\Omega_k$, that are meant to be plugged into Equation (4) to produce an unbiased estimate of $p_{\text{test}}$. For all $k \in \{1, \ldots, K\}$, we have

$$\Omega_k = \int \omega_k(z) dp_{\text{test}}(z) \quad (8)$$

and

$$p_{\text{test}} = \left( \sum_{l=1}^{K} \frac{\lambda_l \omega_l}{\Omega_l} \right)^{-1} \hat{p}. \quad (9)$$

Plugging (9) into (8), we obtain that for all $k \in \{1, \ldots, K\}$

$$\Omega_k = \int \frac{\omega_k(z)}{\sum_{l=1}^{K} \frac{\lambda_l \omega_l}{\Omega_l}} d\hat{p}. \quad (10)$$

In other words, the vector $\mathbf{\Omega} = (\Omega_1, \ldots, \Omega_K) \in \mathbb{R}^K$ is a solution to the system of $K$ equations

$$\forall k \in \{1, \ldots, K\}, \quad \Gamma_k(W) = 1,$$

where $\Gamma_k : \mathbb{R}^K \to \mathbb{R}$ is defined as

$$\Gamma_k(W) = \frac{1}{\hat{W}_k} \int \frac{\omega_k(z)}{\sum_{l=1}^{K} \frac{\lambda_l \omega_l}{\Omega_l}} d\hat{p}(z). \quad (11)$$

Therefore, a natural estimator $\hat{W} = (\hat{W}_1, \ldots, \hat{W}_K)$ of $\mathbf{\Omega}$ is given by the solution to the system of equations

$$\forall k \in \{1, \ldots, K\}, \quad \hat{\Gamma}_k(W) = 1,$$

where $\hat{\Gamma}_k$ is the empirical counterpart of $\Gamma_k$, i.e., defined as (10) but with $\hat{p}$ instead of $\hat{p}$.

Note that $\Gamma_k$ (and similarly $\hat{\Gamma}_k$) is homogeneous of degree 0, such that an infinite number of solutions might exist. To ensure uniqueness, one can enforce $\hat{W}_K = 1$ and solve the system composed of the first $K-1$ equations only. This convention has no impact on $\hat{p}$, as it is equivalent to plug $W$ or $tW$ into Equation (4).

The remaining question is: how to solve system (11)? Let $u_k = \log(\lambda_k/W_k)$. We can rewrite system (11) as:

$$\forall k \in \{1, \ldots, K\}$$

$$\int \frac{e^{u_k} \omega_k(z)}{\sum_{l=1}^{K} e^{u_l} \omega_l(z)} d\hat{p}(z) - \lambda_k = 0. \quad (12)$$

Now, the left-hand side of (12) can be interpreted as the $k$-th component of the gradient of the function $D : \mathbb{R}^K \to \mathbb{R}$ defined as

$$D(u) = \log \left( \sum_{l=1}^{K} e^{u_l} \omega_l(z) \right) d\hat{p}(z) - \sum_{l=1}^{K} \lambda_l u_l. \quad (13)$$

Thus, solving system (11) is equivalent to solving system (12), and amounts to maximizing over $\mathbb{R}^K$ the function $D$, which has been shown in [1] to be strongly convex if the distributions overlap sufficiently. A standard Gradient Descent strategy (whose gradient is given by Equation (12)) can thus be used, and converges towards the unique minimum. Recall also that

$$\hat{p}(z) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n_k} 1 \{ z = z_i^{(k)} \},$$

so that $\hat{D}$ actually rewrites

$$\hat{D}(u) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left( \log \left( \sum_{l=1}^{K} e^{u_l} \omega_l(z_i^{(l)}) \right) - \sum_{l=1}^{K} \lambda_l u_l \right).$$

The function $\hat{D}$ is thus separable in the observations, and a stochastic or minibatch variant of Gradient Descent can be used to reduce the computational cost.

We conclude with a remark about the datasets relative sizes. In this paper, we have assumed for simplicity that the datasets proportions $n_k/n$ are constant equal to ideal proportions $\lambda_k$. A natural way to relax this assumption is to consider random empirical proportions $\hat{\lambda}_k = n_k/n$ that converge to the ideal proportions $\lambda_k$ as $n$ goes to infinity. We highlight that our methodology remains valid in the above setting, as $\hat{p} = \sum_{k=1}^{K} \hat{\lambda}_k \hat{p}_k$ still converges towards $\hat{p} = \sum_{k=1}^{K} \lambda_k p_k$. 
ACKNOWLEDGMENTS

This research work was partially supported by the research chair “Good In Tech : Rethinking innovation and technology as drivers of a better world for and by humans”, under the auspices of the “Fondation du Risque” and in partnership with the Institut Mines-Télécom, Sciences Po, Afnor, Ag2r La Mondiale, CGI France, Danone and Sycomore.

REFERENCES

[1] R. D. Gill, Y. Vardi, and J. A. Wellner, “Large sample theory of empirical distributions in biased sampling models,” Annals of Statistics, vol. 16, no. 3, pp. 1069–1112, 1988.

[2] P. Laforgue and S. Cléménçon, “Statistical learning from biased training samples,” CoRR, vol. abs/1906.12304, 2019.

[3] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” University of Massachusetts, Amherst, Tech. Rep. 07-49, 2007.

[4] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “Ms-celeb-1m: A dataset and benchmark for large-scale face recognition,” in Computer Vision - ECCV 2016 - 14th European Conference, Proceedings, Part III, ser. Lecture Notes in Computer Science, vol. 9907. Springer, 2016, pp. 87–92.

[5] P. J. Phillips, F. Jiang, A. Narvekar, J. H. Ayyad, and A. J. O’Toole, “An other-race effect for face recognition algorithms,” ACM Transactions on Applied Perception, vol. 8, no. 2, pp. 14:1–14:11, 2011.

[6] P. Grother and M. Ngan, “Face Recognition Vendor Test (FRVT) — Performance of Automated Gender Classification Algorithms,” National Institute of Standards and Technology (NIST), Tech. Rep. NISTIR 8052, 2001.

[7] M. Wang, W. Deng, J. Hu, X. Tao, and Y. Huang, “Racial faces in the wild: Reducing racial bias by information maximization adaptation network,” in 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019. IEEE, 2019, pp. 692–702.

[8] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” in Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, 2016, pp. 3630–3638.

[9] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.

[10] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” Machine Learning, vol. 79, no. 1-2, pp. 151–175, 2010.

[11] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, “Simultaneous deep adaptation with residual transfer networks,” in Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, 2016, pp. 136–144.

[12] K. Bourmobiles, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, “Domain separation networks,” in Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, 2016, pp. 343–351.

[13] D. Li, Y. Yang, Y. Song, and T. M. Hospedales, “Learning to generalize: Meta-learning for domain generalization,” in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18). AAAI Press, 2018, pp. 3490–3497.

[14] Y. Vardi, “Empirical distributions in selection bias models,” Ann. Statist., vol. 13, no. 1, pp. 178–203, 03 1985.