REPAIR: Removing Representation Bias by Dataset Resampling

Yi Li
UC San Diego
yil898@ucsd.edu

Nuno Vasconcelos
UC San Diego
nvasconcelos@ucsd.edu

Abstract

Modern machine learning datasets can have biases for certain representations that are leveraged by algorithms to achieve high performance without learning to solve the underlying task. This problem is referred to as "representation bias". The question of how to reduce the representation biases of a dataset is investigated and a new dataset REPresentAtion bIas Removal (REPAIR) procedure is proposed. This formulates bias minimization as an optimization problem, seeking a weight distribution that penalizes examples easy for a classifier built on a given feature representation. Bias reduction is then equated to maximizing the ratio between the classification loss on the reweighted dataset and the uncertainty of the ground-truth class labels. This is a minimax problem that REPAIR solves by alternatingly updating classifier parameters and dataset resampling weights, using stochastic gradient descent. An experimental set-up is also introduced to measure the bias of any dataset for a given representation, and the impact of this bias on the performance of recognition models. Experiments with synthetic and action recognition data show that dataset REPAIR can significantly reduce representation bias, and lead to improved generalization of models trained on REPAIRed datasets. The tools used for characterizing representation bias, and the proposed dataset REPAIR algorithm, are available at https://github.com/JerryYLi/Dataset-REPAIR/.

1. Introduction

Over the last decade, deep neural networks (DNNs) have enabled transformational advances in various fields, delivering superior performance on large-scale benchmarks. However like any other machine learning systems, the quality of DNNs is only as good as that of the datasets on which they are trained. In this regard, there are at least two sources of concern. First, they can have limited generalization beyond their training domain [32, 2]. This is classically known as *dataset bias*. Second, the learning procedure could give rise to biased deep learning algorithms [3, 25]. *Representation bias* is an instance of this problem, that follows from training on datasets that favor certain representations over others [22]. When a dataset is easily solved by adoption of a specific feature representation ϕ, it is said to be biased towards ϕ. Bias is by itself not negative: If the classification of *scenes*, within a certain application context, is highly dependent on the detection of certain *objects*, successful scene recognition systems are likely to require detailed object representations. In this application context, scene recognition datasets should exhibit *object bias*. However, in the absence of mechanisms to measure and control bias, it is unclear if conclusions derived from experiments are tainted by undesirable biases. When this is the case, learning algorithms could simply overfit to the dataset biases, hampering generalization beyond the specific dataset.

This problem is particularly relevant for action recognition, where a wide range of diverse visual cues can be informative of action class labels, and leveraged by different algorithms. In the literature, different algorithms tend to implement different representations. Some models infer action categories from one or a few video frames [27, 14, 40], while others attempt to model long-term dependencies [35, 37, 9]; some focus on modeling human pose [15], and some prefer to incorporate contextual information [10]. In general, two algorithms that perform equally well on a dataset biased towards a representation, *e.g.* a dataset with *static* or single frame bias, can behave in a drastically different manner when the dataset is augmented with examples that eliminate this bias, *e.g.* by requiring more temporal reasoning. Without the ability to control the static bias of the dataset, it is impossible to rule out the possibility that good performance is due to the ability of algorithms to pick up spurious static visual cues (*e.g.* backgrounds, objects, *etc.*) instead of modeling action.

In this work, we investigate the question of how to reduce the representation biases of a dataset. For this, we introduce a new *REPresentAtion bIas Removal* (REPAIR) procedure for dataset resampling, based on an a formulation of bias minimization as an optimization problem. REPAIR seeks a set of example-level weights penalizing examples that are easy for a classifier built on a given feature representation. This is implemented by using a DNN as feature extractor for

the representation of interest and learning an independent linear classifier to classify these features. Bias reduction is then equated to maximizing the ratio between the loss of this classifier on the reweighted dataset and the uncertainty of the ground truth class labels. We show that this reduces to a minimax problem, solved by alternatingly updating the classifier coefficients and the dataset resampling weights, using stochastic gradient descent (SGD).

Beyond introducing the dataset REPAIR procedure, we develop an experimental procedure for its evaluation. We consider two scenarios in this work. The first is a controlled experiment where we explicitly add *color bias* to an otherwise unbiased dataset of grayscale images. This enables the design of experiments that explicitly measure recognition performance as a function of the amount of bias. The second is action recognition from videos, where many popular datasets are known to have *static bias*. In both cases, dataset REPAIR is shown to substantially reduces representation bias, which is not possible with random subsampling. A generic set-up is then introduced to evaluate the effect of representation bias on model training and evaluation. This has two main components. The first measures how the performance of different algorithms varies as a function of the bias of datasets towards a given representation. The second analyzes how representation bias affects the ability of algorithms to generalize across datasets. Various experiments in this set-up are performed leading to a series of interesting findings about behavior of models on resampled datasets.

Overall, the paper makes three main contributions. The first is a novel formulation of representation bias minimization as a differentiable and directly optimizable problem. The second is a SGD-based dataset resampling strategy, REPAIR, which is shown able to significantly reduce representation bias. The third is a new experimental set-up for evaluating dataset resampling algorithms, that helps determine the importance of such resampling to achieving both model generalization and fair algorithm comparisons.

2. Related Work

Fair Machine Leaning. As data-driven learning systems are used in an increasingly larger array of real-world applications, the fairness and bias of the decisions made by these systems becomes an important topic of study. In recent years, different criteria have been proposed to assess the fairness of learning algorithms [38, 7, 12], stimulating attempts to build unbiased algorithms. In general, deep learning systems are apt at capturing or even magnifying biases in their supervisory information [25, 39, 1, 29]. This is in part due to the end-to-end nature of their training, which encourages models to exploit biased features if this leads to accurate classification. Prior works have mostly focused on uncovering and addressing different instances of bias in learned models, including gender bias [3, 39, 1] and racial bias [29]. However, the bias of the data itself has received less attention from the community.

Dataset Bias. While datasets are expected to resemble the probability distribution of observations, the data collection procedure can be biased by human and systematic factors, leading to distribution mismatch between dataset and reality, as well as between two datasets. This is referred to as dataset bias [32, 30]. [32] analyzed the forms of bias present in different image recognition datasets, and demonstrated its negative effect on cross-dataset model generalization. Dataset bias has been well studied and can be compensated with domain adaptation techniques [18, 8, 24].

Representation bias is a more recent concept, describing the ability of a representation to solve a dataset. It was first explicitly formulated in [22], and used to measure the bias of modern action recognition datasets towards objects, scenes and people. Representation bias is different from dataset bias, in that it enables potential "shortcuts" (the representations for which the dataset is biased) that a model can exploit to solve the dataset, without learning the underlying task of interest. For example, contextual bias allows recognition algorithms to recognize objects by simply observing their environment [31]. Even when an agent does not rely solely on shortcuts, its decisions may be biased for these representations, as [25] showed in their case study of how shape bias is captured by models trained on ImageNet.

Video Action Recognition. Early efforts at human action recognition mainly relied on compact video descriptors encoding hand-crafted spatiotemporal features [20, 34, 35]. Deep learning approaches, like two-stream networks [27], 3D convolutional networks [16, 33] and recurrent neural networks [37], use network architectures that learn all relevant features. A common theme across many action recognition works is to capture long-term temporal structure in the video. However, current datasets have an abundance of static cues that can give away the action (*i.e.* bias towards static representations), making it difficult to assess the importance of long-term temporal modeling. The presence of this static bias has been noted and studied in previous work: [10] exploited contextual cues to achieve state-of-the-art action recognition performance. [6] visualized action models to uncover unwanted biases in training data. Finally, [14] identified action categories that can be recognized without any temporal reasoning.

Dataset Resampling. Resampling refers to the practice of obtaining sample points with different frequencies than those of the original distribution. It is commonly used in machine learning to balance datasets, by oversampling minority classes and under-sampling majority ones [5]. By altering relative frequencies of examples, dataset resampling enables the training of fairer models, which do not discriminate against minority classes.

3. Minimum-bias Dataset Resampling

3.1. Representation Bias

Representation bias [22] captures the bias of a dataset with respect to a representation. Let $\phi : \mathcal{X} \rightarrow \mathcal{Z}$ be a feature representation. The bias of dataset $\mathcal{D}$ towards ϕ is the *best achievable performance* of the features $\phi(x)$ on $\mathcal{D}$ normalized by chance level. In this work, we measure classification performance with the risk defined by the cross-entropy loss

$$\mathcal{R}^*(\mathcal{D}, \phi) = \min_{\theta} \mathbb{E}_{X,Y}[-\log P(Y \mid Z; \theta)] \tag{1}$$

where X and Y are examples and their respective labels, and $Z = \phi(X)$ is the feature-space representation of X. Here $P(Y \mid \phi(X); \theta)$ is computed by a softmax layer (weight matrix plus softmax nonlinearity) of input Z and parameters θ, which are optimized by gradient descent. We do not fine-tune the representation ϕ itself to retain its original semantics; only the parameters of the softmax layer are learned. Noting that minimizing the cross-entropy loss encourages the softmax classifier to output the true posterior class probabilities $P(Y \mid Z)$, we may rewrite (1) as

$$\begin{aligned}\mathcal{R}^*(\mathcal{D}, \phi) &= \mathbb{E}_{Z,Y}[-\log P(Y \mid Z)] \\ &= \mathbb{E}_{Z,Y}\left[-\log P(Y) - \log \frac{P(Z,Y)}{P(Z)P(Y)}\right] \\ &= H(Y) - I(Z,Y)\end{aligned} \tag{2}$$

The risk $\mathcal{R}^*(\mathcal{D}, \phi)$ is therefore upper-bounded by the entropy of class label Y and decreases as the mutual information between the feature vector Z and the label Y increases. Hence, a lower $\mathcal{R}^*(\mathcal{D}, \phi)$ indicates that ϕ is more informative for solving $\mathcal{D}$, *i.e.* the representation bias is larger. This is captured by defining bias as

$$\mathcal{B}(\mathcal{D}, \phi) = \frac{I(Z,Y)}{H(Y)} = 1 - \frac{\mathcal{R}^*(\mathcal{D}, \phi)}{H(Y)}. \tag{3}$$

Intuitively, bias has a value in $[0, 1]$ that characterizes the *reduction in uncertainty* about the class label Y when feature Z is observed. The normalization term $H(Y)$ guarantees fairness of bias measurements when datasets have different numbers of classes. In practice the terms used to define bias (3) are estimated by their empirical values

$$\mathcal{R}^*(\mathcal{D}, \phi) \approx \min_{\theta} -\frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},y)\in\mathcal{D}} \log P(y \mid \mathbf{x}; \theta) \tag{4}$$

$$H(Y) \approx -\frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},y)\in\mathcal{D}} \log p_y \tag{5}$$

where p_y is the frequency of class y. Measuring the bias thus amounts to learning a linear classifier θ, referred to as *bias estimator*, and recording its cross-entropy loss as well as the class frequencies. It should be noted that the bias formulation of (3) differs from that of [22], in that 1) the bias value is properly normalized to the range $[0, 1]$, and 2) bias is differentiable w.r.t. θ. The last property is particularly important, as it enables bias optimization.

3.2. Adversarial Example Reweighting

Representation bias can be problematic because it implies that the dataset $\mathcal{D}$ favors some representations over others. While there is an unknown ground-truth representation ϕ^* that achieves the best performance on a task, this may not be the case for a dataset $\mathcal{D}$ of that task, if the dataset is biased towards other representations. We provide some simple examples of this in Sections 4.1 and 4.2. When this is the case, it is desirable to modify the dataset so as to minimize bias. One possibility, that we explore in this work, is to perform dataset resampling. While the risk of (4) and entropy of (5) assign equal weight to each example in $\mathcal{D}$, bias can be controlled by prioritizing certain examples over others. In other words, we attempt to create a new dataset $\mathcal{D}'$ of reduced bias, by non-uniformly sampling examples from the existing dataset $\mathcal{D}$. For this, it suffices to augment each example $(\mathbf{x}_i, y_i) \in \mathcal{D}$ with a weight w_i that encodes the probability of the example being selected by the resampling procedure. This transforms (4) and (5) into

$$\mathcal{R}^*(\mathcal{D}', \phi) \approx \min_{\theta} -\sum_{i=1}^{|\mathcal{D}|} \frac{w_i}{\sum_i w_i} \log P(y_i \mid \mathbf{x}_i; \theta) \tag{6}$$

$$H(Y') \approx -\sum_{i=1}^{|\mathcal{D}|} \frac{w_i}{\sum_i w_i} \log p'_{y_i}, \tag{7}$$

where

$$p'_y = \frac{\sum_{i:y_i=y} w_i}{\sum_i w_i}. \tag{8}$$

The goal is then to find the set of weights $\{w_i\}_{i=1}^{|\mathcal{D}|}$ that minimizes the bias

$$\mathcal{B}(\mathcal{D}', \phi) = 1 - \frac{\mathcal{R}^*(\mathcal{D}', \phi)}{H(Y')}. \tag{9}$$

This leads to the optimization problem

$$(w^*, \theta^*) = \min_{w} \max_{\theta} \mathcal{V}(w, \theta) \tag{10}$$

$$\mathcal{V}(w, \theta) = 1 - \frac{\sum_i w_i \log P(y_i \mid \mathbf{x}_i; \theta)}{\sum_i w_i \log p'_{y_i}} \tag{11}$$

To solve the minimax game of (10), we optimize the example weights $\mathbf{w} = (w_1, \ldots, w_{|\mathcal{D}|})$ and the bias estimator θ in an alternating fashion, similar to the procedure used to train adversarial networks [11]. To guarantee that the

weights w_i are binary probabilities, we define $\mathbf{w}$ as the output of a sigmoid function $w_i = \rho(\omega_i) = (1 + e^{-\omega_i})^{-1} \in (0, 1)$, and update ω_i directly. Throughout the training iterations, the optimization of θ with a classification loss produces more accurate estimates of the representation bias. On the other hand, the optimization of $\mathbf{w}$ attempts to minimize this bias estimate by assigning larger weights to misclassified examples. Upon convergence, θ^* is a precise measure of the bias of the reweighted dataset, and $\mathbf{w}^*$ ensures that this bias is indeed minimized.

Resampling $\mathcal{D}$ according to the distribution w_i leads to a dataset $\mathcal{D}'$ that is less biased for representation ϕ, while penalizing classes that do not contribute to the classification uncertainty. Because this has the effect of equalizing the preference of the dataset for different representations, we denote this process as dataset *REPresentAtion bIas Removal* (REPAIR).

3.3. Mini-batch Optimization

Efficient optimization on a large-scale dataset usually requires mini-batch approximations. The objective function above can be easily adapted to mini-batch algorithms. For this, it suffices to define

$$r_i = \frac{w_i}{\bar{w}} = |\mathcal{D}| \frac{w_i}{\sum_i w_i} \tag{12}$$

where $\bar{w}$ is the sample average of w_i. The risk of (6) and the entropy of (7) can then be rewritten as

$$\mathcal{R}^*(\mathcal{D}', \phi) \approx \min_\theta -\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} r_i \log P(y_i \mid \mathbf{x}_i; \theta) \tag{13}$$

$$H(Y') \approx -\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} r_i \log p'_{y_i}, \tag{14}$$

and estimated from mini-batches, by replacing $|\mathcal{D}|$ with the mini-batch size. This enables the use of mini-batch SGD for solving the optimal weights of (10). In practice REPAIR is performed on training and test splits of $\mathcal{D}$ combined, to ensure that the training and test sets distributions are matched after resampling.

4. Case studies

In this section, we introduce two case studies for the study of bias reduction. The first is based on an artificial setting where bias can be controlled explicitly. The second uses the natural setting of action recognition from large-scale video datasets. While the ground-truth representation is not known for this setting, it is suspected that several biases are prevalent in existing datasets. In both cases, we investigate how representation biases can impair the fairness of model evaluation, and prevent the learning of representations that generalize well.

4.1. Colored MNIST

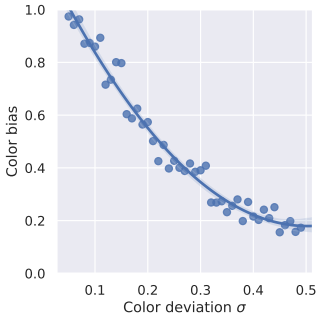
The first case study is based on a modified version of MNIST [21], which is denoted *Colored MNIST*. It exploits the intuition that digit recognition does not require color processing. Hence, the ground-truth representation for the *task* of digit recognition should not involve color processing. This is indeed guaranteed for representations learned on a grayscale dataset like MNIST. However, by introducing color, it is possible to create a dataset biased for color representations.

Experiment Setup. To introduce *color bias*, we color each digit, using a different color for digits of different classes, as shown in Figure 1a. Coloring was performed by assigning to each example $\mathbf{x}_i$ a color vector $\mathbf{z}_i = (r_i, g_i, b_i)$ in the RGB color space. Color vectors were sampled from class-dependent color distributions, *i.e.* examples of digit y were colored with vectors sampled from a normal distribution distribution $p_y(\mathbf{z})$ of mean $\mu_y = (\mu_y^r, \mu_y^g, \mu_y^b)$ and covariance covariance $\Sigma_y = \sigma^2 I$. Since the simple observation of the color gives away the digit, Colored MNIST is biased for color representations $\mathbf{z}$. When learned on this dataset, a CNN can achieve high recognition accuracies without modeling any property of digits other than color. The color assignment scheme also enables control over the strength of this bias. By altering the means and variances of the different classes, it is possible to create more or less overlap between the color distributions, making color more or less informative of the class label.

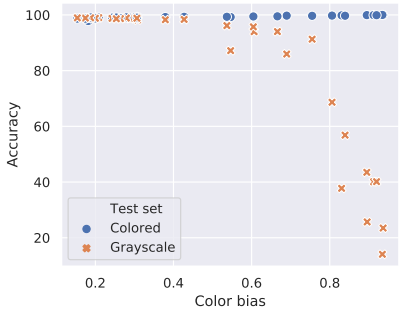
Bias and Generalization. To understand how representation bias affects the fair evaluation of models, we trained a LeNet-5 CNN on the Colored MNIST training set and compared its ability to recognize digits on the test sets of both the Colored MNIST and the original (grayscale) MNIST datasets. To control the color bias of Colored MNIST, we varied the variance σ of the color distributions. Figure 1b shows how the bias, computed with (3) on the colored test set, varies with σ. Clearly, increasing the variance σ reduces bias. This was expected, since large variances create more overlap between the colors of the different classes, making color less discriminant.

Figure 1c shows the recognition accuracy of the learned CNN on the two test sets, as a function of the color bias. A few observations can be drawn from the figure. First, it is clear that CNN performance on MNIST degrades as the bias increases. This shows that representation bias can hurt the generalization performance of the CNN. Second, this effect can be overwhelming. For the highest levels of bias, the performance on MNIST drops close to chance level (10% on this dataset). This shows that, when Colored MNIST is strongly biased for color, the CNN learns a representation that mostly accounts for color. While sensible to solve the training dataset (Colored MNIST), this is a terrible strategy

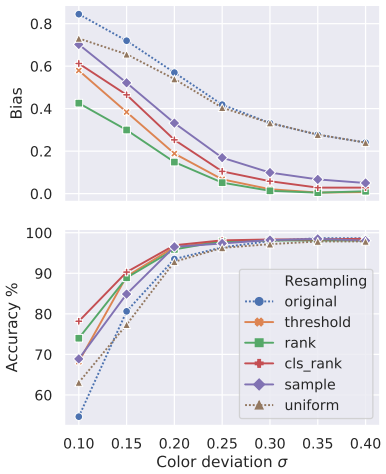
(a) Random digit examples, before (**top**) and after (**bottom**) resampling.



(b) Controlling bias via intra-class color variation.



(c) Test accuracy on biased (**colored**) and unbiased (**grayscale**) test sets.



(d) **Top:** Bias of resampled datasets. **Bottom:** Generalization performance.

Figure 1: Dataset Resampling on Colored MNIST Dataset.

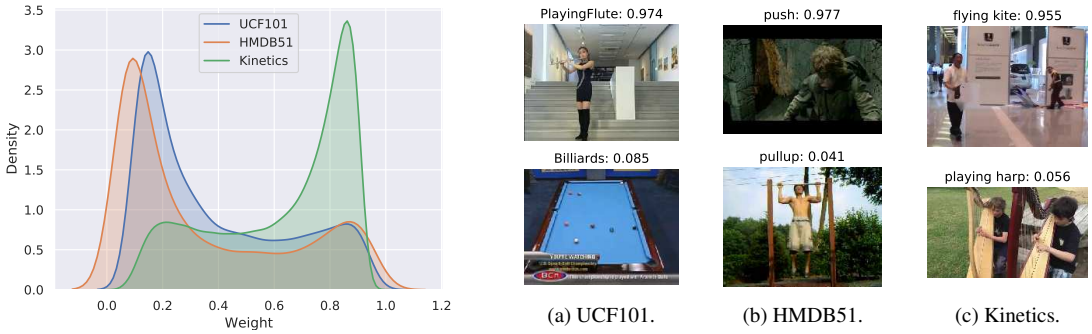
to solve the digit recognition *task* in general. As demonstrated by the poor performance on MNIST, the CNN has not learned anything about digits or digit recognition, simply overfitting to the bias of the training set. Finally, and perhaps most important, this poor generalization is not visible on the Colored MNIST test set, on which the CNN reports deceptively high classification accuracy. The problem is that, like the training set, this is biased for color. Note that adding more Colored MNIST style data will not solve the problem. The overfitting follows from the bias induced by the procedure used to *collect* the data, not from a *shortage* of data. Unless the dataset collection procedure is changed, adding more data only makes the CNN more likely to overfit to the bias.

While this example is contrived, similar problems frequently occur in practice. A set of classes is defined and a data collection procedure, *e.g.* data collection on the web, is chosen. These choices can introduce representation biases, which will be present independently of how large the dataset is. There are many possible sources of such biases, including the fact that some classes may appear against certain types of backgrounds, contain certain objects, occur in certain types of scenes or contexts, exhibit some types of motion, *etc*. Any of these can play the role of the digit colors of Colored MNIST. Since, in general, the test set is collected using a protocol similar to that used to collect the training set, it is impossible to detect representation bias from test set results or to reduce bias by collecting more data. Hence, there is a need for bias reduction techniques.

Resampling Strategies. We next tested the ability of REPAIR to reduce representation bias on Colored MNIST. REPAIR was implemented according to (10) on the colored training and test sets combined, with learning rates $\gamma_\theta = 10^{-3}$ and $\gamma_w = 10$ for 200 epochs, yielding an optimal weight vector $\mathbf{w}^*$. This was then used to implement a few sampling strategies.

1. **Thresholding** (`threshold`): Keep all examples i such that $w_i \geq t$, where $t = 0.5$ is the threshold;
2. **Ranking** (`rank`): Keep $p = 50\%$ examples of largest weights w_i;
3. **Per-class ranking** (`cls_rank`): Keep the $p = 50\%$ examples of largest weight w_i from each class;
4. **Sampling** (`sample`): Keep each example i with probability w_i (discard with probability $1 - w_i$).
5. **Uniform** (`uniform`): Keep $p = 50\%$ examples uniformly at random.

To evaluate the resampling strategies, we tested their ability to reduce representation bias and improve model generalization (test accuracy on MNIST). The experiments were performed with different color variances σ, to simulate different level of bias. The results were averaged over 5 runs under each setting. Figure 1d (**top**) shows the bias after resampling, as a function of σ. All four strategies where resampling leverages the weights w_i led to a significant reduction in color bias, relative to both the bias before resampling and that achieved by uniform resampling. Among them, thresholding and ranking were more effective for large biases (small values of σ). The reduction in color bias also led to better model generalization, as shown in Figure 1d (**bottom**). This confirms the expectation that large bias harms the generalization ability of the learned model. Visual inspection of examples from the REPAIRed dataset, shown in Figure 1a (**bottom**), explains this behavior. Since it becomes harder to infer the digits from their color, the CNN must rely more strongly on shape modelling, and thus generalizes better.



(a) UCF101. (b) HMDB51. (c) Kinetics.

Figure 2: **Left**: Histograms of resampling weights. **Right**: Examples with highest and lowest weights from each dataset.

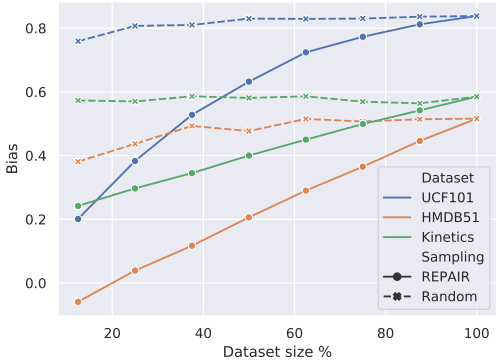


Figure 3: Static bias as a function of dataset size. Examples are removed either randomly or according to their weights.

4.2. Scenario II: Action Recognition

Video action recognition is a complex task with various potential sources of bias, as shown by the analysis of [22]. In this work, we focus on static bias, *i.e*. bias towards single-frame representations. The observation that popular action recognition datasets like UCF101 [28] and Kinetics [17] are biased for static features, in that substantial portions of their data can be solved without leveraging temporal information, has been reported by several recent works [14, 6]. Yet, little attention has been given to the impact of bias on learning and evaluation of action recognition models.

In this section, we present an in-depth analysis on the connection between static dataset bias and model performance on the dataset. We used REPAIR to manipulate the static bias of a dataset, through the selection of examples according to their learned weights. We then evaluated how the performance of prevailing action recognition models changes as a function of static bias. This allowed us to compare the sensitivity of the models to the presence of static cues in the data. Finally, by examining models trained on datasets with different level of static bias, we assessed their ability to capture temporal information and learn human actions that generalize across datasets.

Static Bias Minimization. We implemented ϕ with ImageNet features extracted from the ResNet-50 [13], a typical representation for static image recognition. REPAIR weights were learned for 20k iterations with learning rate $\gamma_\theta = 10^{-3}$ and $\gamma_w = 10^{-3}|\mathcal{D}|$, as the number of weights w_i to be learned grows linearly with dataset size. Figure 2 (**left**) shows the distribution of resampled weights learned for UCF101 [28], HMDB51 [19] and Kinetics [17]; A random frame from videos of highest and lowest weights is displayed in Figure 2 (**right**). Several observations can be made. First, REPAIR uncovers videos with abundant static cues (*e.g*. pool tables in *billiards* and parallel vertical lines in *playing harp*). These videos receive lower scores during resampling. On the other hand, videos with no significant static cues (*e.g*. complex human interactions in *push*), are more likely to be selected into the resampled dataset. Second, the optimization does not learn the trivial solution of setting all weights to zero. Instead, the weights of all videos range widely from 0 to 1, forming two clusters at both ends of the histogram. Third, while all datasets contain a substantial amount of videos that contribute to static bias, Kinetics contained more videos of large weight ($w > 0.5$), enabling more freedom in the assembly of new datasets.

Following the *ranking* strategy of section 4.1, the videos were sorted by decreasing weights. Resampled datasets were then formed by keeping the top $p\%$ of the data and eliminating the rest (value of p varies). Figure 3 shows how the static biases of the three datasets are reduced by this resampling procedure. This is compared to random sampling the same number of examples. The bias of (3) was computed as the maximum over 5 measurements, each time training the bias estimator θ with a different weight decay, ranging from 10^{-1} to 10^{-5}, so as to prevent overfitting due to insufficient training data. The bias curves validate the effectiveness of REPAIR, as the static classifier performs much weaker on the REPAIRed datasets (hence less static bias). This is unlike random sampling, which does not affect the bias measurements significantly. These results are also interesting because they enable us to alter static dataset bias within a considerable range of values, for further experiments with action recognition models.

Video Models *vs*. Static Bias. To evaluate how representation bias affects the action recognition performance of dif-

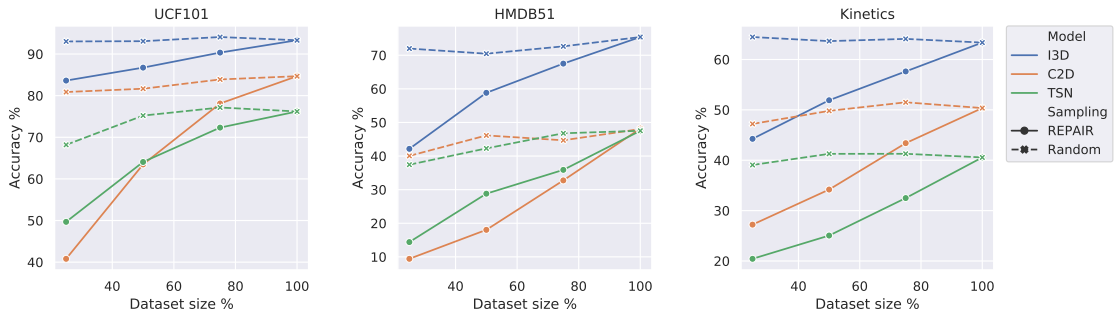


Figure 4: Evaluations of action recognition models on resampled datasets.

ferent models, we trained and evaluated three models from the literature on the original and REPAIRed action datasets:

1. 2D ConvNet (**C2D**): Baseline ResNet-50 applied independently to each frame, predictions then averaged. Pre-trained on ImageNet [26].
2. Temporal segment network (**TSN**) [36]: Aggregating features (we used *RGB-diff*) from multiple snippets of the video according to their segmental consensus. Pre-trained on ImageNet.
3. Inflated 3D ConvNet (**I3D**) [4]: Spatiotemporal convolutions inflated from a 2D Inception-v1 network. Pre-trained on ImageNet and Kinetics.

The networks were fine-tuned through SGD with learning rate 10^{-3} and momentum 0.9, for 10k iterations on UCF101 and HMDB51 and 50k iterations on Kinetics. Figure 4 shows the performance of all three models on the three datasets. It is clear that all networks have weaker performance on the REPAIRed datasets (smaller static bias) than on the origonal ones. The drop in accuracy is a measure of the reliance of the action models on static features, which we denote as the static bias dependency of the models. More precisely, we define the *static bias dependency coefficient* β of a model on representation ϕ as the difference between model performance on randomly sampled and REPAIRed datasets, averaged over resampling rates (0.25, 0.5 and 0.75 in this case). The larger β is, the more the model leverages static bias to solve the dataset; $\beta = 0$ indicates that model performance is independent of static bias. Table 1 summarizes the dependency coefficients of the different models, showing that C2D has much larger static bias dependency than TSN and I3D. While this comparison is not, by itself, enough to conclude that one model is superior to the rest, the reduced static bias dependency of the more recent networks suggests that efforts towards building better spatiotemporal models are paying off.

Another notable observation from Figure 4 is that the ranking of models by their performance on the original dataset is not necessarily meaningful. For example, while C2D outperforms TSN on UCF101, the reverse holds after 50% and 25% resampling. This shows that rankings

	C2D [27]	TSN [36]	I3D [4]
UCF101	0.213	0.115	**0.065**
HMDB51	0.236	**0.148**	0.155
Kinetics	0.146	0.146	**0.128**
Average	0.198	0.136	**0.116**

Table 1: Static bias dependency coefficient β of the three action recognition models, evaluated on the three different datasets.

Dataset size	100% (*orig.*)	75%	50%	25%
mAP %	61.48	**63.45**	63.06	63.24

Table 2: Cross-dataset generalization from **Kinetics** to **HMDB51** over 12 common classes. See Figure 5 for per-class AP.

of action recognition architectures could simply reflect how much they leverage representations biases. For example, stronger temporal models could underperform weaker static models if the dataset has a large static bias, potentially leading to unfairness in model evaluation. By reducing representation bias, REPAIR can alleviate this unfairness.

Cross-dataset Generalization. We next compared the performance of the I3D models trained on the original and resampled datasets. Unlike the Colored MNIST experiment of Figure 1c, it is not possible to evaluate generalization on an unbiased test set. Instead, we measured cross-dataset generalization, with similar setup to [32]. This assumes that the datasets do not have the exact same type of representation bias, in which case overfitting to the biases of the training set would hamper generalization ability.

We used Kinetics as the training set and HMDB51 as the test set for generalization performance. The two datasets had 12 action classes in common. While more classes are shared among UCF101 and Kinetics, they are both collected on YouTube and have very similar distributions. HMDB51, on the contrary, consists of videos sourced from movies and other public databases and poses a stronger generalization challenge. The I3D models were trained on the 12 classes of the original and REPAIRed versions of Kinetics, and evaluated *without fine-tuning* on the same classes of HMDB51. Model generalization was evaluated by average precision (AP), measured for each of the common classes.

Figure 5 summarizes the generalization performance of

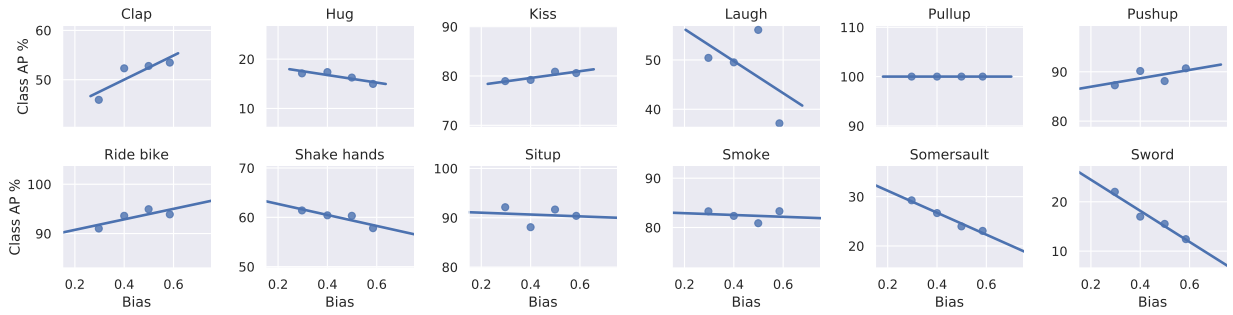


Figure 5: Class-level cross-dataset generalization of I3D models trained on REPAIRed **Kinetics** datasets. Test set is **HMDB51**.

the I3D models as a function of the static bias in the *training* set, for each of the 12 classes. To visualize the correlation among the two variables we also show a line regressed on the different points. The four points in each subplot, from *right* to *left*, correspond to models trained on the original dataset and the REPAIRed ones with 75%, 50% and 25% sampling rate, respectively. Of the 12 classes, 7 showed a negative correlation between bias and generalization. Furthermore, the correlation tends to be strongly negative for the classes where the model generalizes the worst, namely *hug*, *somersault* and *sword*. On the contrary, positive correlation occurs on the classes of high generalization performance. This indicates that, at the class level, there are strong differences between the biases of the two datasets. Classes that generalize well are those where biases are shared across datasets, while low performance ones have different biases. The mean average precision (*mAP*) of all 12 classes increased by $\sim$2% after resampling as shown in Table 2, validating the effectiveness of REPAIR on improving model generalization.

Temporal Reasoning in Learned Models. Finally, we analyzed in greater detail the I3D models learned on the REPAIRed datasets, aiming to understand the improvement in their generalization performance. We hypothesize that, with less static cues to hold on to, the network (even with unchanged structure) should learn to make inferences that are more dependent on the temporal structure of the video. To test this hypothesis, we performed a simple experiment. Given an input video, we measured the Euclidean distance between the feature vectors extracted from its regular 64-frame clip and its time reversed version. This distance was averaged over all video clips in the test set, and is denoted as the *temporal structure score* of the model. Larger scores reflect the fact that the model places more emphasis on the temporal structure of the video, instead of processing frames individually. Note that, because the 3D convolution kernels of I3D are initialized by duplicating the filters of a 2D network [4], the temporal structure score should be zero in the absence of training.

For this experiment, we used the test set of the 20BN-Something-Something-V2 [23] dataset, which is known for the fact that its action classes are often dependent on the arrow of time (*e.g. opening vs. closing*, or *covering vs. uncovering*). Table 3 summarizes the scores obtained for all learned models on the test set of Something-Something. The table shows that, for REPAIRed datasets, the score increases as more biased videos are removed from the dataset. This is not a mere consequence of reduced dataset size, since the score varies little for random discarding of the same number of examples. This is evidence that static bias is an obstacle to the modeling of video dynamics, and dataset REPAIR has the potential to overcome this obstacle.

Training set	Sampling	Training set size			
		100%	75%	50%	25%
UCF101	REPAIR	1.76	**1.76**	**1.92**	**1.96**
	Random		1.75 $\pm$ .03	1.79 $\pm$ .04	1.78 $\pm$ .05
HMDB51	REPAIR	2.04	**2.03**	**2.25**	**2.31**
	Random		2.02 $\pm$ .02	2.07 $\pm$ .07	2.08 $\pm$ .02
Kinetics	REPAIR	3.67	3.63	**3.68**	**3.83**
	Random		**3.66** $\pm$ .08	3.56 $\pm$ .04	3.59 $\pm$ .03

Table 3: Temporal structure scores of I3D models trained on UCF101, HMDB51, and Kinetics, evaluated on the Something-Something-V2 test set.

5. Conclusion

We presented *REPresentAtion bIas Removal* (REPAIR), a novel dataset resampling procedure for minimizing the representation bias of datasets. Based on our new formulation of bias, the minimum-bias resampling was equated to a minimax problem and solved through stochastic gradient descent. Dataset REPAIR was shown to be effective, both under controlled settings of Colored MNIST and in large-scale modern action recognition datasets. We further introduced a set of experiments for evaluating the effect of bias removal, which relates representation bias to the generalization capability of recognition models and the fairness of their evaluation. We hope our work will motivate more efforts on understanding and addressing the representation biases in different areas of machine learning.

Acknowledgement This work was partially funded by NSF awards IIS-1546305 and IIS-1637941, and NVIDIA GPU donations.

References

[1] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *European Conference on Computer Vision (ECCV)*, pages 771–787, 2018.

[2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *European Conference on Computer Vision (ECCV)*, pages 472–489, 2018.

[3] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4349–4357, 2016.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[5] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority oversampling technique. *Journal of Artificial Intelligence Research (JAIR)*, 16:321–357, 2002.

[6] Christoph Feichtenhofer, Axel Pinz, Richard P Wildes, and Andrew Zisserman. What have we learned from deep representations for action recognition? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7844–7853, 2018.

[7] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 259–268, 2015.

[8] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *International Conference on Computer Vision (ICCV)*, pages 2960–2967, 2013.

[9] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3165–3174, 2017.

[10] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual action recognition with r*cnn. In *International Conference on Computer Vision (ICCV)*, pages 1080–1088, 2015.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

[12] Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3315–3323, 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[14] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7366–7375, 2018.

[15] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *International Conference on Computer Vision (ICCV)*, pages 3192–3199, 2013.

[16] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(1):221–231, 2013.

[17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[18] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, pages 158–171, 2012.

[19] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *International Conference on Computer Vision (ICCV)*, pages 2556–2563, 2011.

[20] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision (IJCV)*, 64(2-3):107–123, 2005.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[22] Yingwei Li, Yi Li, and Nuno Vasconcelos. RESOUND: Towards action recognition without representation bias. In *European Conference on Computer Vision (ECCV)*, pages 513–528, 2018.

[23] Farzaneh Mahdisoltani, Guillaume Berger, Waseem Gharbieh, David Fleet, and Roland Memisevic. Fine-grained video classification and captioning. *arXiv preprint arXiv:1804.09235*, 2018.

[24] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1717–1724, 2014.

[25] Samuel Ritter, David GT Barrett, Adam Santoro, and Matt M Botvinick. Cognitive psychology for deep neural networks: A shape bias case study. In *International Conference on Machine Learning (ICML)*, pages 2940–2949, 2017.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[27] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Ad-*

vances in Neural Information Processing Systems (NIPS), pages 568–576, 2014.

[28] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[29] Pierre Stock and Moustapha Cisse. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *European Conference on Computer Vision (ECCV)*, pages 498–512, 2018.

[30] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In *Domain Adaptation in Computer Vision Applications*, pages 37–55. Springer, 2017.

[31] Antonio Torralba. Contextual priming for object detection. *International Journal of Computer Vision (IJCV)*, 53(2):169–191, 2003.

[32] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528, 2011.

[33] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

[34] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176, 2011.

[35] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *International Conference on Computer Vision (ICCV)*, pages 3551–3558, 2013.

[36] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision (ECCV)*, pages 20–36, 2016.

[37] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702, 2015.

[38] Richard Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning (ICML)*, pages 325–333, 2013.

[39] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[40] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *The European Conference on Computer Vision (ECCV)*, pages 803–818, 2018.