On The Impact of Machine Learning Randomness on Group Fairness

Prakhar Ganesh
National University of Singapore
Singapore
pganesh@comp.nus.edu.sg

Hongyan Chang
National University of Singapore
Singapore
hongyan@comp.nus.edu.sg

Martin Strobel
National University of Singapore
Singapore
martin.r.strobel@gmail.com

Reza Shokri
National University of Singapore
Singapore
reza@comp.nus.edu.sg

ABSTRACT

Statistical measures for group fairness in machine learning reflect the gap in performance of algorithms across different groups. These measures, however, exhibit a high variance between different training instances, which makes them unreliable for empirical evaluation of fairness. What causes this high variance? We investigate the impact on group fairness of different sources of randomness in training neural networks. We show that the variance in group fairness measures is rooted in the high volatility of the learning process on *under-represented groups*. Further, we recognize the dominant source of randomness as the stochasticity of *data order* during training. Based on these findings, we show how one can control group-level accuracy (i.e., model fairness), with high efficiency and negligible impact on the model's overall performance, by simply changing the data order for a single epoch.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **General and reference** → *Evaluation*.

KEYWORDS

neural networks, fairness, randomness in training, evaluation

ACM Reference Format:
Prakhar Ganesh, Hongyan Chang, Martin Strobel, and Reza Shokri. 2023. On The Impact of Machine Learning Randomness on Group Fairness. In *2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT '23), June 12–15, 2023, Chicago, IL, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3593013.3594116

1 INTRODUCTION

Machine learning models are shown to manifest and escalate historical biases present in their training data [1, 4, 16, 56]. Understanding these biases and the resulting ethical obligations have led to the rise of fair machine learning research [13, 15, 35]. However, recent work has observed high variance in fairness measures across multiple training runs, usually attributed to non-determinism in training (e.g., weight initialization, data reshuffling, etc.). These findings challenge the effectiveness of many bias mitigation algorithms [3, 46], and even the legitimacy of several fairness trends present in literature [49]. Thus, a reliable extraction of fairness measures requires accounting for the high variance due to randomness in the learning process to avoid lottery winners (see Fig. 1).

The standard solution to this concern is executing a large number of training runs with different randomness. However, such a solution creates huge computational demands when examining biases in neural networks. For instance, it costs about \$450K to train a model of similar quality as GPT-3 [54], and thus executing multiple training runs of such a model is not practical. But, are multiple identical runs essential? Can we instead find an efficient alternative to measure this variance? Our paper answers this critical yet unsolved question.

In this work, we perform an empirical investigation into the high fairness variance due to randomness in neural network training, with a diverse set of experiments on a multitude of settings, including different datasets across modalities, various fairness metrics, and changing hyperparameters and model architecture. More specifically, our empirical analysis answers the following questions

- **Is there a dominant source of randomness?** We show that the fairness variance observed in the literature is dominated by randomness due to data reshuffling during training. Reshuffling causes large changes in fairness even between consecutive epochs within a single run, while other forms of randomness have minimal influence.
- **Why are fairness measures highly sensitive to data reshuffling?** We show a higher vulnerability of minorities to changing model behavior, i.e., a higher prediction uncertainty for under-represented groups. This disparate prediction uncertainty between groups is reflected in any statistical fairness measure defined on model predictions.
- **How does data order impact fairness?** We demonstrate an immediate impact of the data order on fairness. That is, we show that a model's fairness score is heavily influenced by the most recent gradient updates, irrespective of the preceding training. We also demonstrate how to create custom data orders that can efficiently control group-level performances (and thus in turn, model fairness), with a minor impact on the overall accuracy.

FAccT '23, June 12–15, 2023, Chicago, IL, USA

ACM ISBN 979-8-4007-0192-4/23/06.
https://doi.org/10.1145/3593013.3594116

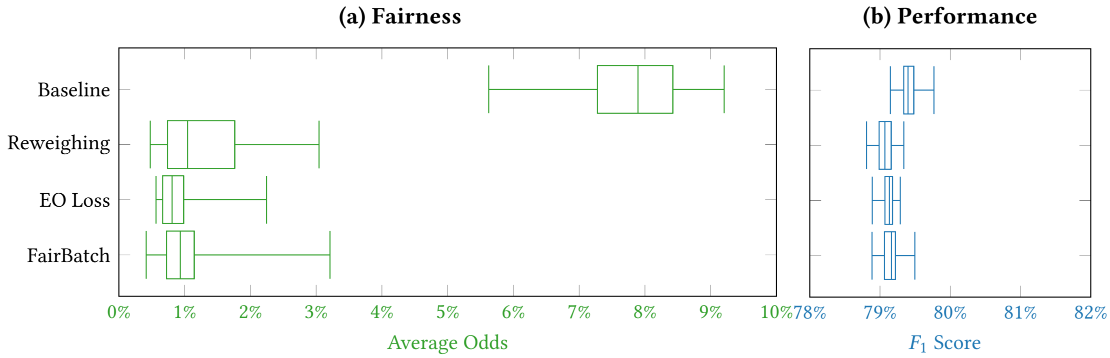

Figure 1: Variance of fairness and accuracy: (a) Fairness (average odds) has a high variance across multiple runs due to non-determinism in training. This variance persists even with state-of-the-art bias mitigation algorithms (*Reweighing [28]; Equalized Odds Loss [21]; FairBatch [44]*) (b) The overall performance (F_1 score), however, has a significantly smaller range of variance.

- **What are the practical implications?**
  - Given the immediate impact of data order on model fairness and the nature of data order reshuffling in neural network training, we propose that using fairness variance across epochs in a *single* training run is a good proxy to study fairness variance across multiple runs, thus reducing the computational requirements by a significant factor.
  - We also propose a custom data order that can improve model fairness within a single training epoch, and compete with existing bias mitigation algorithms. Interestingly, we show that similar custom data orders can also be created by adversaries to *freely control* fairness gaps in only a single epoch of training, even under explicit bias mitigation.

2 BACKGROUND AND RELATED WORK

In this section, we first introduce the relevant background in fair machine learning and randomness in neural network training. We then discuss the related work on the impact of randomness on fairness evaluation in deep learning.

2.1 Fairness in Machine Learning

Fair machine learning can be broadly divided into two categories, (i) group fairness [15], and (ii) individual fairness [19]. Group fairness relies on measuring the disparity between the average performance of protected groups against other privileged groups, and thus focuses on highlighting systematic bias against certain groups. Individual fairness instead relies on some form of similarity between individuals and requires consistency in the decision-making, i.e., similar individuals should be treated similarly.

In this work, we focus specifically on group fairness. Group fairness has a diverse set of definitions in the literature, usually chosen based on the stakeholders involved, known even to have opposing behavior in specific settings [38, 45]. We will rely on three commonly used group fairness metrics, i.e., demographic parity, average odds, and equal opportunity [25]. Demographic parity is the measure of disparity between the percentage of positive outcomes for each group, i.e., it does not allow model predictions to depend on sensitive attributes. Average odds (and its relaxed version, equal opportunity) is instead a measure of disparity between predictions for each group conditioned on the true labels, i.e., it does allow overall predictions to depend on sensitive attributes, but does not allow predictions for certain ground-truth labels to depend on sensitive attributes. Bias calculation and mitigation for group fairness have accumulated extensive literature in recent years, along with many open-source benchmarks [5, 8, 21, 43].

2.2 Randomness in Neural Network Training

Deep learning involves various forms of randomness that impact a neural network's path to convergence. This randomness during training can introduce noise into the optimization objective and works as a regularizer for the learning algorithm [40]. It makes the model prioritize generalization, avoid overfitting, escape local minima, and even speed up convergence [12]. Thus, randomness during training is integral to the success of neural networks, but its impact on model behavior needs to be carefully examined [7, 10].

Broadly, randomness in neural networks can be studied in the context of the following categories (see Fig. 2),

- **Data Splitting:** For any experimental setup in machine learning, the dataset under consideration is randomly divided into train-val-test (or just train-test) splits, to avoid information leakage and perform a fair evaluation.
- **Weight Initialization:** Weight initialization refers to the initial parameter vector that is the starting point for the gradient descent. Randomness in weight initialization is crucial for breaking the symmetry between model parameters and allows the neural network to learn complex functions [24].
- **Random Reshuffling:** Neural network training relies on gradient descent to optimize a chosen objective iteratively. Calculating the gradient over the entire dataset for every optimization step is expensive. A commonly adopted alternative is uniformly sampling a subset of the dataset to approximate the gradient, known as stochastic gradient descent (SGD). In practice, it has been shown that instead of uniform sampling, SGD can also be implemented by simply traversing a random order, i.e., random reshuffling, of the training data [12, 36, 39].

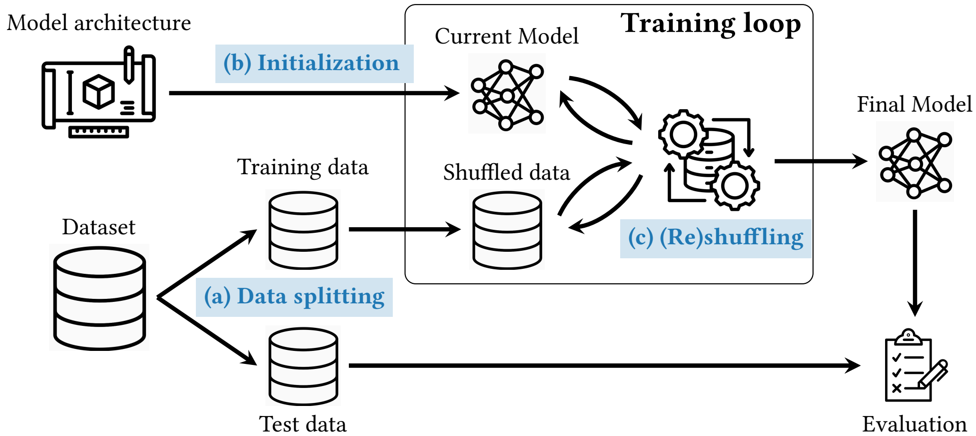

Figure 2: Sources of randomness during training: Randomness is introduced at several points during the training of a neural network. (a) Data splitting before training is important to avoid information leakage during evaluation, and involves randomness for the same. (b) Randomness in weight initialization at the start of the training is necessary to break symmetry and learn complex representations. (c) Random reshuffling at every epoch implements stochastic gradient descent (SGD) in practice. There are many other forms of randomness in neural network training. Yet, most are only used for certain specific settings.

- **Other Sources of Randomness:** Several additional components in the learning algorithm can introduce further randomness. These components are not standard, but are introduced in special cases to achieve specific objectives. For example, data augmentation to increase dataset size [29], dropout to perform regularization [55], gradient noise for private training [14], etc.

2.3 High Variance in Fair Deep Learning

There has been a growing awareness of high variance in fair deep learning, associated with non-determinism in model training or the underlying implementation [3, 20, 41, 46, 49], and the uncertainty of existing results in the literature.

Soares et al. [49] investigate the relationship between various algorithmic choices and the corresponding fairness variance, in large language models. They study the correlation of fairness with model size and found no obvious trends, as opposed to various claims previously made in literature [6, 26]. They also found that fairness is heavily affected by the random seed, i.e. simply changing the randomness can cause a huge variance in fairness.

Sellam et al. [46] have trained and released 25 pre-trained BERT checkpoints, each trained from scratch under identical settings but with a different random seed. They also analyze the variance of model fairness and the impact of commonly used bias mitigation algorithms on downstream tasks when starting with different pre-trained models. They show significant variance across changing random seeds and question the value of such mitigation techniques.

Amir et al. [3] revisit bias mitigation techniques in clinical texts and show a lack of statistically significant improvement after accounting for non-determinism in training. Friedler et al. [20] explore the stability of fairness under a rarely studied source of randomness, i.e. data splitting, and show notable impact on fairness evaluation.

While existing literature focuses on exploring the impact of high fairness variance in bias evaluation, we instead focus on investigating its source. Furthermore, we propose to move away from the practice of simply executing multiple runs to capture fairness variance and instead provide a computationally efficient proxy.

3 PROBLEM STATEMENT

We start by formally defining the problem statement and detailing our experiment setting for the rest of the paper.

3.1 Neural Network Training

Most machine learning algorithms can be abstracted down to an optimization problem for a given objective, usually a loss function. More specifically, for a training dataset $(x, y) \in \mathcal{D}$, a family of hypothesis functions $\mathcal{F}$, and a loss function $\mathcal{L}$, the optimization goal for the learning algorithm can be defined as,

$$f^* \leftarrow \underset{f \in \mathcal{F}}{\text{argmin}} \sum_{(x,y) \in \mathcal{D}} \mathcal{L}(f(x), y) \tag{1}$$

The above formulation of the learning objective is known as empirical risk minimization (ERM) [53]. However, finding a global optimum for ERM in deep learning is typically intractable, due to the high dimensional, non-convex formulation of neural networks. Neural networks are instead trained iteratively, starting with a randomly sampled function f_0, refining the model with a learning algorithm $\mathcal{A}$ for T epochs, to finally output the trained model f_T. The learning algorithm at every epoch t takes in the current model, complete training data $\mathcal{D}$, and a number of hyperparameters ξ (e.g., batch size, learning rate, etc.), to progressively improve the model by a single epoch of training. The learning algorithm can contain various sources of randomness, as discussed above. In our work, we will focus on two standard forms of randomness found in every neural network training, i.e., weight initialization and random reshuffling of data order at every epoch. More specifically, neural network training can be defined as,

$$f_t := \mathcal{A}(f_{t-1}, \mathcal{D}, \xi, r_s, t) \qquad f_0 \sim \mathcal{F}; r_s \sim R \tag{2}$$

The function f_0, i.e., the weight vector initialization in a parameterized neural network, is randomly sampled from pre-defined distribution $\mathcal{F}$, and the random seed for reshuffling r_s is sampled from a uniform distribution R. Note that both random seed r_s and epoch number t are together responsible for the data shuffling of epoch t. Thus, for fixed reshuffling (i.e., fixed r_s), the data order is still shuffled at every epoch during a single training run but is the same at any epoch t across two different training runs. More details on the random seed setup can be found in Appendix A.

3.2 Metrics and Variance

A model f's performance can be evaluated using its outputs on the test dataset. We will stick to the commonly used binary classification and binary sensitive attribute $a \in \{0, 1\}$ setting in fairness literature for the rest of the discussion. In the main paper, we will rely on F_1 Score and average odds (AO) to measure the model's average performance and group fairness respectively. AO can be empirically interpreted as the average disparity between separately calculated *true positive rates* (TPR) and *false positive rates* (FPR) of various groups [25]. The metrics are defined as

$$F_1(f, \mathcal{D}) := \frac{2TP}{P + PP} = \frac{2\sum_{\mathcal{D}^e} \mathbb{1}_{[f(x)=y \wedge y=1]}}{\sum_{\mathcal{D}^e} \mathbb{1}_{[y=1]} + \sum_{\mathcal{D}^e} \mathbb{1}_{[f(x)=1]}} \tag{3}$$

$$AO(f, \mathcal{D}) := \frac{(\Delta TPR + \Delta FPR)}{2} \tag{4}$$

$$= \frac{1}{2} \sum_{\substack{h=\{0,1\} \\ y=h}} \left| \frac{\sum_{\mathcal{D}^e} \mathbb{1}_{[f(x)=1 \wedge a=0]}}{\sum_{\mathcal{D}^e} \mathbb{1}_{[a=0]}} - \frac{\sum_{\mathcal{D}^e} \mathbb{1}_{[f(x)=1 \wedge a=1]}}{\sum_{\mathcal{D}^e} \mathbb{1}_{[a=1]}} \right|$$

where $\sum_{\mathcal{D}^e}$ is the sum over all data points in the test set $\mathcal{D}^e$, i.e., $\sum_{(x,y,a) \in \mathcal{D}^e}$, and $\mathbb{1}_{[z]}$ is an indicator function which is 1 when the boolean expression z is true, and 0 otherwise. We also include additional experiments for two more fairness measures, equal opportunity (EOpp) and demographic parity (DP) in Appendix I. Moreover, we will show that the non-determinism in fairness originates from high prediction uncertainty for minority (Section 5), and thus will be reflected in any fairness metric defined on these predictions. We report all metrics in percentage.

At the heart of our work is the study of fairness variance across model checkpoints. We define variance across multiple runs and variance across epochs in a single run as,

$$Var_{F_1}^{runs}(\mathcal{A}, T) := \underset{f_0 \sim \mathcal{F}; r_s \sim R}{Var}(F_1(f_T)), \tag{5}$$

$$Var_{F_1}^{epochs}(\mathcal{A}, f_0, r_s, T_1, T_2) := \underset{t \in [T_1, T_2]}{Var}(F_1(f_t)), \tag{6}$$

$$Var_{AO}^{runs}(\mathcal{A}, T) := \underset{f_0 \sim \mathcal{F}; r_s \sim R}{Var}(AO(f_T)), \tag{7}$$

$$Var_{AO}^{epochs}(\mathcal{A}, f_0, r_s, T_1, T_2) := \underset{t \in [T_1, T_2]}{Var}(AO(f_t)), \tag{8}$$

Existing work in the literature has shown high variance in fairness scores across multiple runs $Var_{AO}^{runs}(\mathcal{A}, T)$. In our work, we first decouple the impact of two standard sources of randomness, i.e., study $\underset{f_0 \sim \mathcal{F}}{Var}(F_1(f_T))$ and $\underset{r_s \sim R}{Var}(F_1(f_T))$ separately. In doing so, we find high variance in fairness scores even across epochs in a single training run (Section 4), and thus further study variance across epochs in fairness scores $Var_{AO}^{epochs}(\mathcal{A}, f_0, r_s, T_1, T_2)$. Note that for F_1 score, variance across multiple runs $Var_{F_1}^{runs}(\mathcal{A}, T)$ and across epochs in a single run $Var_{F_1}^{epochs}(\mathcal{A}, f_0, r_s, T_1, T_2)$ are both relatively stable. Unless otherwise specified, we train our models for a total of $T = 300$ epochs, and we measure variance across epochs from $T_1 = 100$ to $T_2 = 300$. We make this choice because the models have converged to stable accuracy before epoch 100 (refer to the training curve in Appendix C for more details).

3.3 Datasets and Models

We will conduct our investigation on ACSIncome and ACSEmployment tasks of the Folktables dataset [17], and binary classification of the 'smiling' label in CelebA dataset [31], with perceived gender (Female vs. Male) as the sensitive attribute for all datasets. For CelebA, input features are obtained by passing the image through a pre-trained ResNet-50 backbone and extracting the output feature vector. More details on the datasets are provided in Appendix B.

We train a feed-forward network with a single hidden layer of 64 neurons and *ReLU* activation, and train the model with *cross-entropy (CE)* loss for $T = 300$ epochs at batch size 128 and learning rate $1e{-3}$, in all our experiments unless specified otherwise. Note that while we measure fairness scores in our experiments, we do not explicitly train the models with any fairness constraints (except in Section 7.2 when training baseline bias mitigation algorithms). We also include additional experiments by changing training hyperparameters, i.e., batch size, learning rate, and model architecture, in Section 4.2 (and Appendix G). We use a train-val-test split of $0.7 : 0.1 : 0.2$, and maintain the same split throughout all our experiments, i.e. we do not consider potential randomness due to data splitting. All our experiments and evaluations are performed only on the test split. We will focus primarily on the ACSIncome task in the main text, while additional experiments on CelebA and the ACSEmployment task are included in Appendix F.

4 THE DOMINANT SOURCE OF RANDOMNESS

In this section, we move past the observation that different training runs lead to different outcomes, and investigate the high fairness variance by studying and contrasting the two canonical sources of randomness.

4.1 Impact of Weight Initialization and Random Reshuffling on Fairness Variance

We start by decoupling the two sources of randomness inherent to the widely adapted SGD, i.e., weight initialization and random reshuffling, and study their impact on fairness variance separately in Fig. 3. We collect average odds (AO) and F_1 score at epoch 300 for 50 unique training runs each while, *(i)* allowing for both sources of randomness, *(ii)* changing only the weight initialization while keeping the random reshuffling fixed, and *(iii)* changing only the random reshuffling while keeping the weight initialization fixed, respectively. The large range of fairness scores reported by allowing for both sources of randomness in Fig. 3(a) represents the variance observed in the existing literature. Interestingly, when these sources are examined separately, the variance under fixed initialization in Fig. 3(c) is equivalently large but the variance under fixed reshuffling in Fig. 3(b) drops significantly. It is clear that fairness variance

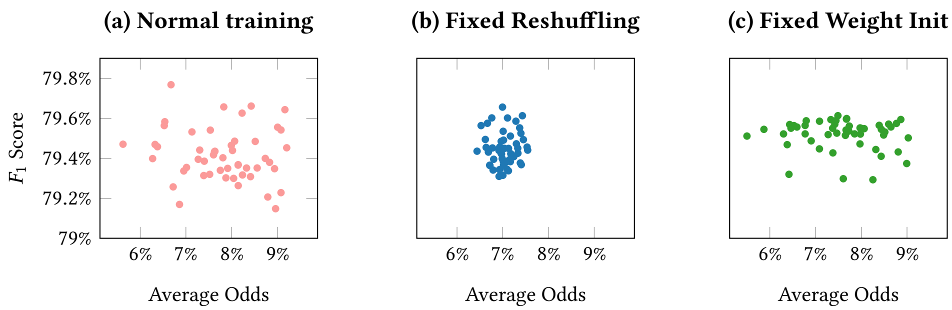

Figure 3: Decoupling the effect of randomness in weight initialization and reshuffling: (a) Variance in average odds (AO) by allowing both sources of randomness simultaneously represents the fairness variance in existing literature. (b) We see a significant drop in variance if we change only the weight initialization while keeping the reshuffling fixed. (c) However, we observe high range of variance by changing only the reshuffling, even for a fixed weight initialization. These results suggest reshuffling of the data order as the dominant source of fairness variance, with little influence from weight initialization.

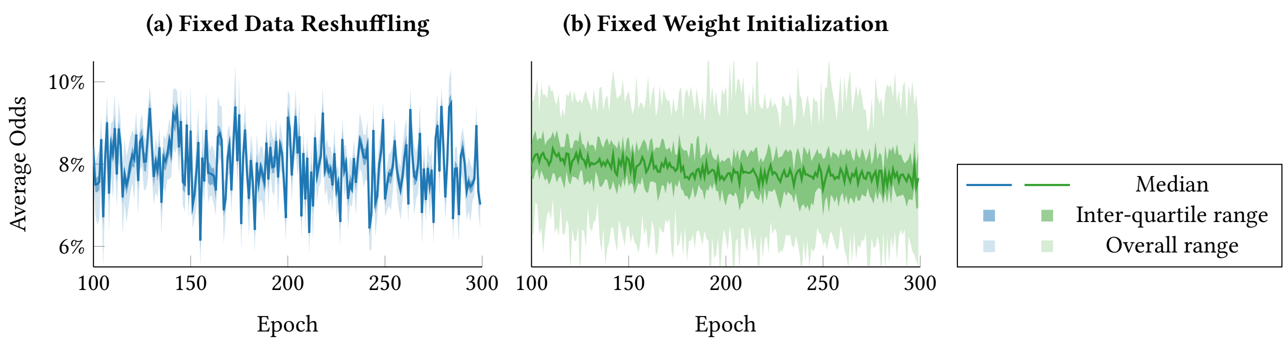

Figure 4: Training dynamics under fixed weight initialization and reshuffling: Median, inter-quartile range, and overall range of average odds across 50 training runs while keeping the data reshuffling or the weight initialization fixed respectively. (a) Despite different initializations, models with fixed data reshuffling have very little variance across training runs, but high variance across epochs. This highlights the dominant impact of random reshuffling on model fairness. (b) High variance even across training runs at the same epoch under fixed weight initialization further supports our claim.

originates from data order shuffling, while randomness in weight initialization has minimal impact.

To further probe the difference between these two sources of randomness, we study the training dynamics of the previous set of models across epochs, instead of just the final model checkpoint, in Fig. 4. We plot the median, inter-quartile range, and overall range of average odds (AO) across the complete set of 50 training runs from epoch 100 to 300 in the two isolated settings from the previous experiment. We find a high correlation in fairness scores across training runs with fixed data reshuffling (average pairwise pearson coefficient ≈ 0.94), which supports our observations of low variance in fairness scores at the final epoch in Fig. 3(b). Furthermore, there is a lack of any reasonable correlation between fairness scores of training runs with fixed weight initialization (average pairwise pearson coefficient ≈ 0.04). Interestingly, the high fairness variance across epochs inside a single training run in Fig. 4(a) closely matches the variance that we observe across multiple training runs in Fig. 4(b). In other words, for fixed data reshuffling the average odds value at any epoch is almost the same between different training runs, but the average change even between consecutive epochs is large, while for fixed weight initialization, even the variance between runs is quite high.

4.2 Dominance of Random Reshuffling across Datasets, Metrics and Hyperparameters

We extend our previous experiment and calculate correlation across multiple runs for additional datasets, fairness measures, as well as hyperparameter choices of batch size, learning rate, model architecture, and dropout regularization with different dropout rates in Table 1. Here we measure the correlation (i.e., average pairwise pearson coefficient) across 50 training runs in each setting for fixed data shuffling and fixed weight initialization. It is clear from the results that even under diverse settings, the correlation between multiple runs with fixed data reshuffling is significantly high, while the correlation with fixed weight initialization is close to zero.

In addition to the overall trends supporting our initial claim, individual trends in Table 1 under various settings are also quite interesting. The correlation score for fixed weight initialization under hyperparameters that induce noisier training (i.e., smaller

Table 1: Average pairwise pearson coefficient for correlation across multiple runs: Fixed random reshuffling (RR) (i.e., changing only the weight initialization) has a high correlation score across multiple runs, while fixed weight initialization (WI) (i.e., changing only the random reshuffling) has a correlation score close to zero, which establishes the dominance of data reshuffling on fairness. These trends exist across different datasets, fairness metrics, and hyperparameter choices.

(a) Different Datasets

	Fixed RR	Fixed WI
ACSIncome	.94	.04
ACSEmployment	.89	.01
CelebA	.92	.01

(b) Different Fairness Metrics

	Fixed RR	Fixed WI
Average Odds	.94	.04
Equal Opportunity	.94	.05
Demographic Parity	.96	.03

(c) Changing Hyperparameters

	Fixed RR	Fixed WI
Default Hyperparameters	.94	.04
Batch Size = 16	.95	.00
Learning Rate = 0.01	.93	.00
Arch = {2048, 64}	.92	.00
No Dropout	.94	.04
Dropout Rate = 10%	.88	.03
Dropout Rate = 20%	.85	.04
Dropout Rate = 30%	.80	.13

batch size, larger learning rate, etc.) drops even lower than the default setup. Note that we do not report results for a bigger batch size or a smaller learning rate, as these models face issues with convergence even after 300 epoch of training (see Appendix G for more details), which further indicates the need of randomness and noise in the learning algorithm to facilitate better and faster convergence. However, this randomness will also create a dominant dependence of model fairness on data reshuffling, as we study in our work. We also see the trends diminishing (although still clear) with higher dropout rates. This suggests that appropriate regularization during training can indeed, to some extent, reduce the impact of randomness in training (or more specifically, data reshuffling) on fairness scores. We also provide detailed results for each hyperparameter setting above in Appendix G, H.

Takeaway 1: *Random reshuffling of data order during training is the dominant cause of high fairness variance as seen in the literature, while randomness in weight initialization has minimal influence.*

5 WHY IS FAIRNESS HIGHLY SENSITIVE TO RANDOMNESS?

In the previous section, we showed the dominant impact of data reshuffling on model fairness. In this section, we show that its in fact the imbalance in the underlying data distribution for training which creates high volatility in predictions for the minority. Thus, groups with smaller representations are more significantly influenced by the randomness in reshuffling, resulting in high fairness variance.

5.1 Changing Predictions Across Epochs

We observed high variance in model fairness even between consecutive epochs during training (see Fig. 4). The changing predictive behavior of neural networks beyond training loss convergence is not surprising, and has been studied extensively in literature [27, 30, 51, 52]. As we are concerned with the fairness of the final decisions made by the model, we will focus on a change in the model's discrete output class when discussing changing predictions. More specifically, a model is said to have undergone a change in prediction for some input x during epoch t, if $f_t(x) \neq f_{t-1}(x)$, where $f_t(x)$ is the output class when passing the input x through the model checkpoint at the end of epoch t. While these changing predictions maintain an overall stable average performance, they can still have a disparate impact on individual groups, the exact characteristics of which are less known.

We study this instability by investigating individual data points which change their predictions. We plot the dataset distribution across groups in Fig. 5(a) and the percentage of data points from each group that changed their prediction at least once between epochs 100 and k, where we gradually increase the value of k, in Fig. 5(b). Clearly, the trends in the percentage of unique data points with changing predictions mirror the representation of each group in the original dataset, i.e. the groups which are represented the least are the most vulnerable to changing model behavior. For example, positive labels from the group Female are severely under-represented and consequently have almost twice the percentage of unique examples with changing predictions than any other groups.

5.2 Disparate Prediction Uncertainty

Higher vulnerability to changing discrete predictions for minorities can be interpreted as an indication of higher uncertainty in the underlying model predictions. To further probe the disparate model behavior across different groups, we record the cumulative distribution of prediction uncertainty for each group separately in Fig. 6. We rely on two commonly used methods to measure prediction uncertainty, i.e., *(i)* monte-carlo dropout [22], and *(ii)* training a bayesian neural network [11]. We execute 1000 forward passes for each method and record the standard deviation in outputs as the prediction uncertainty. Despite different distribution trends, both methods highlight the higher prediction uncertainty of minorities. As expected, the order of prediction uncertainty across groups follows the training data distribution (Fig. 5(a)), i.e. groups with a larger representation in the training data have smaller number of examples with large prediction uncertainty, which is quite intuitive.

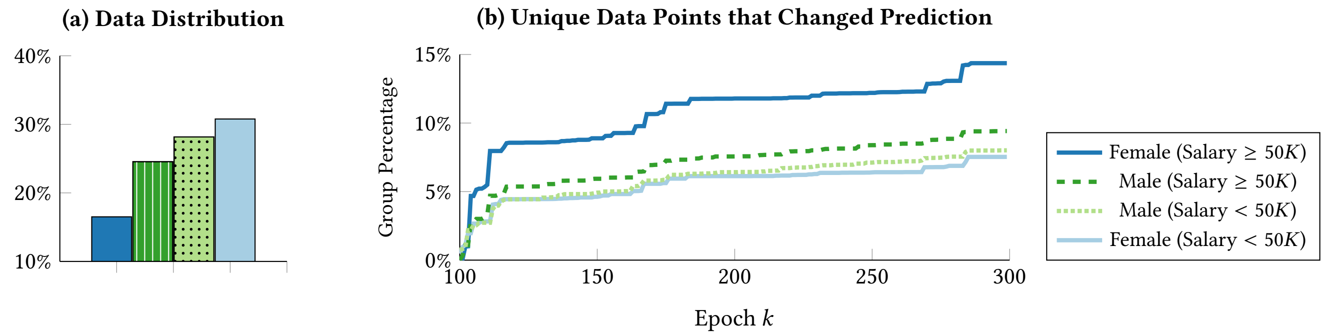

Figure 5: Disparate percentage of changing predictions across epochs: (a) The underlying data distribution of ACSIncome shows positive labels from group Female as an under-represented minority. (b) Total percentage of unique data points from each subgroup that change prediction across epochs follow the opposite order of their representation in the training data. These results highlight subgroups with least representation being the most vulnerable to changing predictions.

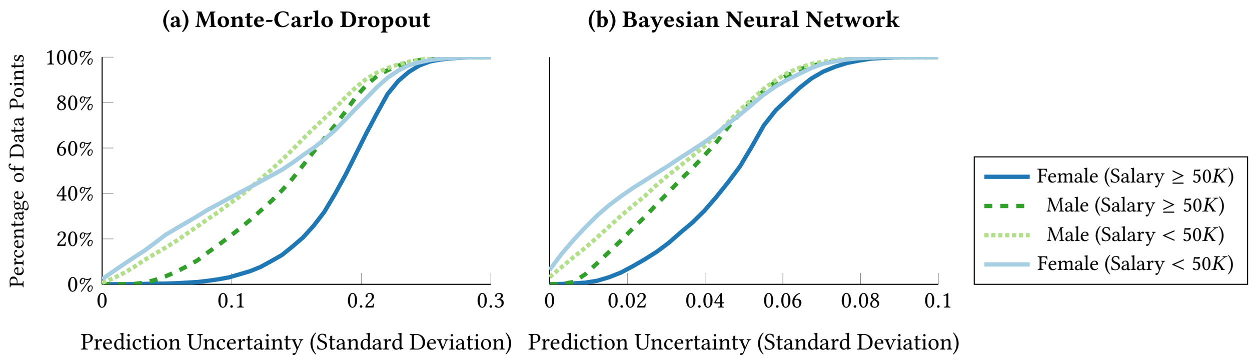

Figure 6: Normalized cumulative distribution of prediction uncertainty for various groups: The distribution of the minority group (Female with Salary $\geq$ 50K) is significantly more skewed towards higher uncertainty than any other group, i.e. the minority contains far more percentage of data points with high prediction uncertainty than the majority.

As the model has skewed cumulative distribution towards uncertainty for the minority, any fairness metric defined on the output of such a model will also reflect this instability, and thus manifests as fairness variance in existing literature [3, 46, 49].

Takeaway 2: *Under-represnted groups have higher prediction uncertainty in the final trained model, and thus predictions for data points from such minorities are more sensitive to the randomness.*

6 IMPACT OF DATA ORDER ON FAIRNESS

In Section 4, we observed the dominance of reshuffling on fairness variance. Data order during training governs gradient updates and thus its impact on fairness is unsurprising [32, 37, 42, 47, 48, 50]. Even under randomly shuffled data order, neural networks are known to undergo changes in predictive behavior during training [27, 30, 51, 52]. However, its the immediacy of the impact of data order that is surprising and a novel observation of our work. We now study the impact of data order in a single epoch on fairness.

6.1 Data Order's Immediate Impact

To study the immediacy and characteristics of the impact of data order on model fairness, we fine-tune a set of already converged model checkpoints for a common sequence of b batches and record the fairness variance across checkpoints for different values of b in Fig. 7. This allows us to measure fairness variance across models which have experienced the same most recent b gradient updates. The choice of the common sequence of b batches fed to the model was done by separately training a model and choosing the suffix of data order corresponding to epochs with the best and worst fairness scores on the validation set. As the number of fixed batches b increases, the fairness variance decreases, and it is clear from the results that the impact of data order on fairness is quite immediate (an epoch of ACSIncome dataset is b = 1070 batches). Moreover, the resulting fairness is also characteristically stable for a specific data order, i.e. batches taken from the suffix of the data order corresponding to the best fairness epoch of an individual training run also helps fine-tune all other checkpoints towards the same best fairness, and vice-versa for the worst fairness epoch.

The set of checkpoints were chosen by sampling 1000 different checkpoints from epochs 100 to 300 of 50 different training runs while allowing for both forms of randomness simultaneously. The checkpoints were chosen in this manner to create a diverse training history, and show that these models achieve the same fairness

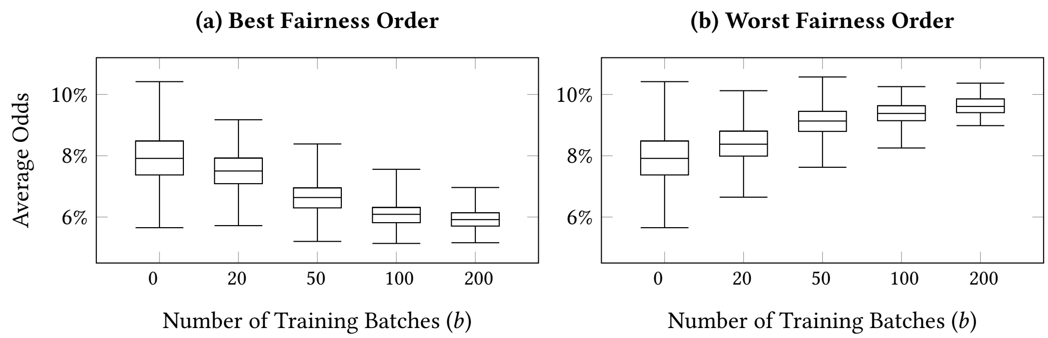

Figure 7: Fairness variance under common b most recent gradients: Average odds stabilize as the number of most recent common training batches b increases, highlighting the immediate impact of these gradient updates on model fairness. Moreover, it even predictably stabilizes to low or high fairness based on the corresponding data order from the epoch with best or worst fairness.

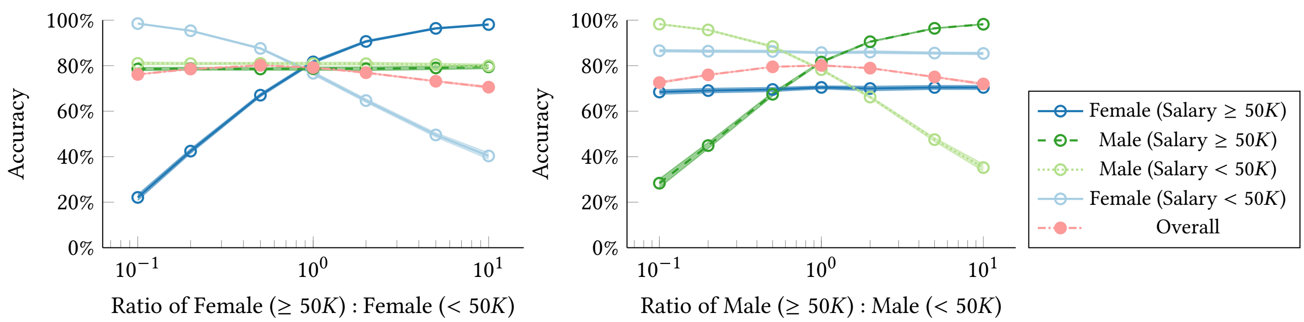

Figure 8: Manipulating group level accuracy with data order: We show how to control group level accuracy by changing data distribution of the most recent gradient updates, tested separately for ratio between positive and negative labels for group Female, and group Male, respectively, while keeping other ratios fixed. In only a single epoch of training, we are able to manipulate the group level accuracy trade-off, with relatively small impact on overall accuracy.

scores when fine-tuned on a common set of batches, irrespective of training prior to those updates. We also extend our experiment to show the same behavior even for batches from any random data order, instead of deliberately chosen data orders as above, in Appendix D. One possible explanation of this immediate impact is the presence of no energy valleys in deep learning loss landscape between minima of separately trained models [18, 23]. Another possible explanation can be built on a recent line of work that shows there is only one functionally unique minima in loss landscape of neural networks, while all other minima simply contain permutation or scale symmetries of the same set of models [2].

6.2 Manipulating Group Accuracy Distribution with Data Order

In the previous section, we saw a stable relationship between data order and the resulting fairness score. However, the data orders in the experiment above were sampled from a set of random data orders. We now show that it is possible to create our own custom data order to achieve any target fairness score. We hypothesize that since the data order in the most recent batches has an immediate impact on model fairness, the distribution in these batches must be temporarily changing the loss landscape and nudging the group-level accuracy. This could allow us to manipulate group-level accuracy in only a single epoch of fine-tuning.

To test this hypothesis, we fine-tune a set of 50 already converged models for exactly a single epoch on custom data orders with chosen distributions and record group-level accuracy in each setting in Fig. 8. To create this custom data order, we start by fixing the ratio between different groups and then form batches for the data order suffix in this exact ratio until we run out of data points (which will happen for any ratio that is not the dataset distribution). The excess data points are then shuffled randomly and placed at the prefix of data order. We do two sets of isolated experiments, by changing the ratio between positive and negative labels for group Female (Fig. 8(a)) and group Male (Fig. 8(b)) respectively, while keeping other ratios fixed. It is clear that by manipulating the data distribution in the most recent gradient updates, we can control the group-level accuracy of the model. While the overall accuracy drops noticeably for extreme ratios, it does not change much in the middle despite significant variance in group-level performances.

Note that our custom data order still has the same distribution as the original dataset, i.e., we have not changed the distribution of the complete data order, but only moved around the data points to change the distribution of the data order suffix. These results further

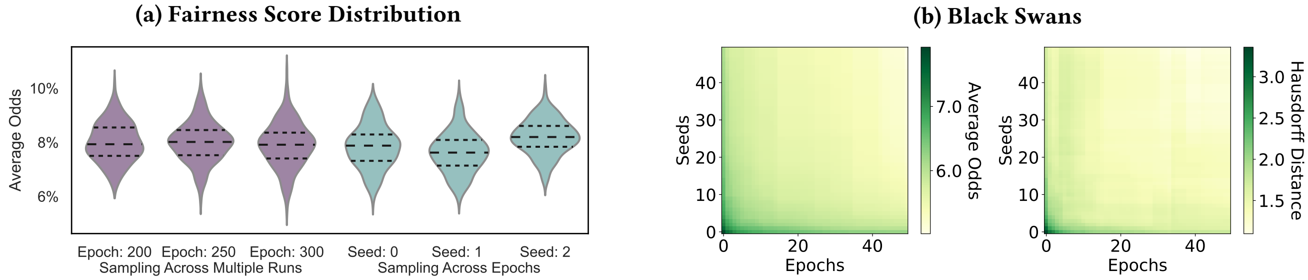

Figure 9: Fairness variance across multiple runs vs across epochs in a single run: (a) Fairness scores (average odds) across multiple training runs and across epochs in a single training run have similar empirical distributions. (b) Quality of black swans (i.e. extremely rare checkpoints) improves with more checkpoints collected, either in terms of fairness (the lowest achievable average odds score) or the trade-off between overall performance and fairness (the Hausdorff distance to the best pareto front). This improvement occurs at the same rate, irrespective of sampling checkpoints across multiple random seeds (x-axis) or multiple epochs in a single seed (y-axis). But, sampling across epochs is significantly cheaper, providing a highly efficient alternative to executing multiple runs.

strengthen our claim on the immediate impact of the most recent gradient updates on model fairness and group-level accuracy. *Fairbatch* [44], a recently proposed bias mitigation algorithm, follows a similar formulation (although it additionally changes the overall distribution). Fairbatch creates batches with a fixed ratio between groups, and this ratio is continuously optimized to counter the existing bias in the model. Our results not only explain their success, but also state that instead of regularly adapting the distribution to compensate for the model bias (and oversampling/undersampling certain groups), one can directly use the desired distribution to create a custom data order and the model will adapt to it immediately.

Takeaway 3: *The training data order has an immediate impact on the model's fairness scores. That is, the data distribution in the most recent gradient updates can control the model's group level accuracy in only a single epoch of fine-tuning.*

7 APPLICATIONS OF THE IMPACT OF DATA ORDER ON FAIRNESS

With a better understanding of the impact of data order on model fairness and how to control it, we now explore some practical applications of our observations.

7.1 Capturing Fairness Variance in a Single Run

We now return to our original problem of capturing fairness variance without wasting computing resources on a large number of training runs. We saw an immediate impact of data order on model fairness, which shows that fairness variance across multiple training runs can instead be studied as simply the randomness in data order at their last epochs. As these orders randomly reshuffled, their distribution across multiple trainings should be the same as their distribution across epochs in a single training run. Thus, we propose evaluating fairness of intermediate checkpoints in a single training run as a proxy for multiple runs.

To test the similarity in both distributions, we simply plot the distribution of fairness scores for checkpoints across 200 training runs (for three different stopping epochs) and across epochs 100 to 300 in a single training run (for three different training runs) in Fig. 9(a). Clearly, the empirical distribution of fairness across multiple training runs closely matches the distribution across a single training run. We also perform the Kolmogorov–Smirnov (KS) test [33] to match sampling across multiple runs (at epoch 300) and sampling across epochs (for a single training run with seed 0). The maximum difference in empirical CDF of the two distributions was only 0.07, and the KS test gave the p-value of 0.712, i.e. the probability of the hypothesis that both set of fairness scores were sampled from the same underlying distribution.

Furthermore, we also test the quality of black swans, i.e. the best models under certain quality measure, as a function of number of unique training runs and number of epochs evaluated per training run, in Fig. 9(b). For all $(t, s) \in [1, 50]$, we perform s unique training runs (while changing both forms of randomness, i.e., weight initialization and random reshuffling), and evaluate the model for last t epochs in each training run, thus accumulating a total of $t * s$ checkpoints. We then calculate two different quality measures for these set of checkpoints, i.e., *(i)* the best fairness achieved across all checkpoints, and *(ii)* the Hausdorff distance [9] of the Pareto-front (including both fairness and F_1 scores) from the best achievable Pareto-front, i.e., for $t = 50; s = 50$. Finally, the experiment is repeated and averaged 50 times to compensate for randomness in the s training runs. Interestingly, the black swans for both quality measures show no significant distinction between increasing the number of training runs or evaluating multiple epochs per training run, i.e., sampling more checkpoints in either direction gives us similar improvements.

It is clear from our results that the commonly used method to capture fairness variance in literature ($t = 1; s = 50$) is highly inefficient use of computing resources, and one can extract the same quality of black swans (and overall fairness variance) by simply observing fairness across multiple checkpoints in a single training run ($t = 50; s = 1$), which would require 50 times less computation. With these experiments, we showed direct benefits of evaluating multiple epochs in a single training run, saving huge amounts of resources and time in capturing the overall fairness variance.

Takeaway 4: *Fairness distribution across multiple runs is empirically the same as that across epochs within a single run.*

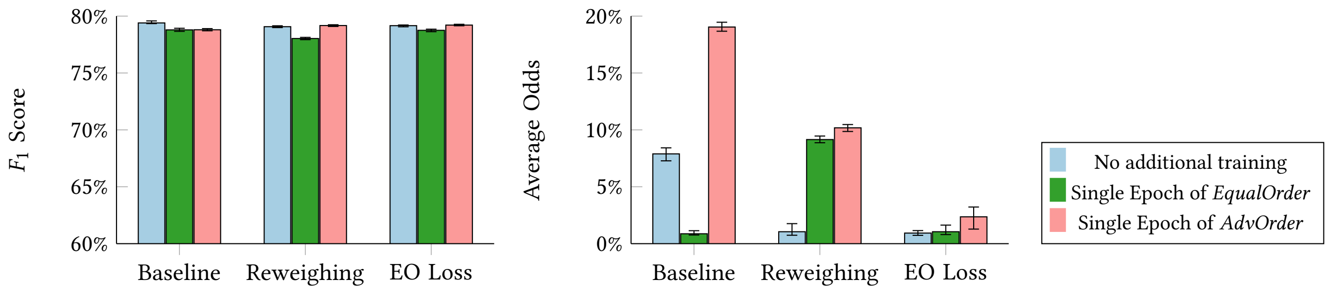

Figure 10: Comparing data order manipulation with other bias mitigation algorithms: Under baseline training setup, by changing the data order for just a single epoch of fine-tuning, *EqualOrder* gets competitive performance to commonly used bias mitigation methods. Similarly, *AdvOrder* gets significantly worse fairness than even the baseline. For Reweighing, we see an increase in bias even with *EqualOrder*, as the beneficial weighing of minority creates bias towards the majority. On the other hand, even reweighing is not enough to counter the effects of *AdvOrder*. Finally, equalized odds loss is capable of dynamically adapting to the model's changing predictive behavior, yet we still observe increase in bias under *AdvOrder*.

7.2 Bias Mitigation via Data Order Manipulation

To measure the effectiveness of our group accuracy manipulation, we extend the discussion from Section 6.2 for two special ratios, 1 : 1 and 1 : 3 between positive and negative labels of group Female (for ACSIncome dataset), and call them *EqualOrder* and *AdvOrder* respectively. More specifically, we fine-tune converged models with a single epoch of *EqualOrder* (and *AdvOrder*), and record F_1 score and average odds in Fig. 10. We perform experiments with three unique setups, (i) Baseline training, (ii) Reweighing [28], a data pre-processing which weighs every label-group pair based on its representation in the overall dataset, and (iii) Equalized Odds Loss [21], an in-processing loss function to nudge the model towards fair predictions. By training with *EqualOrder* for a single epoch, the baseline model achieves competitive fairness scores to other bias mitigation algorithms. On the other hand, using *AdvOrder* can further push the model bias, emphasizing the adversarial power of data ordering, even in presence of explicit mitigation algorithms.

Notably, reweighing suffers from an unexpected high bias even under *EqualOrder*, as the combination of ideally distributed data order suffix along with increase in the minority data weights pushes the model towards significant unfair behavior against the majority (as opposed to against the minority in all other results). Moreover, equalized odds loss shows controlled damage under *AdvOrder* due to the loss function regularly adapting to the degrading behavior, but the unfairness still increases. *AdvOrder* is dangerous as it still maintains the overall accuracy, but favors the majority. We can force even worse fairness gaps by pushing the ratio to its extreme, however that will impact the model's overall accuracy. These results cement the effectiveness of manipulating group level accuracy by controlling the data order for just a single epoch of fine-tuning.

Takeaway 5: *A data order with a balanced suffix can significantly improve in fairness scores. Similarly, even bias mitigation algorithms can fail when trained with an adversarial data order.*

8 CONCLUSION

Fairness variance due to changing randomness in deep learning has raised concerns regarding the reliability of existing results in literature [3, 46, 49]. In our work, we took a close look at various sources of randomness, and found a dominant impact of data order on model fairness, which we showed was in turn due to a higher prediction uncertainty of the trained model on under-represented groups in the dataset. We further demonstrated that the distribution seen by the model in the most recent gradient updates can be easily exploited to achieve desirable group-level accuracy behavior, and proposed several practical applications of this immediate impact of data order on model fairness, including a highly efficient alternative to executing multiple training runs when studying fairness variance due to randomness in training.

In our work, we focused only on the discrete decisions made by the model, as we were investigating the impact of non-determinism in model training on its fairness. However, further extensions of this discussion to trends in the internal state of the learned model can reveal even granular characteristics, and has potential application in similar fields of research, for example, understanding high variance in out-of-distribution generalization [34], exploiting model multiplicity under various settings [10], and many more.

ACKNOWLEDGMENTS

This research is supported by Google PDPO faculty research award, Intel within the www.private-ai.org center, Meta faculty research award, the NUS Early Career Research Award (NUS ECRA award number NUS ECRA FY19 P16), and the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore.

REFERENCES

[1] Mohsen Abbasi, Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2019. Fairness in representation: quantifying stereotyping as a representational harm. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 801–809.
[2] Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. 2022. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836* (2022).

[3] Silvio Amir, Jan-Willem van de Meent, and Byron C Wallace. 2021. On the Impact of Random Seeds on the Fairness of Clinical Classifiers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 3808–3823.
[4] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *Calif. L. Rev.* 104 (2016), 671.
[5] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. 2018. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. *arXiv preprint arXiv:1810.01943* (2018).
[6] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 610–623.
[7] Steven Bethard. 2022. We need to talk about random seeds. *arXiv preprint arXiv:2210.13393* (2022).
[8] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. 2020. Fairlearn: A toolkit for assessing and improving fairness in AI. *Microsoft, Tech. Rep. MSR-TR-2020-32* (2020).
[9] T Birsan and Dan Tiba. 2005. One hundred years since the introduction of the set distance by Dimitrie Pompeiu. In *IFIP Conference on System Modeling and Optimization*. Springer, 35–39.
[10] Emily Black, Manish Raghavan, and Solon Barocas. 2022. Model multiplicity: Opportunities, concerns, and solutions. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 850–863.
[11] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International conference on machine learning*. PMLR, 1613–1622.
[12] Léon Bottou. 2012. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*. Springer, 421–436.
[13] Simon Caton and Christian Haas. 2020. Fairness in machine learning: A survey. *arXiv preprint arXiv:2010.04053* (2020).
[14] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12, 3 (2011).
[15] Alexandra Chouldechova and Aaron Roth. 2018. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810* (2018).
[16] Kate Crawford. 2013. The hidden biases in big data. *Harvard business review* 1, 4 (2013).
[17] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems* 34 (2021).
[18] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. 2018. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*. PMLR, 1309–1318.
[19] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.
[20] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. 2019. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*. 329–338.
[21] Akihiko Fukuchi, Yoko Yabe, and Masashi Sode. 2020. FairTorch. https://github.com/wbawakate/fairtorch.
[22] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.
[23] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. 2018. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems* 31 (2018).
[24] Abhijit Ghatak and Abhijit Ghatak. 2019. Initialization of Network Parameters. *Deep Learning with R* (2019), 87–102.
[25] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).
[26] Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. 2020. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058* (2020).
[27] Matthew Jagielski, Om Thakkar, Florian Tramèr, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, et al. 2022. Measuring Forgetting of Memorized Training Examples. *arXiv preprint arXiv:2207.00099* (2022).
[28] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems* 33, 1 (2012), 1–33.
[29] Cherry Khosla and Baljit Singh Saini. 2020. Enhancing performance of deep learning models with different data augmentation techniques: A survey. In *2020 International Conference on Intelligent Engineering and Management (ICIEM)*. IEEE, 79–85.
[30] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.
[31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.
[32] Yucheng Lu, Wentao Guo, and Christopher M De Sa. 2022. GraB: Finding Provably Better Data Permutations than Random Reshuffling. *Advances in Neural Information Processing Systems* 35 (2022), 8969–8981.
[33] Frank J Massey Jr. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association* 46, 253 (1951), 68–78.
[34] R Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. 217–227.
[35] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.
[36] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. 2020. Random reshuffling: Simple analysis with vast improvements. *Advances in Neural Information Processing Systems* 33 (2020), 17309–17320.
[37] Amirkeivan Mohtashami, Sebastian Stich, and Martin Jaggi. 2022. Characterizing & Finding Good Data Orderings for Fast Convergence of Sequential Gradient Methods. *arXiv preprint arXiv:2202.01838* (2022).
[38] Arvind Narayanan. 2018. Translation tutorial: 21 fairness definitions and their politics. In *Proc. Conf. Fairness Accountability Transp., New York, USA*, Vol. 1170. 3.
[39] Lam M Nguyen, Quoc Tran-Dinh, Dzung T Phan, Phuong Ha Nguyen, and Marten Van Dijk. 2021. A unified convergence analysis for shuffling-type gradient methods. *The Journal of Machine Learning Research* 22, 1 (2021), 9397–9440.
[40] Hyeonwoo Noh, Tackgeun You, Jonghwan Mun, and Bohyung Han. 2017. Regularizing deep neural networks by noise: Its interpretation and optimization. *Advances in Neural Information Processing Systems* 30 (2017).
[41] Shangshu Qian, Viet Hung Pham, Thibaud Lutellier, Zeou Hu, Jungwon Kim, Lin Tan, Yaoliang Yu, Jiahao Chen, and Sameena Shah. 2021. Are my deep learning systems fair? An empirical study of fixed-seed training. *Advances in Neural Information Processing Systems* 34 (2021).
[42] Shashank Rajput, Kangwook Lee, and Dimitris Papailiopoulos. 2022. Permutation-Based SGD: Is Random Optimal?. In *International Conference on Learning Representations*.
[43] Charan Reddy, Deepak Sharma, Soroush Mehri, Adriana Romero-Soriano, Samira Shabanian, and Sina Honari. 2021. Benchmarking Bias Mitigation Algorithms in Representation Learning through Fairness Metrics. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.
[44] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. 2020. Fair-Batch: Batch Selection for Model Fairness. In *International Conference on Learning Representations*.
[45] Nripsuta Ani Saxena. 2019. Perceptions of fairness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 537–538.
[46] Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Raluca Turc, Jacob Eisenstein, et al. 2021. The MultiBERTs: BERT Reproductions for Robustness Analysis. In *International Conference on Learning Representations*.
[47] Vatsal Shah, Xiaoxia Wu, and Sujay Sanghavi. 2020. Choosing the sample with lowest loss makes sgd robust. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2120–2130.
[48] Ilia Shumailov, Zakhar Shumaylov, Dmitry Kazhdan, Yiren Zhao, Nicolas Papernot, Murat A Erdogdu, and Ross J Anderson. 2021. Manipulating sgd with data ordering attacks. *Advances in Neural Information Processing Systems* 34 (2021), 18021–18032.
[49] Ioana Baldini Soares, Dennis Wei, Karthikeyan Natesan Ramamurthy, Moninder Singh, and Mikhail Yurochkin. 2022. Your Fairness May Vary: Pretrained Language Model Fairness in Toxic Text Classification. In *Annual Meeting of the Association for Computational Linguistics*.
[50] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision* (2022), 1–40.
[51] Kushal Tirumala, Aram H Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization Without Overfitting: Analyzing the Training Dynamics of Large Language Models. *arXiv preprint arXiv:2205.10770* (2022).
[52] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2018. An Empirical Study of Example Forgetting during Deep Neural Network Learning. In *International Conference on Learning Representations*.

[53] Vladimir Vapnik. 1991. Principles of risk minimization for learning theory. *Advances in neural information processing systems* 4 (1991).
[54] Abhinav Venigalla and Linden Li. 2022. Mosaic LLMs (Part 2): GPT-3 quality for $< \$500k$. https://www.mosaicml.com/blog/gpt-3-quality-for-500k
[55] Stefan Wager, Sida Wang, and Percy S Liang. 2013. Dropout training as adaptive regularization. *Advances in neural information processing systems* 26 (2013).
[56] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2979–2989.

IMAGE CREDITS

- Blueprint by Berkah Icon from Noun Project
- Neural Network by Ian Rahmadi Kurniawan from Noun Project
- Data by shashank singh from Noun Project
- Data processing by Eko Purnomo from Noun Project
- Evaluate by Justin Blake from Noun Project