# Are My Deep Learning Systems Fair? An Empirical Study of Fixed-Seed Training

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Deep learning (DL) systems have been gaining popularity in critical tasks such as credit evaluation and crime prediction. Such systems demand fairness. Recent work shows that DL software implementations introduce variance: identical DL training runs (i.e., identical network, data, configuration, software, and hardware) with a fixed seed produce different models. Such variance could make DL models and networks violate fairness compliance laws, resulting in negative social impact. In this paper, we conduct the first empirical study to quantify the impact of software implementation on the fairness and its variance of DL systems. Our study of 22 mitigation techniques and five baselines reveals up to 12.6% fairness variance across identical training runs with identical seeds. In addition, most debiasing algorithms have a negative impact on the model such as reducing model accuracy, increasing fairness variance, or increasing accuracy variance. Our literature survey shows that while fairness is gaining popularity in artificial intelligence (AI) related conferences, only 34.4% of the papers use multiple identical training runs to evaluate their approach, raising concerns about their results' validity. We call for better fairness evaluation and fairness testing approaches to improve fairness and fairness variance of DL systems and improve DL research validity and reproducibility.

## 1 Introduction

DL systems, which consist of DL models and DL software (libraries such as TensorFlow) [51], are widely used and their fairness is paramount [32, 43, 69, 70]. Biases in DL systems have been identified as a major concern for civil rights by the U.S. government [17]. Yet, fairness testing for DL systems is still an open challenge.

Although the DL community is aware of the nondeterministism caused by using different random seeds (e.g., Question 3(c) in the NeurIPS checklist), such algorithmic factors are not the sole source of DL nondeterminism. Even **with the same seed**, e.g., the same initial weights, DL systems are nondeterministic, i.e., multiple training runs with the same network configuration, the same dataset, and the same seed, referred to as *fixed-seed identical training runs* (*FIT runs*), produce models with different accuracies [51]. Such nondeterminism is caused by **software implementation** (e.g., software parallelism and floating-point computation of DL software, details in Appendix E). Pham et al. [51] show that software implementation alone causes models' accuracies to vary by up to 2.9%, and most (83.8%) of the DL researchers and practitioners surveyed are unaware or unsure about the nondeterminism caused by software implementation. The survey participants are researchers and practitioners with DL experience from universities and AI players including Google, NVidia, Microsoft, and Facebook. The results suggest that even with FIT runs, one may still need to report error bars or use statistical tests.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Since the focus of the prior study [51] was on the variance of accuracy and training time only, it is unknown how DL software implementation affects the variance of *fairness*. Various debiasing algorithms [2, 32, 43, 70, 71] optimize DL models toward a specific fairness goal and it is unclear how software implementation affects these debiasing algorithms. In addition, unlike accuracy, which is a universally accepted metric, dozens of fairness metrics exist [19, 22, 31, 44, 47, 71, 80]. It is unclear if software implementation causes different levels of variance for different fairness metrics.

In this paper, **we study the impact of software implementation on the fairness variance in the training of DL models**. Our results show that models considered fair for one specific run might not be fair in another *fixed-seed identical training run*.

For example, when reproducing a previous work [32] proposing a new network and a debiasing algorithm, one of the FIT run generates a fair model (i.e., a model that does not discriminate based on gender) based on the threshold (20%) for the normalized disparate impact ($\overline{\text{DI}}$) metric [19, 47] ($\overline{\text{DI}}$ of 17.9%). A $\overline{\text{DI}}$ of 0% is the fairest, while a 100% $\overline{\text{DI}}$ is the least fair. However, a second FIT run produces an unfair model ($\overline{\text{DI}}$ of 20.4%, above the 20% threshold). This 20% threshold for $\overline{\text{DI}}$ has been used in **a U.S. legal case** to determine a system's fairness [7], where fairness variance could have changed the outcome. Overall, we found that fairness variance is large—up to 12.6% difference (27.3% versus 39.9%) across 16 FIT runs.

Variance of fairness is also problematic for evaluating debiasing algorithms. For example, when studying the interaction between the DL software implementation and the debiasing algorithms such as adversarial training with gradient projection [79], our experiments show that while this debiasing algorithm reduces the bias amplification to 6.1% in one FIT run (compared with a BA score of 7.8% for its baseline), the Mann-Whitney U-test with 16 FIT runs shows that *this technique increases the bias amplification instead of reducing it*: an average BA score of 8.7% compared to an average BA score of 7.4% for the baseline. This observed variance across 16 runs is solely caused by software implementation. Without fixed seed, a similar behavior is also observed by Agrawal et al. [1].

Finally, many mitigation techniques increase fairness variance or hurt accuracy. With Mann–Whitney U-test and Levene's test, our experiments show that a majority (67.9%) of the mitigation techniques reduce biases with a cost: they increase fairness variance, increase accuracy variance, or decrease model accuracy.

Since existing debiasing algorithms [32, 43] often do not measure the variance of biases, they could introduce risks. When such debiasing algorithms are applied to online learning, where DL models are trained continuously [37], *the increased variance from debiasing algorithms could cause the resulting models' fairness to miss compliance requirements or service-level agreements*. For example, Microsoft Tay [62] became a "racist chatbot" only after a few hours of deployment.

What's worse, even with fairness auditing before deployment, which is still *not a standard practice in industry* [25], a model can be fair on pre-deployment data, but unfair on real-world data, due to the gap between training/test data and real-world data. The fairness variance puts the trained models at a greater risk of being unfair on real-world data. Theoretically, while debiasing techniques on average improve fairness on the training set, they increase the variance of fairness on the test set [1]. In practice, such gap are the key reason that facial recognition systems [42, 54] have racial bias.

In addition to quantifying fairness variance, **we conduct a survey of research papers** from top artificial intelligence (AI) conferences of the recent years (2016–2020) to understand how researchers handle the fairness variance caused by DL software implementations. Only 34.4% of the 32 papers evaluating or addressing DL fairness issues use multiple training runs in their evaluation.

To the best of our knowledge, we are the first to study software implementations' impact on DL fairness variance. The contributions are:

- A variance study of DL fairness of 27 techniques (22 mitigation techniques and five baselines). The experiments are performed on four popular datasets (CelebA, MS-COCO, imSitu, and CIFAR-10S) with three DL networks (ResNet-18, ResNet-50, and NIFR [32]), measured by seven popular bias metrics (Section 3). **Finding 1**: Across 16 FIT runs, software implementation causes a maximum difference in bias metrics of up to 12.6% in DP metric, and up to 11.8% in $\overline{\text{DI}}$ metric. Yet, most (83.8%) are unaware or unsure about the DL nondeterminism caused by software implementation [51]. **Finding 2**: The effectiveness of the debiasing algorithms can be inaccurate with only one fixed-seed training run. **Finding 3**: Most of (15 out of 22) the

mitigation techniques tested increase at least one of the bias metrics. **Finding 4**: In a majority (67.9%, or 70) of 103 experiments, debiasing algorithms have a negative impact on the model.

- A literature survey on the fairness research papers of top machine learning (ML), (NeurIPS/NIPS, ICML, AAAI, FAccT), and computer vision (CVPR, ICCV, ECCV) conferences (Section 4).
- Implications and suggestions for DL fairness researchers and practitioners (Section 5), including one should use FIT runs (setup details in Section 2.5) with statistical tests (Appendix B.3) to evaluate debiasing algorithms, when nondeterminism from DL software is unavoidable.

# 2 Experimental method

We investigate 27 techniques of DL fairness on four popular datasets with three networks by reproducing four previous works [32, 43, 70, 71] that fit our scope.

## 2.1 Scope

We focus on the impact of DL software implementation on the variance of group fairness [16] in classification tasks, where two orthogonal labels are assigned to each instance in the dataset, a *task label* is used for training a classifier, and a *protected label* should not be discriminated against (e.g. gender). The set of instances with the same protected label forms a *protected group*.

Note that even the protected labels are not used in the classification task, the trained models could still be biased because the protected labels can be leaked into the prediction by proxy variables. Such leaking has been shown by many papers [2, 43, 70, 80].

## 2.2 Bias metrics

Various metrics exist to measure biases of ML algorithms. Yet, it is still an open problem to select these metrics as an optimization goal for real-world problems [34]. Therefore, we choose three fairness conditions and seven bias metrics derived from them. Fairness conditions include statistical parity [16] (SP), predictive equality [12] (PE), and equality of opportunity [22] (EOp). Seven bias metrics include demographic parity [10] (DP), normalized disparate impact [10] ($\overline{\text{DI}}$), statistical parity subgroup fairness [31] (SPSF), false positive subgroup fairness [31] (FPSF), equalized odds false positive [14] (EOFP), equalized odds true positive [22] (EOTP), and bias amplification [71, 80] (BA). For all bias metrics, 0 indicates the fairness condition is satisfied, i.e., a completely fair model, and 1 indicates a completely unfair model. Details of these metrics are in Appendix A.

**Per-class and overall bias:** Bias calculated on each class is called *per-class bias* and the bias for each model is the *overall bias*. For example, there are ten per-class biases in models trained on CIFAR-10 (one per class). The overall bias of the model is the average of all per-class biases.

**Multi-(task)label Classification:** We treat each task label as an independent binary classification, and the overall bias value of the model is the average across all labels.

## 2.3 Reproduced studies

We conduct a literature survey of published papers from the recent five years (2016–2020) of the top ML (NeurIPS/NIPS, ICML, and AAAI), and computer vision (CVPR, ICCV, and ECCV) conferences (Section 4). Out of the 32 papers relevant to the fairness of DL systems, only four [32, 43, 70, 71] have a working replication package so that we can reproduce their results. These papers study the fairness of DL systems for the five tasks described in Table 1 (expanded details in Appendix B.1). Overall, we replicate 27 different techniques from four state-of-the-art papers on five classification tasks. In the next section, we introduce each technique and detail our experiments.

## 2.4 Techniques

Table 1 lists the 27 techniques that we reproduce for our study. Column *Task* lists the tasks on which these techniques are applied. Column *Technique* lists the technique abbreviations we used. Column *Debiasing Algorithm* lists the debiasing algorithms used for each technique. Column *Network* lists

Table 1: Evaluated mitigation techniques and baselines

| Task | Technique | Debiasing Algorithm | Network | Tgt |
|---|---|---|---|---|
| Multi-class classification in CIFAR-10S [71] (1) | S-Base | Baseline | ResNet-18 | BA |
| | S-RS | Data Rebalancing | | |
| | S-UC | Uniform confusion loss | | |
| | S-GR | Gradient reversal | | |
| | S-DD1 | Sum of probabilities | | |
| | S-DD2 | Max of prob. w/ prior shift | | |
| | S-DD3 | Sum of prob. w/ prior shift | | |
| | S-DD4 | RBA debias | | |
| | S-DI1 | Domain independent, conditional | | |
| | S-DI2 | Domain independent, sum | | |
| Multi-label classification in MS-COCO [70] (2) | C-Base | Baseline | ResNet-50 | DP (BA*) |
| | C-R1 | Data rebalancing, M/F = 1 | | |
| | C-R2 | Data rebalancing, M/F < 2 | | |
| | C-R3 | Data rebalancing, M/F < 3 | | |
| | C-A4 | Adv training at 4th conv block | | |
| | C-A5 | Adv training at 5th conv block | | |
| Multi-class classification in imSitu [70] (3) | I-Base | Baseline | ResNet-50 | DP (BA*) |
| | I-R1 | Data rebalancing, M/F = 1 | | |
| | I-R2 | Data rebalancing, M/F < 2 | | |
| | I-R3 | Data rebalancing, M/F < 3 | | |
| | I-A4 | Adv training at 4th conv block | | |
| | I-A5 | Adv training at 5th conv block | | |
| Binary classification in CelebA [43] (4) | A-Base | Baseline | ResNet-18 | EO |
| | A-L2 | L2 penalty | | |
| | A-ALM | FairALM | | |
| Binary classification in CelebA [32] (5) | N-Base | Baseline | NIFR | DP |
| | N-Flow | Conditional Flow | | |

the DL network used to train the models. Finally, column *Tgt* indicates which bias metric the corresponding debiasing algorithm is designed to reduce.

Five techniques (S-Base, C-Base, I-Base, A-Base, and N-Base) are baselines (i.e., do not use any debiasing algorithm), and the other 22 use debiasing algorithms. Techniques starting with "S" are from [71]; techniques starting with "C" and "I" are from [70] trained for tasks (2) and (3) respectively; techniques starting with "A" are from [43]; and techniques starting with "N" are from [32].

Apart from five baseline techniques, the remaining algorithms can be grouped into three categories including data rebalancing, fairness through blindness, and fairness through awareness. Details can be found in Appendix B.2.

**Optimization target for tasks (2) and (3)**: The study [70] uses "bias amplification" as their optimization goal. However, literature does not use this term consistently. The original study definition (noted as BA*) is different from the one proposed by Zhao et al. [80] (noted as BA) and BA* is more analog to the DP metric. Therefore, we denote this optimization target as *DP (BA*)*.

### 2.5 Variance experiments setup

Following the prior work [51], we use *FIT runs* to measure the variance of trained model fairness. For each technique, all the training runs are executed with the same training data (also the original training/test split), hyper-parameters, and optimizers. With the fixed seed, all training runs also have the same order of data and the same initial weights. We perform 16 FIT runs with the same random seed for each technique, and then evaluate the fairness of the trained models using seven bias metrics.

Since there is no theoretical bound on the variance caused by DL software implementations, and each set of 16 FIT runs can only be viewed as a sample of the real distribution, we use statistical

Table 2: Maximum difference (MaxDiff) and standard deviation (STDEV) on the seven bias metrics for S-GR, A-L2, C-Base, and C-A5. All numbers are in percentage points.

| **Metric** | | DP | $\overline{\text{DI}}$ | SPSF | FPSF | EOFP | EOTP | BA | DP | $\overline{\text{DI}}$ | SPSF | FPSF | EOFP | EOTP | BA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Technique** | | **S-GR** | | | | | | | **A-L2** | | | | | | |
| **Overall** (%) | **MaxDiff** | 1.7 | **11.8** | 0.9 | 0.5 | 1.2 | 7.2 | 4.2 | **12.6** | 8.0 | 6.0 | 1.6 | 7.2 | 7.2 | 11.6 |
| | **STDEV** | 0.5 | 3.5 | 0.3 | 0.2 | 0.4 | 2.0 | 1.3 | 2.9 | 2.4 | 1.4 | 0.5 | 2.1 | 2.1 | 3.1 |
| **Per-class** (%) | **MaxDiff** | 7.3 | 50.1 | 3.6 | 2.5 | 5.7 | 34.6 | 19.3 | 12.6 | 31.1 | 6.0 | 3.5 | 14.3 | 14.3 | 15.9 |
| | **STDEV** | 1.9 | 12.7 | 1.0 | 0.7 | 1.4 | 8.1 | 5.0 | 2.9 | 8.8 | 1.4 | 1.2 | 4.8 | 4.8 | 4.2 |
| **Technique** | | **C-Base** | | | | | | | **C-A5** | | | | | | |
| **Overall** (%) | **MaxDiff** | <0.1 | 2.6 | <0.1 | <0.1 | <0.1 | 1.9 | 1.4 | 0.1 | 2.4 | <0.1 | <0.1 | <0.1 | 2.6 | 3.1 |
| | **STDEV** | <0.1 | 0.8 | <0.1 | <0.1 | <0.1 | 0.6 | 0.4 | <0.1 | 0.6 | <0.1 | <0.1 | <0.1 | 0.6 | 0.8 |
| **Per-class** (%) | **MaxDiff** | 1.6 | **100.0** | 0.8 | 0.5 | 1.1 | 33.3 | 33.3 | 3.7 | **100.0** | 1.8 | 1.0 | 2.1 | 43.9 | 65.4 |
| | **STDEV** | 0.5 | 38.9 | 0.3 | 0.2 | 0.4 | 11.7 | 8.1 | 0.9 | 28.2 | 0.4 | 0.2 | 0.5 | 10.8 | 21.8 |

tests, including Mann-Whitney U-test, Cohen's $d$ value, and Levene's test to ensure the statistical significance of our findings. Details of these statistical tests can be found in Appendix B.3.

# 3 Findings

We evaluate the fairness variance of 27 techniques (22 mitigation and five baseline) using seven bias metrics. These are 189 experiments (technique-metric pairs). To evaluate the impact of DL software implementation on different bias metrics, we perform 16 FIT runs for each of the 27 techniques, which produce 432 models. Each experiment generates one **bias value** per run (i.e., 16 bias values per experiment), which is a total of 3,024 bias values. The training time of the 189 experiments is 7,117 hours (**about 9.8 months**). Details about the software and hardware environment is in Appendix B.4.

To measure the difference across multiple FIT runs, we use two metrics: (1) the difference in percentage point as *MaxDiff* (maximum difference) and (2) the standard deviation in percentage point as *STDEV*. *MaxDiff* is the range of the bias values from a set of 16 FIT runs. For example, the MaxDiff of 16 runs, whose bias values are between 20.0%–30.0% inclusive, is 10.0%.

## 3.1 RQ1: How much variance of fairness does DL software implementation introduce?

Table 2 shows the fairness variance introduced by software implementation. Row *Technique* lists the techniques we focus on for this RQ. Row *Metric* shows the bias metrics used for evaluation. Row *Overall* shows the bias for the overall technique. Row *Per-class* shows the results for the class within each technique that has the most variance. The full results of 189 experiments are in the supplementary material.

Table 2 shows that software implementation impacts all bias metrics. For example, we observed a maximum difference in DP with technique A-L2 of 12.6% (27.3%–39.9%), i.e., the most unfair model has 12.6% more gender bias measured by DP than the most fair model. Also, in technique S-GR, we observed a maximum difference in $\overline{\text{DI}}$ of 11.8% (21.3%–33.1%). Software implementation solely causes such variance, and thus cannot be eliminated by fixing the random seeds.

**Per-class variance of fairness and single class fairness failure** could be problematic if a class is privileged. For example, in a DL model for loan application, the approval is the privileged class. A difference of 100% for $\overline{\text{DI}}$ means that in one run, the model is completely fair for all races (protected groups) to get approval. In another run, none of the applicants of a particular race gets approved, which is 100% bias against this minority group, even if the overall fairness variance is small.

**Finding 1**: Across 16 FIT runs, software implementation causes a maximum difference in bias of up to 12.6% in DP metric, and up to 11.8% in $\overline{\text{DI}}$ metric. A complete single-class fairness failure is possible, where one run produces a model with 0% bias while another with 100% bias.

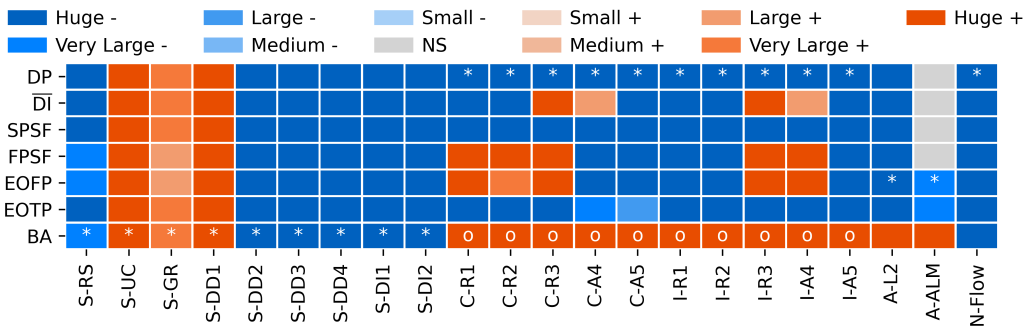


Figure 1: Effect size of *bias changes* between before and after mitigation application. Blue is a bias decrease (good), orange an increase (bad), and grey not statistically significant (NS).

Additional results of this RQ can be found in Appendix D.1. We discuss the implications of these results for researchers and practitioners in Section 5.

### 3.2 RQ2: How does software implementation affect debiasing algorithms?

To quantify the impact of *software implementation* on debiasing algorithms, we use Mann-Whitney U-test and Levene's test to check whether the *average bias value* and the *variance of bias value* after debiasing is statistically different from those of its baseline repectively. We normalize each technique's bias values before Levene's test to eliminate the impact of the average value [4, 18, 55, 65].

For 22.9% (34 of 154) of the experiments (the comparison of 22 mitigation techniques with their baseline techniques evaluated using seven bias metrics), the mitigation techniques' bias values overlap with those of the baselines. For example, the range of EOTP bias values for the A-ALM debiasing is 19.6%–24.3%, while its baseline's (A-Base) range is 22.6%–24.7%.

For 9.1% (14 of 154) of the experiments, the mitigation technique's average bias metric is higher than the baseline, yet a single run's bias value is lower than a single run of the baseline. Thus, using a single run, one may incorrectly conclude that the mitigation technique is fairer, but on average, the mitigation technique is less fair. For example, *S-Base* (7.4% BA) is fairer than *S-GR* (8.7% BA) on average of 16 FIT runs ($p = 0.0011$ in U-test). However, one may obtain a fairer model with S-GR from one training run (e.g., BA of 6.1% for S-GR vs. 7.8% for S-Base). Therefore, it is necessary that DL fairness researchers perform multiple runs and check statistical tests to ensure the effectiveness and validity of the debiasing algorithms. Our Levene's test results also confirm that the variance of bias values of *S-GR* is statistically higher than that of *S-Base* ($p = 0.0005$).

**Finding 2**: The effectiveness of the debiasing algorithms can be inaccurate with only one FIT run.

Since there is no consensus on which bias metric to use [12, 34], and some bias constraints cannot be satisfied simultaneously [35], we study if and how a debiasing algorithm reduces biases measured by one bias metric but increases biases measured by a different metric.

For the 154 experiments above, we conduct a series of Mann-Whitney U-tests on the bias values obtained. For each test that exhibits a statistically significant difference, we compute effect size using Cohen's $d$ value between the two sets of bias values (baseline and mitigation) involved. The $d$ values tell us whether each mitigation technique has a meaningful effect or not. Raw values before and after mitigation are available in the supplementary material.

Figure 1 shows the effect sizes of bias value between 22 mitigation techniques and their corresponding baselines. The x-axis and y-axis labels are the abbreviation for the techniques and bias metrics. If a mitigation technique statistically reduces the bias value on a bias metric (with respect to its baseline), the corresponding block is blue. Oppositely, the block is orange. The block is grey for mitigation techniques that do not induce a statistically significant difference from its baseline. The different shades of color show the standard grouping and interpretation of the effect size, such as "Huge" and "Very large". Figure 1's legend shows the color details. Blocks with a "*" are the bias metric that the mitigation technique is designed to reduce in the original research (e.g., BA for S-DI2 and DP for N-Flow). An "o" is a reminder that the optimization target of the mitigation techniques that begin with "C" and "I" is better captured by the DP metric (BA* in the original research, see Section 2.4).

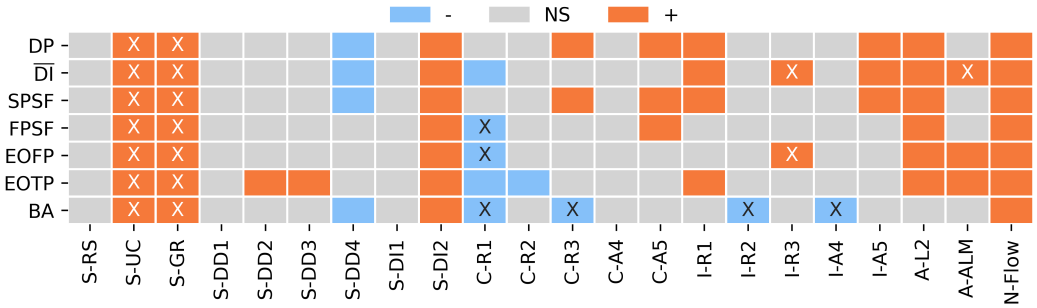


Figure 2: Cost of bias mitigation techniques on the *variance* of bias metrics after normalization (Levene's tests). Blue denotes a decrease in variance (good), orange an increase in variance (bad), and grey not statistically significant (NS).

Figure 1 shows that most (96 out of 103) mitigation techniques that induce a statistically significant improvement on the bias value have a huge effect (deep blue block). However, the vast majority (68.2%, 15 of 22) of mitigation techniques fail to reduce biases measured by all seven bias metrics. For example, A-L2 and A-ALM aim to improve bias metric EOFP. Although they succeeded, they also increase biases for BA.

This experiment shows that for most of the mitigation techniques, bias metrics may not be compatible with each other. Practitioners applying mitigation techniques must understand the techniques' optimization goal and use statistical tests.

**Finding 3**: Most of (15 out of 22) the mitigation techniques tested increase at least one of the bias metrics: all 15 of them fail to improve the newly introduced bias amplification metric.

### 3.3 RQ3: How does software implementation affect the cost of mitigation techniques?

The most known cost of mitigation techniques is the trade-off between fairness and model accuracy [15, 72]. However, with the software implementation taken into consideration, mitigation techniques could introduce new costs including more variance of model accuracy and fairness.

There are three dimensions to the cost of mitigation techniques: lower accuracy, higher accuracy variance, and higher fairness variance. We excluded the techniques that increase bias for all metrics compared to its baseline. For accuracy, we use U-test to check if the reduction is statistically significant. For the variance of both model accuracy and model bias, we use Levene's test.

Figure 2 shows the cost of mitigation techniques on fairness variance (Levene's test). Labels on x-axis and y-axis are the same as those in Figure 1. Blue and orange blocks respectively indicate lower and higher variance than the baseline. Grey blocks mean no statistically significant change in the variance of bias values, and "X" indicates mitigation technique fails to improve the bias metric tested. For example, the orange [N-Flow, DP] block shows that N-Flow generates models whose variance of DP bias values is higher than that of the baseline, and the comparison results are statistically significant.

To study the relationship among the three types of cost, we further examine the 154 experiments on mitigation techniques. Most of the experiments (103) reduce biases. The majority of them (70 out of 103) achieve bias reduction with a cost in one or more of the three dimensions. Eight experiments come with all three costs: lower accuracy, higher accuracy variance, and higher fairness variance, and are from A-L2 and A-ALM. However, there is no correlation between accuracy variance and bias variance. Interestingly, mitigation technique S-DD4 (RBA [80] debiasing) successfully reduces all of the seven bias metrics without sacrificing model accuracy or introducing more variance.

DL researchers and practitioners might overlook the cost of fairness variance when optimizing a model toward a particular fairness goal if they are unaware of the variance caused by DL software [51].

**Finding 4**: Out of 154 experiments with debiasing algorithms, 103 experiments show the applied algorithms reduce the bias values. In a majority (67.9%, or 70) of these 103 experiments, the debiasing algorithms have a negative impact on the model (36 increase the variance of bias value, 54 decrease the accuracy, and 15 increase the variance of accuracy, not exclusively).

# 4 Literature survey

We conduct a literature survey from the top ML conferences to evaluate the awareness of fairness variance of DL systems.

**Paper Selection Criteria:** We extract papers from the last five years (2016-2020) of the top ML (NeurIPS/NIPS, ICML, AAAI), and computer vision (CVPR, ICCV, ECCV) conferences. We also include FAccT and its predecessor FAT/ML workshop, an interdisciplinary venue of ML and fairness.

We only consider papers that contain fairness-related keywords (fair, bias, discriminate, aware, balance, disparity, opportunity, audit, gender, stereotype, race, amplification, ethic) in their titles using the Snowball Stemmer [53]. Then, we read the abstract and the paper and keep relevant papers involving fairness of a classification task. Out of 24,186 papers, we extracted 274 papers studying fairness, including 32 dedicated to DL systems' fairness.

**Paper Survey Results:** While still taking only a small portion of accepted papers, the number of papers related to fairness has been increasing over the last five years.Researchers need to be aware of DL models' fairness variance caused by DL software implementations.

**Finding 5**: DL fairness is gaining more attention in the research community since 2017. The number of papers related to DL fairness in top ML and computer vision conferences has increased 750% since 2017 (two papers in 2017, 17 papers in 2020).

To understand whether researchers are aware of the fairness variance of DL training, we manually check whether the papers on the fairness of DL systems use multiple identical runs in their evaluation.

**Finding 6**: Of 32 papers relevant to the fairness of DL systems, only 34.4% (11 out of 32) of them use multiple identical runs to evaluate their proposed approaches.

# 5 Implications and suggestions

**Research validity:** When comparing the results of a proposed method (e.g., a new debiasing algorithm or a new DL network), it is necessary to run statistical tests to confirm any improvements.

**Testing for fairness in online learning:** When testing an online-learning software system for fairness (where models are continuously updated), such as identity-inference-based advertisement targeting systems [6, 50, 73], multiple identical runs are necessary, and statistical test results should be reported to measure the variance of bias values caused by the DL software implementation.

**Research reproducibility:** Given the increasing number of papers on fairness in AI conferences [5, 21, 33, 46, 57, 77], reproducibility becomes increasingly important [3, 56]. Following prior efforts [28, 40, 52] that improve the reproducibility of DL methods, researchers may want to develop such guidance for fairness, e.g., considering multiple bias metrics, as variance levels are different for different bias metrics. Proper documentation of the software and hardware development environment and configuration parameters would also facilitate reproduction of research results. As shown in Section 3.1, software implementation alone causes software fairness to vary by up to 12.6%, which is much higher than the 2.9% accuracy variance discovered by previous work [51].

**Balancing hidden costs of debiasing:** Since variance is a hidden cost for debiasing algorithms and could harm the fairness performance of online learning algorithms (Section 3.3), there is usually a trade-off between model accuracy, fairness, and variance. It is important that DL researchers and practitioners understand debiasing algorithms' costs, especially variance, and strike a balance among them.

**Deterministic DL software for model tuning and debugging:** Building deterministic DL libraries could be a important research topic to improve model tuning and the debugging of DL systems. For now, one could use the deterministic mode (Appendix E) when it is provided by DL libraries.

**Maximizing fairness through multiple runs:** To take advantage of software variance, practitioners could train multiple times to select the fairest model. In this case, proper testing of the DL model is

recommended to ensure the satisfaction of regulatory requirements, considering the pre-deployment and real-world data gap.

**Selection of bias metrics:** In most cases, a debiasing algorithm does not reduce bias on all targets (Section 3.2), and multiple fairness constraints could contradict each other [34, 35]. Therefore, developers and testers for DL systems should carefully select fairness requirements. Furthermore, optimizing on one bias metric may not be enough as it might make the model less fair on other metrics. DL researchers should consider the task and use the appropriate bias metric. In the strictest sense, one may need to use multiple bias metrics to ensure fairness across multiple dimensions.

# 6 Related work

**Anecdotal Evidence in Fairness Variance:** Agrawal et al. [1] apply debiasing on the random forest classifiers and test for fairness variance with different random seeds and data splits for cross-validation. They find that debiasing algorithms could increase the fairness variance introduced by algorithmic nondeterminism. Different from their work, we focus on DL models and variance caused by DL software implementations, which cannot be removed by fixing random seeds.

**Fairness Studies of ML Systems:** The fairness of ML and DL systems have gained attention and several studies pointed out fairness issues in datasets (e.g., CelebA [58], GitHub [26], and StackOverflow [45]) and major ML systems such as language models [9, 66, 68], face recognition [64], and sentiment analysis [63]. Such studies demonstrate that ML systems can be unfair and highlight the importance of fairness. Yet to the best of our knowledge, we are the first to study the impact of DL software implementation on fairness.

**Bias Metrics:** Many metrics [5, 10, 12, 14, 16, 20, 22, 29, 31, 49, 59, 74, 75] have been proposed to measure biases of ML systems. Most of them can be applied to DL systems too. We evaluate fairness using seven of the most popular metrics and show that they are all impacted by DL implementation variance. None of the existing metrics considers such variance in their definition.

**Debiasing Algorithms:** To mitigate biases of DL systems, many debiasing algorithms or bias resistant networks have been proposed and evaluated [8, 10, 11, 30, 32, 38, 43, 48, 60, 70, 71, 78]. These techniques are important to improve the fairness of DL systems but their evaluation can be affected by fairness variance introduced by DL implementation. Our study shows that the impact of some debiasing algorithms is not consistent during different identical runs.

# 7 Conclusion and discussion

This work studies the fairness variance of DL systems caused by software implementations. We examine five tasks on three networks and four datasets using 22 mitigation techniques. We find up to 12.6% difference in bias values across FIT runs and 67.9% of the experiments reduce model biases with cost in fairness variance, accuracy, or accuracy variance. Our paper survey shows that only 34.4% of the papers use multiple identical runs to evaluate the proposed approaches. We call for the awareness of the fairness variance of DL systems caused by software implementations.

**Limitation**: It is known that gender bias exists in language embeddings [80]. We only study image classification tasks because they are better fit for our scope of group fairness and bias metrics on image classification tasks are better defined. Also, we only empirically obtain the upper bound of fairness variance and did not include any theoretical proof. We leave both limitations as future work.

**Negative Social Impact**: With FIT runs to estimate the fairness variance of neural networks, our proposed approach could increase the power consumption and $CO_2$ emissions of DL model training. However, such additional evaluation is required for experimental validity and fairness compliance. In other words, the negative social impact of not performing multiple training runs include disproportionally hurting certain races or genders. Our paper raises awareness of this issue and calls for new solutions to strike a balance. We are working on theoretically quantifying the upper bound of fairness variance and cheaper ways to estimate/bound variance.

# References


[1] Ashrya Agrawal, Florian Pfisterer, Bernd Bischl, Jiahao Chen, Srijan Sood, Sameena Shah, Francois Buet-Golfouse, Bilal A Mateen, and Sebastian J Vollmer. 2020. Debiasing classifiers: is reality at variance with expectation? (2020), 11 pages. arXiv:2011.02407

[2] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. 2019. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *Computer Vision (ECCV '18 Workshops)*. Springer, Cham, 556–572.

[3] Andrew L Beam, Arjun K Manrai, and Marzyeh Ghassemi. 2020. Challenges to the reproducibility of machine learning models in health care. *JAMA* 323, 4 (2020), 305–306.

[4] Arthur G Bedeian and Kevin W Mossholder. 2000. On the use of the coefficient of variation as a measure of diversity. *Organ. Res. Methods* 3, 3 (2000), 285–297.

[5] Omer Ben-Porat, Fedor Sandomirskiy, and Moshe Tennenholtz. 2019. Protecting the protected group: Circumventing harmful fairness. *arXiv preprint arXiv:1905.10546* (2019).

[6] Sebastian Benthall and Bruce D Haynes. 2019. Racial categories in machine learning. In *Conference on Fairness, Accountability, and Transparency (FAT* '19)*. ACM, New York, 289–298.

[7] Dan Biddle. 2006. *Adverse impact and test validation: A practitioner's guide to valid and defensible employment testing* (2 ed.). Routledge, London and New York.

[8] Sumon Biswas and Hridesh Rajan. 2020. Do the machine learning models on a crowd sourced platform exhibit bias? An empirical study on model fairness. In *28th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020)*. ACM, New York, 642–653.

[9] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems (NIPS '16, Vol. 29)*. Curran Associates, Red Hook, NY, 4349–4357.

[10] Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Min. Knowl. Disc.* 21, 2 (2010), 277–292.

[11] Joymallya Chakraborty, Suvodeep Majumder, Zhe Yu, and Tim Menzies. 2020. Fairway: a way to build fair ML software. In *28th Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20)*. ACM, New York, 654–665.

[12] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data* 5, 2 (2017), 153–163.

[13] COCODataset. [n.d.]. *Common Objects in Context Terms of Use*. `https://cocodataset.org/#termsofuse` Last accessed 2021-05-25.

[14] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. 2020. Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems* (2020), 7 pages.

[15] Sanghamitra Dutta, Dennis Wei, Hazar Yueksel, Pin-Yu Chen, Sijia Liu, and Kush Varshney. 2020. Is there a trade-off between fairness and accuracy? A perspective using mismatched hypothesis testing. In *International Conference on Machine Learning (ICML '20, Vol. 119)*. PMLR, 2803–2813.

[16] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *3rd Innovations in Theoretical Computer Science Conference (ITCS '12)*. ACM, New York, 214–226.

[17] Executive Office of the President. 2016. Big data: A report on algorithmic systems, opportunity, and civil rights.

[18] DS Faber and H Korn. 1991. Applicability of the coefficient of variation method for analyzing synaptic plasticity. *Biophys. J.* 60, 5 (1991), 1288–1294.

[19] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *21st SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. ACM, New York, 259–268.

[20] Stephen Gillen, Christopher Jung, Michael Kearns, and Aaron Roth. 2018. Online learning with an unknown fairness metric. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2605–2614.

[21] Naman Goel, Alfonso Amayuelas, Amit Deshpande, and Amit Sharma. 2020. The Importance of Modeling Data Missingness in Algorithmic Fairness: A Causal Perspective. *arXiv preprint arXiv:2012.11448* (2020).

[22] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NIPS'16, Vol. 29)*. Curran Associates, Red Hook, NY, 3315–3323.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR '16)*. IEEE, New York, 770–778.

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

[25] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. 2019. Improving fairness in machine learning systems: What do industry practitioners need?. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–16.

[26] Nasif Imtiaz, Justin Middleton, Joymallya Chakraborty, Neill Robson, Gina Bai, and Emerson Murphy-Hill. 2019. Investigating the effects of gender bias on GitHub. In *41st International Conference on Software Engineering (ICSE '19)*. IEEE/ACM, New York, 700–711.

[27] Jörn-Henrik Jacobsen, Arnold W M Smeulders, and Edouard Oyallon. 2018. *i*-RevNet: Deep invertible networks. In *6th International Conference on Learning Representations (ICLR '18)*. 11 pages.

[28] Christoph Jansen, Jonas Annuscheit, Bruno Schilling, Klaus Strohmenger, Michael Witt, Felix Bartusch, Christian Herta, Peter Hufnagl, and Dagmar Krefting. 2020. Curious Containers: A framework for computational reproducibility in life sciences with support for Deep Learning applications. *Future Gener. Comp. Sy.* 112 (2020), 209–227.

[29] Nathan Kallus and Angela Zhou. 2019. The fairness of risk scores beyond classification: Bipartite ranking and the xauc metric. *arXiv preprint arXiv:1902.05826* (2019).

[30] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD '12)*. Springer, Berlin Heidelberg, 35–50.

[31] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *35th International Conference on Machine Learning (ICML '18, Vol. 80)*. PMLR, Stockholm, 2564–2572.

[32] Thomas Kehrenberg, Myles Bartlett, Oliver Thomas, and Novi Quadrianto. 2020. Null-sampling for Interpretable and Fair Representations. In *Computer Vision (ECCV '20)*. Springer, Cham, 565–580.

[33] Hyemi Kim, Seungjae Shin, JoonHo Jang, Kyungwoo Song, Weonyoung Joo, Wanmo Kang, and Il-Chul Moon. 2020. Counterfactual Fairness with Disentangled Causal Effect Variational Autoencoder. *arXiv preprint arXiv:2011.11878* (2020).

[34] Joon Sik Kim, Jiahao Chen, and Ameet Talwalkar. 2020. FACT: A diagnostic for group fairness trade-offs. In *37th International Conference on Machine Learning (PMLR, Vol. 119)*. PMLR, 5264–5274.

[35] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS '17) (LIPIcs, Vol. 67)*. Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 43:1–23.

[36] Alex Krizhevsky. 2009. *Learning multiple layers of features from tiny images*. Master's thesis. University of Toronto.

[37] Gopiram Roshan Lal, Sahin Cem Geyik, and Krishnaram Kenthapadi. 2020. Fairness-Aware Online Personalization. (2020), 10 pages. arXiv:2007.15270

[38] Peizhao Li, Han Zhao, and Hongfu Liu. 2020. Deep fair clustering for visual learning. In *Conference on Computer Vision and Pattern Recognition (CVPR '20)*. IEEE/CVF, New York, 9070–9079.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Computer Vision (ECCV '14)*. Springer, Cham, 740–755.

[40] Chao Liu, Cuiyun Gao, Xin Xia, David Lo, John Grundy, and Xiaohu Yang. 2020. On the Replicability and Reproducibility of Deep Learning in Software Engineering. (2020). arXiv:2006.14244

[41] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *International Conference on Computer Vision (ICCV '15)*. IEEE, New York, 3730–3738.

[42] Steve Lohr. 2018. Facial recognition is accurate, if you're a white guy. *New York Times* 9 (2018), 8.

[43] Vishnu Suresh Lokhande, Aditya Kumar Akash, Sathya N Ravi, and Vikas Singh. 2020. FairALM: Augmented Lagrangian Method for Training Fair Models with Little Regret. In *Computer Vision (ECCV '20)*. Springer, Cham, 365–381.

[44] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 2018. Learning Adversarially Fair and Transferable Representations. In *35th International Conference on Machine Learning (ICML '18, Vol. 80)*. PMLR, Stockholm, 3384–3393.

[45] Anna May, Johannes Wachs, and Anikó Hannák. 2019. Gender differences in participation and reward on Stack Overflow. *Empir. Softw. Eng.* 24, 4 (2019), 1997–2019.

[46] Ninareh Mehrabi, Muhammad Naveed, Fred Morstatter, and Aram Galstyan. 2020. Exacerbating Algorithmic Bias through Fairness Attacks. *arXiv preprint arXiv:2012.08723* (2020).

[47] Aditya Krishna Menon and Robert C Williamson. 2018. The cost of fairness in binary classification. In *1st Conference on Fairness, Accountability and Transparency (FAT* '18, Vol. 81)*. PMLR, New York, 107–118.

[48] Shaobo Min, Hantao Yao, Hongtao Xie, Chaoqun Wang, Zheng-Jun Zha, and Yongdong Zhang. 2020. Domain-aware visual bias eliminating for generalized zero-shot learning. In *Conference on Computer Vision and Pattern Recognition (CVPR '20)*. IEEE/CVF, New York, 12664–12673.

[49] Debarghya Mukherjee, Mikhail Yurochkin, Moulinath Banerjee, and Yuekai Sun. 2020. Two simple ways to learn individual fairness metrics from data. In *International Conference on Machine Learning*. PMLR, 7097–7107.

[50] Claudia Perlich, Brian Dalessandro, Troy Raeder, Ori Stitelman, and Foster Provost. 2014. Machine learning for targeted display advertising: Transfer learning in action. *Mach. Learn.* 95, 1 (2014), 103–127.

[51] Hung Viet Pham, Shangshu Qian, Jiannan Wang, Thibaud Lutellier, Jonathan Rosenthal, Lin Tan, Yaoliang Yu, and Nachiappan Nagappan. 2020. Problems and Opportunities in Training Deep Learning Software Systems: An Analysis of Variance. In *35th International Conference on Automated Software Engineering (ASE '20)*. IEEE/ACM, New York, 771–783.

[52] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. 2020. Improving reproducibility in machine learning research (a report from the NeurIPS 2019 Reproducibility Program). (2020), 22 pages. arXiv:2003.12206

[53] Martin F Porter. 2001. Snowball: A language for stemming algorithms.

[54] Inioluwa Deborah Raji, Timnit Gebru, Margaret Mitchell, Joy Buolamwini, Joonseok Lee, and Emily Denton. 2020. Saving face: Investigating the ethical concerns of facial recognition auditing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 145–151.

[55] George F Reed, Freyja Lynn, and Bruce D Meade. 2002. Use of coefficient of variation in assessing variability of quantitative assays. *Clin. Diagn. Lab. Immun.* 9, 6 (2002), 1235–1239.

[56] Félix Renard, Soulaimane Guedria, Noel De Palma, and Nicolas Vuillerme. 2020. Variability and reproducibility in deep learning for medical image segmentation. *Sci. Rep.* 10, 1 (2020), 1–16.

[57] Ashkan Rezaei, Anqi Liu, Omid Memarrast, and Brian Ziebart. 2020. Robust Fairness under Covariate Shift. *arXiv preprint arXiv:2010.05166* (2020).

[58] Hee Jung Ryu, Hartwig Adam, and Margaret Mitchell. 2017. InclusiveFaceNet: Improving face attribute detection with race and gender diversity. In *Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML '18)*. 6 pages. arXiv:1712.00193

[59] Sivan Sabato and Elad Yom-Tov. 2020. Bounding the fairness and accuracy of classifiers from population statistics. In *International Conference on Machine Learning*. PMLR, 8316–8325.

[60] Mhd Hasan Sarhan, Nassir Navab, Abouzar Eslami, and Shadi Albarqouni. 2020. Fairness by learning orthogonal disentangled representations. In *Computer Vision (ECCV '20)*. Springer, Cham, 746–761.

[61] Shlomo S Sawilowsky. 2009. New effect size rules of thumb. *J. Mod. Appl. Stat. Methods* 8, 2 (2009), 597–599.

[62] Oscar Schwartz. 2019. In 2016 Microsoft's Racist Chatbot Revealed the Dangers of Online Conversation. *IEEE Spectrum* 11 (2019).

[63] Judy Hanwen Shen, Lauren Fratamico, Iyad Rahwan, and Alexander M Rush. 2018. *Darling or babygirl?* Investigating stylistic bias in sentiment analysis. In *Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML '18)*.

[64] Richa Singh, Akshay Agarwal, Maneet Singh, Shruti Nagpal, and Mayank Vatsa. 2020. On the robustness of face recognition algorithms against attacks and bias. In *Conference on Artificial Intelligence (AAAI '20, Vol. 34)*. AAAI, Palo Alto, CA, 13583–13589.

[65] Jesper B Sørensen. 2002. The use and misuse of the coefficient of variation in organizational demography research. *Sociol. Method Res.* 30, 4 (2002), 475–491.

[66] Yi Chern Tan and L Elisa Celis. 2019. Assessing social and intersectional biases in contextualized word representations. In *Advances in Neural Information Processing Systems (NeurIPS '19, Vol. 32)*. Curran Associates, Red Hook, NY, 12 pages.

[67] Flickr Team. [n.d.]. *Flickr Terms of Use*. `https://www.flickr.com/creativecommons/` Last accessed 2021-05-25.

[68] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS '20, Vol. 33)*. Curran Associates, Red Hook, NY, 12388–12401.

[69] Mei Wang and Weihong Deng. 2020. Mitigating bias in face recognition using skewness-aware reinforcement learning. In *Conference on Computer Vision and Pattern Recognition (CVPR '20)*. IEEE/CVF, New York, 9322–9331.

[70] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. 2019. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *International Conference on Computer Vision (ICCV '19)*. IEEE, New York, 5310–5319.

[71] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. 2020. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Conference on Computer Vision and Pattern Recognition (CVPR '20)*. IEEE/CVF, New York, 8919–8928.

[72] Michael Wick, Swetasudha Panda, and Jean-Baptiste Tristan. 2019. Unlocking fairness: a trade-off revisited. In *Advances in Neural Information Processing Systems (NeurIPS '19, Vol. 32)*. Curran Associates, Red Hook, NY, 10 pages.

[73] Allison Woodruff, Sarah E Fox, Steven Rousso-Schindler, and Jeffrey Warshaw. 2018. A qualitative exploration of perceptions of algorithmic fairness. In *Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, 14 pages.

[74] Sirui Yao and Bert Huang. 2017. Beyond parity: fairness objectives for collaborative filtering. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2925–2934.

[75] Sirui Yao and Bert Huang. 2017. New fairness metrics for recommendation that embrace differences. *arXiv preprint arXiv:1706.09838* (2017).

[76] Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. 2016. Situation recognition: Visual semantic role labeling for image understanding. In *Conference on Computer Vision and Pattern Recognition*. IEEE, New York, 5534–5542.

[77] Mikhail Yurochkin and Yuekai Sun. 2021. SenSeI: Sensitive Set Invariance for Enforcing Individual Fairness. In *International Conference on Learning Representations*. `https://openreview.net/forum?id=DktZb97_Fx`

[78] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *International Conference on Artificial Intelligence and Statistics (AISTATS '17, Vol. 54)*. PMLR, Fort Lauderdale, 962–970.

[79] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Conference on AI, Ethics, and Society (AIES '18)*. AAAI/ACM, New York, 335–340.

[80] 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 *Conference on Empirical Methods in Natural Language Processing (EMNLP '17)*. ACL, Stroudsburg, PA, 2979–2989.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 7
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [N/A]
(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See supplementary material
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report the range and standard deviation for all the experiments with multiple fixed-seed training runs.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix B.4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Table 1 in Section 2.3 and Appendix B.1
   (b) Did you mention the license of the assets? [Yes] See Appendix B.1
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See supplementary material
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We use training data that are publicly available. To the best of our knowledge, the authors of the dataset did not specifically discuss the inclusion of personally identifiable information.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]