# Fairness Reprogramming

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

## Abstract

Despite a surge of recent advances in promoting machine Learning (ML) fairness, the existing mainstream approaches mostly require training or finetuning the entire weights of the neural network to meet the fairness criteria. However, this is often infeasible in practice for those large-scale trained models due to large computational and storage costs, low data efficiency, and model privacy issues. In this paper, we propose a new generic fairness learning paradigm, called FAIRREPROGRAM, which incorporates the model reprogramming technique. Specifically, FAIRREPROGRAM considers the neural model fixed, and instead appends to the input a set of perturbations, called the fairness trigger, which is tuned towards the fairness criteria under a min-max formulation. We further introduce an information-theoretic framework that explains why and under what conditions fairness goals can be achieved using the fairness trigger. We show both theoretically and empirically that the fairness trigger can effectively obscure demographic biases in the output prediction of fixed ML models by providing false demographic information that hinders the model from utilizing the correct demographic information to make the prediction. Extensive experiments on both NLP and CV datasets demonstrate that our method can achieve better fairness improvements than retraining-based methods with far less training cost and data dependency under two widely-used fairness criteria.

## 1 Introduction

Fairness in machine learning (ML) has become a critical concern. Due to the biases in data collection, the output prediction is often spuriously correlated with some demographic attributes, which are thus undesirably incorporated into the decision-making process of machine learning models. For example, it is found that some abusive language detection systems tend to classify texts that contain mere mentioning of certain minority groups, *e.g.,* homosexual groups, as abusive content, even though the texts themselves are not abusive at all [1, 2]. Despite the recent advances in fairness promoting learning method [3–7], the existing mainstreaming approaches mostly require retraining or finetuning the entire model parameters towards an extra fairness objective. However, this is often infeasible in practice, particularly for those well-trained large-scale models, due to the huge computation and storage costs. In addition, for machine learning models that are deployed as a service, model retraining is hindered by limited access to the model parameters.

Recently, model reprogramming has emerged as an alternative technique to model finetuning. In particular, model reprogramming considers the pre-trained model fixed, and instead modifies their input to re-purpose the model towards different objectives. For example, it is shown that a well-crafted input perturbation can re-program an ImageNet classifier to solve the task of counting squares in an image [8, 9]. It is also shown that by learning task-specific embedding prompts concatenated to the inputs, pre-trained language models can achieve better performances than full-parameter tuning in natural language understanding tasks [10–12] Compared with finetuning methods, model reprogramming enjoys lower cost, better scalability, and requires less access to the model parameters.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Hence here come our research questions - *Can model reprogramming techniques be applied to fairness objectives? If so, why and how would it work?*

In this paper, we revisit the model reprogramming and propose a novel generic fairness learning paradigm, called FAIRREPROGRAM. In particular, FAIRREPROGRAM perturbs the input by appending to the input a global constant vector/feature, called the *fairness trigger*, which is optimized towards the fairness objective under a min-max framework. FAIRREPROGRAM is a generic framework that works for various tasks and domains. We further introduce an information-theoretic framework that explains why and under what conditions fairness goals can be achieved using a constant fairness trigger. We show theoretically and empirically that the fairness trigger can effectively obscure demographic biases in the output prediction of fixed ML models by providing false demographic information that hinders the model from utilizing the correct demographic information to make predictions.

We perform extensive experiments across various NLP and CV datasets with in-the-wild biases. The results show that FAIRREPROGRAM can consistently achieve better fairness improvement with the retraining-based methods under the two widely-used fairness notions, but with far less training cost or trade-off in accuracy. For example, with comparable accuracy, our method can outperform the retraining based baseline with 10.5% and 36.5% lower bias scores over two fairness criteria in the `CelebA` dataset. In addition, FAIRREPROGRAM demonstrates outstanding transferability and interpretability. Our theoretical analysis and empirical findings can provide useful insights toward more practical, scalable, and flexible fairness learning paradigms.

## 2 Related Work

**Fairness in ML** Fairness problems in ML models have received increasing attention from both industry [13] and academia [14–17]. There has been a myriad of fairness definitions in the literature [18, 14, 19, 20]. Among them, group fairness notions are one of the most popular [21–23], which require ML models to perform similarly for different demographic groups. In this paper, we mainly focus on the two most widely-used group fairness definitions, demographic parity [21] and equalized odds [22], but it is worth mentioning that our method is general for other fairness notions. Existing fairness promoting methods can be broadly categorized into pre-processing, in-processing, and post-processing methods [24]. Pre-processing methods calibrate the training data to remove the spurious correlations and train fair model on the modified data [25–28, 2, 1, 29, 30]. In-processing methods work on training ML with extra fairness-aware regularization [3–7, 31]. For example, an adversarial framework is introduced to train model parameters to meet fairness requirements [32]. In our method, we adopt a similar adversarial loss but optimize the fairness triggers with a fixed model. Despite the effectiveness, these methods usually consider training fair models from scratch and do not directly apply to already-trained models. Post-processing methods focus on calibrating trained ML models to be fair [24]. Many of them modify the model outputs to meet the fairness criteria [18, 22, 33–43]. For example, the model outputs are directly modified to meet equalized odds by solving an optimization problem [22]. Alternatively, a boosting-based method is introduced to calibrate model outputs [39].

**Model reprogramming** Model reprogramming [44, 9, 8, 45–47] aims to repurpose an already trained neural network for different tasks. Different from the typical transfer learning that requires modifying the structure and parameters of the given pre-trained model, reprogramming technology instead designs a trainable program appended to the input, while keeping the pre-trained model intact. The model reprogramming technology can be designed in the form of an input-agnostic perturbation [44, 8] or a trainable input transformation function together with the label mapping from the source domain to the target domain [9]. In particular, the feasibility of designing a universal input perturbation to reprogram a well-trained ImageNet classifier to the CIFAR-10 dataset is demonstrated in the white-box setting [8]. As an exploration to implement reprogramming in the discrete scenario, another work [45] successfully reprograms the text classification neural network for alternate classification tasks. This work also shows the possibility of developing reprogramming in the black-box setting, where the reprogrammer may not have the access to the parameters of the target model. Recent work [46] shows the possibility of repurposing deep neural networks designed for image classifiers for the natural language processing and other sequence classification tasks. It is argued the success of the reprogramming lies in the size of the average input gradient and the input dimension is crucial to the performance of the reprogrammer [47]. A highly related topic to model reprograming is prompt learning in NLP [11]. It is shown that by designing designated

text prompts appended to inputs, pre-trained language models could be re-directed to perform well under downstream tasks in a few-shot setting [48]. Prompt-based tuning methods have become the mainstream and achieve better performance than fine-tuning in many scenarios [49–52]. Seminal works about prompt learning can be found in [11, 53]. *However*, nearly all existing methods focus on using model reprogramming to improve accuracy in domain-transfer tasks and to our best knowledge, our work is the first to generalize model reprogramming to improve fairness of a trained model.

# 3 Fairness Reprogramming

In this section, we will introduce the FAIRREPROGRAM algorithms. As some notations, upper-cased letters, $\boldsymbol{X}$ and $X$, denote random vectors and random variables, respectively; lower-cased letters, $\boldsymbol{x}$ or $x$, denote deterministic vectors and scalars respectively. $p_{\boldsymbol{X}}(\cdot)$ or $p(\boldsymbol{X})$ denote the probability density function of the (discrete) random variable $\boldsymbol{X}$.

## 3.1 Problem Formulation

Consider a classification task, where $\boldsymbol{X}$ represents the input feature, and $Y$ represents the output label. In addition, there exists some sensitive attributes or demographic group, $Z$, that may be spuriously correlated with $Y$. There is a pre-trained classifier, $f^*(\cdot)$, that predicts $Y$ from $\boldsymbol{X}$, *i.e.* $\hat{Y} = f^*(\boldsymbol{X})$. The weights of the classifier are considered fixed (hence the superscript $*$). Unfortunately, due to the spurious correlation between $Z$ and $Y$, the classifier may be biased against certain demographics.

Our goal is to improve the fairness of the classifier by modifying the input $\boldsymbol{X}$, rather than modifying the classifier's fixed weights. In particular, we aim to achieve either of the following fairness criteria.

$$\textbf{Equalized Odds:} \quad \hat{Y} \perp Z|Y, \quad \text{or} \quad \textbf{Demographic Parity:} \quad \hat{Y} \perp Z, \tag{1}$$

where $\perp$ denotes independence. The following two subsection will explain how to modify input and design the optimization objective respectively.

## 3.2 Modifying the Input Features

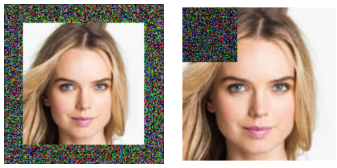
Input modification primarily involves appending a *fairness trigger* to the input. Formally, the input modification takes the following generic form:

$$\tilde{\boldsymbol{X}} = m(\boldsymbol{X}; \boldsymbol{\theta}, \boldsymbol{\delta}) = [\boldsymbol{\delta}, g(\boldsymbol{X}; \boldsymbol{\theta})], \tag{2}$$

where $\tilde{\boldsymbol{X}}$ denotes the modified input; $[\cdot]$ denotes vector concatenation. As can be observed, the input modification consists of two steps. First, $\boldsymbol{X}$ is fed through a transformation function $g(\cdot; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the hyper-parameters of the transformation function. The actual form of $g(\cdot; \boldsymbol{\theta})$ is contingent upon different applications and modalities, but a general requirement is that $g(\cdot; \boldsymbol{\theta})$ should largely retain the information necessary for classification. The second step is to append a fairness trigger, $\boldsymbol{\delta}$, to the input, which is a vector that can be optimized over. It is important to note that $\boldsymbol{\delta}$ is a *constant* – different inputs get appended the same trigger. Although it does not seem intuitive, we will soon show that a constant trigger is all you need to achieve fair prediction on all different inputs.

Below are specific forms of transformations (Eq. (2)) we use.

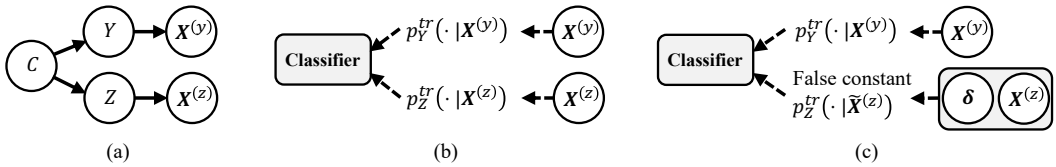
**Text Classification** In text classification, $\boldsymbol{X}$ represents a sequence of input token embeddings. To modify the input, we simply append a fixed number of embeddings after the input text. In this case, $g(\cdot; \boldsymbol{\theta})$ is the identity mapping, and $\boldsymbol{\delta}$ corresponds to the appended embeddings.



(a) Patch trigger (b) Border trigger

Figure 1: Demonstration of the border and patch trigger applied on an image from `CelebA` [54].

**Image Classification** In image classification, $\boldsymbol{X}$ represents the (vectorized) input image. Unlike text classification, where the input can have a variable length, the length of the input to the image classification network is fixed. We thus apply the following two approaches to append the trigger, as shown in Fig. 1. The first approach, called the *patch approach*, removes a patch from the original image, and appends a trigger the same size as the patch to the patch location (as shown in Fig. 1(a)). In this case, $g(\cdot; \boldsymbol{\theta})$ is a function that removes the patch dimension and retain the rest, with $\boldsymbol{\theta}$ representing the patch location; $\boldsymbol{\delta}$ represents the trigger




(a) (b) (c)

Figure 2: Illustration of why fairness trigger works. (a) The data generation process. (b) The information flow from data to the classifier through the sufficient statistics. (c) Fairness trigger strongly indicative of a demographic group can confuse the classifier with a false demographic posterior, and thus preventing the classifier from using the correct demographic information.

feature that replaces the patch. The second approach, called the *border approach*, shrinks the image to a smaller image, and then append the trigger at the border (as shown in Fig. 1(b)). In this case, $g(\cdot;\boldsymbol{\theta})$ is a function that shrinks the image, and $\boldsymbol{\delta}$ represents the trigger feature at the border.

### 3.3 Optimization Objective

Our optimization objective is as follows

$$\min_{\boldsymbol{\delta},\boldsymbol{\theta}} \mathcal{L}_{util}(\mathcal{D}_{tune}, f^* \circ m) + \lambda \mathcal{L}_{fair}(\mathcal{D}_{tune}, f^* \circ m), \tag{3}$$

where $m = m(\cdot;\boldsymbol{\theta},\boldsymbol{\delta})$ represents the input modification function as in Eq. (2); $\circ$ represents nested functions; $\mathcal{D}_{tune}$ represents the dataset that are used to train the fairness trigger. Note that this is different from the dataset where the classifier, $f^*$, is pre-trained.

The first loss term, $\mathcal{L}_{util}$, is the utility loss function of the task. For classification tasks, $\mathcal{L}_{util}$ is usually the cross-entropy loss, *i.e.*,

$$\mathcal{L}_{util}(\mathcal{D}_{tune}, f^* \circ m) = \mathbb{E}_{\boldsymbol{X},Y\sim\mathcal{D}_{tune}}[\mathrm{CE}(Y, f^*(m(\boldsymbol{X})))], \tag{4}$$

where $\mathrm{CE}(\cdot,\cdot)$ denotes the cross-entropy loss.

The second loss term, $\mathcal{L}_{fair}$, encourages the prediction to follow the fairness criteria as in Eq. (1). According to Eq. (1), $\mathcal{L}_{fair}$ should measure how much information about $Z$ is in $\hat{Y}$. To measure this, we introduce another network, called the discriminator, $d(\cdot;\boldsymbol{\phi})$, where $\boldsymbol{\phi}$ represents its parameters. If the equalized odds criterion is applied, then $d(\cdot;\boldsymbol{\phi})$ should predict $Z$ from $\hat{Y}$ and $Y$; if the demographic parity criterion is applied, then the input to $d(\cdot;\boldsymbol{\phi})$ would just be $\hat{Y}$. In the following, we will focus on equalize odds criterion for conciseness. Then, the information of $Z$ can be measured by maximizing the *negative* cross-entropy loss for the prediction of $Z$ over the discriminator parameters, *i.e.*,

$$\mathcal{L}_{fair}(\mathcal{D}_{tune}, f^* \circ m) = \max_{\boldsymbol{\phi}} \mathbb{E}_{\boldsymbol{X},Y,Z\sim\mathcal{D}_{tune}}[-\mathrm{CE}(Z, d(f^*(m(\boldsymbol{X})), Y;\boldsymbol{\phi}))]. \tag{5}$$

By plugging Eqs. (4) and (5) into (3), we can see that the entire optimization objective becomes a min-max framework, where the discriminator tries to improve its prediction of $Z$ while the fairness trigger tries to make the prediction worse. As shown in [32], when the discriminator cannot predict $Z$ better than chance, the aforementioned fairness criteria can be achieved.

### 3.4 Why Does It Work?

It is not immediately straightforward why a *global* trigger can obscure the demographic information for *any* input. In this section, we will propose a information-theoretic framework that illustrates one of the mechanisms through which the trigger can remove the demographic information.

Our theoretical framework builds upon the data generation process as shown in Fig. 2(a). Specifically, we assume that $\boldsymbol{X}$ consists of a set of features, *i.e.* $\boldsymbol{X} = [\boldsymbol{X}_1, \cdots, \boldsymbol{X}_T]$, where $T$ is the total number of features. In text classification, a feature can be a word or a word piece; in image classification, a feature can be specific shapes, colors, patterns, *etc*. Assume that these features can be divided into two groups. The first group, denoted as $\boldsymbol{X}^{(y)}$, consists of features that are directly governed by the output label $Y$; the second group, denoted as $\boldsymbol{X}^{(z)}$, consists of featuers that are directly governed by the demographic information $Z$. $Z$ and $Y$ can be spuriously correlated, *i.e.* there can be common confounders, $C$, between $Z$ and $Y$. As a result, both $\boldsymbol{X}^{(y)}$ and $\boldsymbol{X}^{(z)}$ are indicative of $Y$.

To further simplify our theoretical analysis, we consider a bag-of-feature scenario, where each feature in $\boldsymbol{X}^{(y)}$ is drawn from the vocabulary set $\mathcal{X}^{(y)}$, and each feature in $\boldsymbol{X}^{(z)}$ is drawn from the vocabulary

set $\mathcal{X}^{(z)}$. There should not be any overlap between the two vocabulary sets, *i.e.* $\mathcal{X}^{(y)} \cap \mathcal{X}^{(z)} = \varnothing$. Otherwise it violates our assumption that demographic-related features are biased features.

It can be shown (in Appendix C) that the posterior distributions, $p_Y(\cdot|\boldsymbol{X}^{(y)})$ and $p_Z(\cdot|\boldsymbol{X}^{(z)})$, are the sufficient statistics of $\boldsymbol{X}^{(y)}$ and $\boldsymbol{X}^{(z)}$ respectively for inferring $Y$. In other words, these two posterior distributions summarize all the information about $\boldsymbol{X}^{(y)}$ and $\boldsymbol{X}^{(z)}$ that the classifier needs to know to predict $Y$. Therefore, we assume that the classifier takes the following generic form

$$\hat{Y} = f^*(\boldsymbol{X}) = h(p_Y^{tr}(\cdot|\boldsymbol{X}^{(y)}), p_Z^{tr}(\cdot|\boldsymbol{X}^{(z)})). \tag{6}$$

Note that we add a superscript, $tr$, to emphasize that the probability distributions are over the data set where the classifier is trained, because the classifier has never been trained on inputs modified with the fairness trigger. Eq. (6) encompasses many common decision functions. For example, it can be shown (in Appendix C) that the posterior distribution $p(Y|\boldsymbol{X})$, which is the minimizer of the cross-entropy loss, is a special case of Eq. (6).

As illustrated in Fig. 2(b), $p_Y(\cdot|\boldsymbol{X}^{(y)})$ and $p_Z(\cdot|\boldsymbol{X}^{(z)})$ provide two sets of information from input features. $p_Y(\cdot|\boldsymbol{X}^{(y)})$ provides the *unbiased* information, because a desirable fair classifier should rely only upon $p_Y(\cdot|\boldsymbol{X}^{(y)})$ to make a decision. On the other hand, $p_Z(\cdot|\boldsymbol{X}^{(z)})$ provides the *biased* information, because it conveys the demographic information. In other words, the fairness goals can be achieved by cutting off the biased information path. Therefore, our research question boils down to: is it possible to cut off the biased information path with a global fairness trigger $\boldsymbol{\delta}$?

Without loss of generality, assume that $\boldsymbol{\delta}$ consists of only one feature. Consider the case where $\boldsymbol{\delta}$ is a demographic feature, *i.e.* $\boldsymbol{\delta} \in \mathcal{X}^{(z)}$. In this case, we assume the transformed input as defined in Eq. (2) can also be divided into two groups:

$$\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{X}}^{(y)}, \tilde{\boldsymbol{X}}^{(z)}], \quad \text{where} \quad \tilde{\boldsymbol{X}}^{(y)} = g(\boldsymbol{X}^{(y)}), \quad \tilde{\boldsymbol{X}}^{(z)} = [\boldsymbol{\delta}, g(\boldsymbol{X}^{(z)})]. \tag{7}$$

The following theorem states our main conclusion:

**Theorem 1.** *Under the assumptions in Eq.* (6) *and* (7)*, and some additional regularity conditions*[1]*, if the fairness trigger* $\boldsymbol{\delta}$ *is indicative of a certain demographic group* $z$*, then*

$$\lim_{p^{tr}(Z=z|\boldsymbol{X}_0^{(z)}=\boldsymbol{\delta})\to 1} MI(\hat{\tilde{Y}}, Z|Y) = 0, \tag{8}$$

*where MI means mutual information;* $\hat{\tilde{Y}} = f^*(\tilde{\boldsymbol{X}})$ *is the classifier's prediction after input is modified.*

$p^{tr}(Z = z|\boldsymbol{X}_0^{(z)} = \boldsymbol{\delta}) \to 1$ means that the fairness trigger is very strongly indicative of the demographic group $z$. Therefore, Thm. 1 essentially states that if the prepended trigger feature is very strongly indicative of a certain demographic group, then equalized odds can be achieved. A formal proof is presented in Appendix C. Here we would like to give an intuitive explanation. When $p^{tr}(Z = z|\boldsymbol{X}_0^{(z)} = \boldsymbol{\delta}) \to 1$, it will also happen that $p^{tr}(Z = z|\boldsymbol{X}^{(z)} = \tilde{\boldsymbol{X}}^{(z)}) \to 1$. In other words, the fairness trigger $\boldsymbol{\delta}$ would overshadow the rest of the demographic features and 'trick' the classifier into believing all the different inputs belong to the same demographic group $z$. As a result, the second argument in Eq (6) would reduce to a constant (1 for demographic group $z$ and 0 elsewhere), effectively blocking the biased information path, as shown in Fig. 2(c). Note that the premise for the fairness trigger to work is that the classifier has never seen the modified input. Otherwise, the classifier will be able to learn to ignore the constant trigger and still elicit the true demographic information from input.

# 4 Experiments

In this section, we evaluate the effectiveness of FairReprogram on both NLP and CV applications in terms of accuracy, fairness, performances under low-data regime, transferability and interpretability.

## 4.1 Experiment Setup

**Datasets** We consider the following two commonly used NLP and CV datasets:

- `Civil Comments` [55, 56]: The dataset contains 448k texts with labels that depict the toxicity of each input. The demographic information of each text is provided.

[1]Formal assumptions stated in the appendix.

- `CelebA` [54]: The dataset contains over 200k human face images and each contains 39 binary attribute annotations. We follow the conventional setting [54] that adopts the hair color prediction task in our experiment and uses gender annotation as the demographic information. [57–59]

For both datasets, we split the entire data into a training set, a tuning set, a validation set, and a testing set. The training set is used for the base model training, *i.e.*, to obtain a biased model for reprogramming. The tunning set and validation set are used for trigger training and hyper-parameter selection. We report our results on the testing set. It is worth mentioning that there is no overlapping data between different sets and the size of the tuning set is much smaller than the training one. Specifically, we set the size ratio between the tunning set and the training as $1/5$ and $1/100$ for `Civil Comments` and `CelebA`, respectively. The full statistics of the datasets can be found in Appendix A.1.

**Metrics** Besides the model accuracy, we introduce two empirical fairness metrics, one under each of the two fairness criteria as in Eq. (1). For binary classification, the metrics are calculated as:

$$\textbf{DP:} \sum_{z\in\mathcal{Z}} |\mathrm{p}(\hat{Y}=1) - \mathrm{p}(\hat{Y}=1|Z=z)|, \quad \textbf{EO:} \sum_{z\in\mathcal{Z}} \left(|\mathrm{FPR} - \mathrm{FPR}_z| + |\mathrm{FNR} - \mathrm{FNR}_z|\right)/2,$$

where DP and EO stand for demographic parity and equalized odds respectively. FPR and FNR are the false positive/negative rate, and the subscript $z$ denotes the score is calculated within a specific demographic group $Z = z$. For example, $\mathrm{FPR}_{male}$ indicates the false positive rate calculated over all examples with the "male" annotation. For a multi-class setting, the bias scores are first calculated similarly using one-vs-all for each class and then averaged across different classes. All reported results are the average of three different random runs. It can be shown that these metrics are non-negative, and will become zero when their corresponding fairness criteria are achieved. For better elaboration, we report the negative bias scores in our experiments, so the larger these negative scores are, the better the model satisfies the corresponding fairness criteria.

**Baselines and implementation details** We consider the following models for comparison:

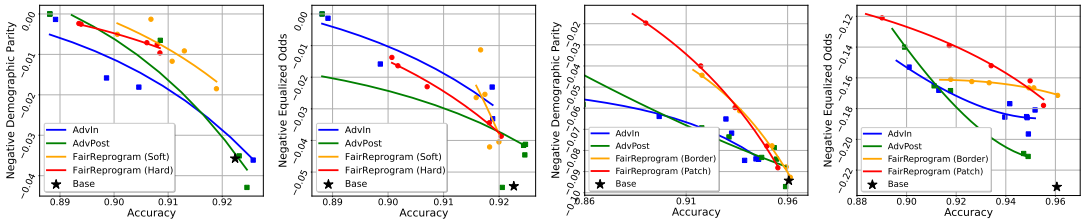
- BASE: the base model to be reprogrammed, trained with the cross-entropy loss on the training set.
- ADVIN [32]: an in-processing adversarial training method that optimizes both model accuracy and fairness using the training set.
- ADVPOST: a post-processing variant of ADVIN, which fine-tunes the BASE model with the same fairness-aware adversarial objectives as ADVIN, but using the (low-resource) tunning set only.

For NLP experiments, we use a pre-trained BERT [60] to obtain the BASE and ADVIN models. We use ADAMW [61] as the optimizer, and set the learning rate to $10^{-5}$ for all baselines and $0.1$ for FAIRREPROGRAM. For CV experiments, we consider a RESNET-18 [62] that pre-trained on ImageNet. ADAM [63] with learning rates of $0.001$ and $0.05$ are used to train the baselines and our method, respectively. The discriminator used in ADVIN, ADVPOST and FAIRREPROGRAM is a three-layer MLP, and the parameters are optimized using ADAM with a learning rate of $0.01$. We pick the best model based on the accuracy (for the BASE) or the bias scores (for all other debiasing methods) of the validation set. More details can be found in Appendix A.2.

Next we introduce the implementation details of the triggers for different variants of FAIRREPROGRAM. For image classification task, we adopt the border and patch trigger as shown in Fig. 1, termed FAIRREPROGRAM (BORDER) and FAIRREPROGRAM (PATCH) correspondingly. We define the trigger size as the width of the trigger frame for border trigger and the width of the square patch for patch trigger. Unless otherwise stated, the default trigger size for each setting are 20 and 80.

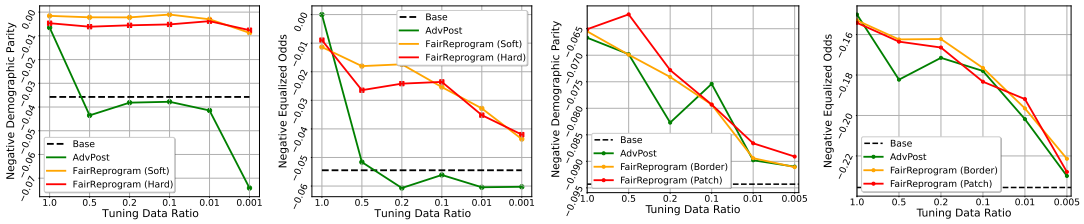
For text classification task, we introduce a probability vector $\boldsymbol{v}_i$ to control the selection of trigger word for each position $i$. Specifically, we have the trigger $\boldsymbol{\delta}_i = \boldsymbol{E}\boldsymbol{v}_i$ where $\boldsymbol{E}$ represents the pretrained word embedding matrix of BERT. Then we simply concatenate $\boldsymbol{\delta}$ after all input texts[2] in the embeddings space as the fairness trigger. We introduce two types of trigger. The first type, called FAIRREPROGRAM (SOFT) , uses continuous $\boldsymbol{v}_i$'s, and each $\boldsymbol{v}_i$ is projected onto the continuous probability simplex using the bisection algorithm after each training step. The second type, called FAIRREPROGRAM (HARD), discretizes each $\boldsymbol{v}_i$ into a one-hot vector $\hat{\boldsymbol{v}}_i$ via $\arg\max$ operation. We adopt the straight through technique [64] to update $\boldsymbol{v}_i$ during training. The triggers found by FAIRREPROGRAM (HARD) enjoy better interpretability as they correspond to a sequence of word tokens. Unless specified otherwise, we set the trigger word number as five for our experiments.

[2]The trigger is appended as a suffix after all input tokens but before [SEP] for BERT.




(a) `Civil Comments` (b) `CelebA`

Figure 3: Results on (a) `Civil Comments` and (b) `CelebA`. We report the negative DP (left) and the negative EO (right) scores. For each method, we vary the trade-off parameter $\lambda$ (as shown in (3)) to record the performance. The closer a dot to the upper-right corner, the better the model is. We consider five different $\lambda$s for each method. The solid curve is the fitted polynomial with order 30.




(a) `Civil Comments` (b) `CelebA`

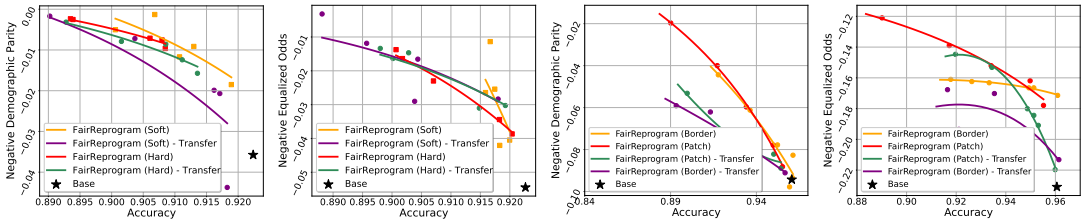
Figure 4: Results on (a) `Civil Comments` and (b) `CelebA` with different tuning data ratio. We report the negative DP (left) and negative EO (right) scores. We consider a fixed BASE model trained with training set, whose negative bias scores are presented as a black dashed line. Then we train other methods with different tuning data ratio to promote fairness of the BASE model.

### 4.2 Results

Fig. 3 shows the performance of the proposed FAIRREPROGRAM with other baselines on both NLP (subfigure (a)) and CV (subfigure (b)) datasets using DP (left) and EO (right) metrics. In each subfigure, the data samples of the same method (dots in the same color) are generated by explicit changing the adversary weight $\lambda$ in (3), which controls the trade-off between fairness and accuracy. We further fit the data with polynomial regression to present the curves. Appendix A.2 shows the detailed $\lambda$ choices for different methods. Here are our key observations. First, our method improves the fairness of the BASE model. In particular, our methods (both orange and red curves) achieve higher negative DP and EO scores with a comparable classification accuracy. Second, our method enjoys a better fairness-accuracy trade-off compared with all other baselines. Specifically, the curves of our method lie farther to the upper-right corner of the plots, which implies that our method improves model fairness with fewer sacrifices on accuracy. It is also worth noting that although ADVIN achieves good fairness scores, it uses much more data for training.

**Limited data setting** We further evaluate ADVPOST and FAIRREPROGRAM with decreasing the number of data in the tuning set. Specifically, we fix a $\lambda$ for each method such that all methods achieve comparable bias score with full tuning set. The detailed $\lambda$ choices are provided in Appendix A.2. Then we apply these methods to subsets of the tuning set with different proportions. The results are shown in Fig. 4. There are two key observations. First, our method can consistently improve fairness upon BASE model even with 1% tuning data, indicating a high data efficiency of FAIRREPROGRAM. Second, FAIRREPROGRAM achieves better fairness than ADVPOST does when tuning data number decreases. For example, in Fig. 4 (a), the curve of our method is significantly above the ADVPOST as tuning data decreases. When the tuning set size is extremely small, ADVPOST significantly deteriorates and even underperforms the BASE model.

**Transferability** Next, we show the transferability of the fairness triggers found by FAIRREPROGRAM. We first tune the triggers with a BASE source model and then apply the trigger on a target model trained with a different random seed. The results are shown in Fig. 5. As can be seen, FAIRREPROGRAM achieves comparable fairness-accuracy trigger on both the source model and the target model, indicating our method has a good transferability. This intriguing property brings two benefits of our method: ① if ML model parameters are infeasible (*e.g.* when ML models are




(a) `Civil Comments` (b) `CelebA`

Figure 5: Results in the transfer setting. We report negative DP (left) and negative EO (right) scores. The triggers are firstly trained in a BASE model. Then we evaluate the triggers based on another unseen BASE model. We change the parameter $\lambda$ to trade-off accuracy with fairness and draw the curves in the same way with Fig. 3. The ★ point corresponds to the average of all BASE models with different random seeds.

| *Text (Non-toxic)* | | *Predicted Toxicity* |
|---|---|---|
| **muslims** need to take a look in **the** **mirror** | | 0.149 |
| **muslims** need to take a look in the mirror | same **diocesan** **bula** **rev** **proceedings** | 0.069 |
| **muslims** need to take a look in the mirror | soto **cc** rib **hebrew** armenian | 0.054 |
| **muslims** need to take a look in the mirror | **paul** long course **parish** **body** | 0.073 |

Figure 6: A text example from `Civil Comments` with INTEGRATED GRADIENT [65, 66] highlighting important words that influence ERM model predictions. The text is concatenated with three triggers generated with different adversary weight. **Green highlights** the words that lean to toxic predictions and **Red highlights** non-toxic leaning words. The model prediction tends to be correct after adding the triggers.

provided as services), the users could train a surrogate model and tune the trigger based on it to promote fairness of the original model; ② when ML model parameters are updated with new data (*e.g.* online learning), the user could still use the original trigger for fixing fairness problems. We further elaborate the results of FAIRREPROGRAM in a black-box setting in Appendix B.

**Input saliency attribution.** Fig. 6 and 7 compare the saliency maps of some example inputs with and without the fairness triggers. Specifically, For the NLP applications, we extract a subset of `Civil Comments` with religion-related demographic annotations, and apply IG [65] to localize word pieces that contribute most to the text toxicity classification. For the CV application, we use GradCam [67] to identify class-discriminative regions of `CelebA`'s test images. As shown in Fig. 6, our fairness trigger consists of a lot of religion-related words (*e.g.*, diocesan, hebrew, parish). Meanwhile, the predicted toxicity score of the benign text starting from 'muslims' significantly reduces. These observations verify our theoretical hypothesis that the fairness trigger is strongly indicative of a certain demographic group to prevent the classifier from using the true demographic information. In addition, Fig. 7 presents the input saliency maps on two input images with respect to their predicted labels, non-blond hair and blond hair, respectively. As can be observed, when there is no fairness trigger, the saliency region incorrectly concentrates on the facial parts, indicating the classifier is likely to use biased information, such as gender, for its decision. With the fairness trigger, the saliency region moves to the hair parts, which matches the behavior of ADVIN. These results confirm that our fairness trigger can drive models to make fairer predictions.

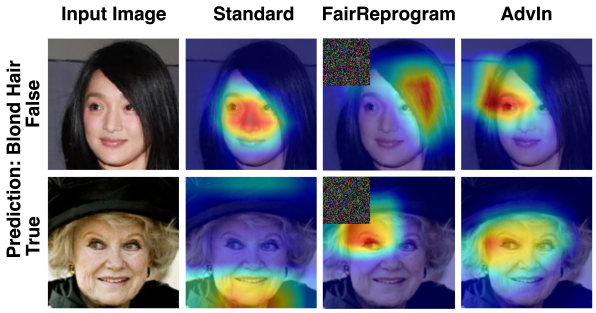
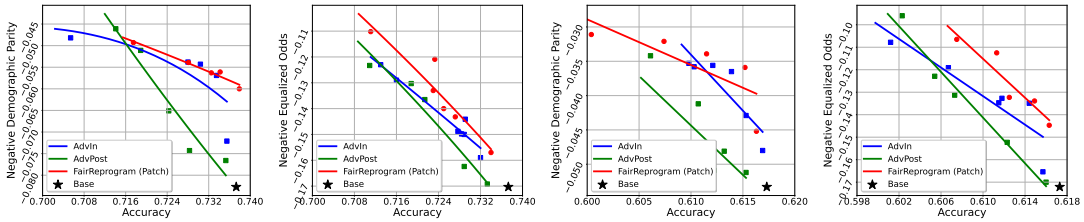



Figure 7: Gradient-based saliency map visualized with GRAD CAM [67] of different methods. The highlighted zones (marked in red) depicting regions exerting major influence on the predicted labels (non-blond hair v.s. blond hair) in each row, which also depict the attention of the model on the input image.

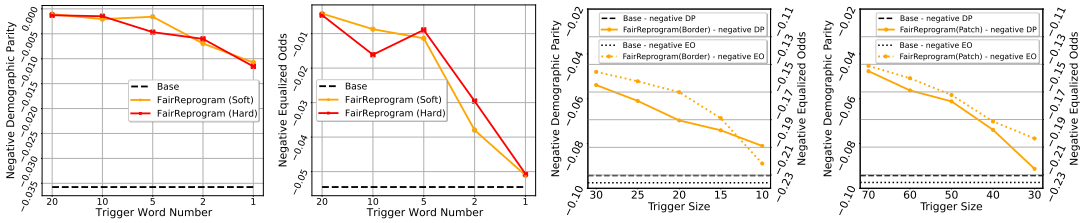
### 4.3 Multi-Class Classification

To extend our evaluation to a multi-class setting, we use the `CelebA` dataset and select $n$ binary attributes that may be spuriously correlated with *gender* [57–59]. Then, following [68], we construct




(a) 8-class negative DP (b) 8-class negative EO (c) 16-class negative DP (d) 16-class negative EO

Figure 8: Performance of multi-class classification. For (a) and (b), we use the attributes *Blond Hair, Smiling, Attractive* for multi-class construction. We add an addition attribute *Wavy Hair* for (c) and (d).




(a) `Civil Comments` (b) `CelebA`

Figure 9: Ablation study of the trigger size. We evaluate the bias scores with different trigger word numbers (`Civil Comments`) and different trigger size (`CelebA`) with fixed adversary weight $\lambda$.

data groups by enumerating all $2^n$ possible binary vectors, where each dimension corresponds to a binary attribute. We index these vectors and treat them as the class labels. Fig. 8 shows the accuracy-fairness trade-off curves similar to Fig. 3. It can be observed that our method outperforms the other methods as the red curves are closer to the top-right corner. Also, as the class label number increases, the post-processing-based ADVPOST falls behind its in-processing counterpart ADVIN, indicating a larger class number may induce more challenges to post-processing methods.

### 4.4 Ablation Studies

We perform an ablation study to investigate the effects of the trigger size. Specifically, we run experiments with different numbers of trigger words / trigger patch sizes on the NLP / CV dataset. We set a $\lambda$ value for each method such that all methods achieve comparable bias scores with the largest trigger size. The detailed $\lambda$ choices can be seen in Appendix A.2. Then we train the triggers with different sizes in the tuning set using the fixed $\lambda$'s. For the text trigger as shown in Fig. 9(a), we see that the negative bias score gets worse as the number of trigger words gets smaller. However, our method can still improve fairness upon the BASE model even with only a one-word trigger. On the other hand, the results with five trigger words and above are all comparable, indicating that five words is enough to achieve the fairness goal. Similarly, for the image trigger as shown in Fig. 9(b), the results suggest a larger trigger would consistently improve fairness. On the other hand, we show that larger trigger size could hurt accuracy in Appendix B, which is similar to the effect of increasing $\lambda$.

## 5 Conclusion

In this paper, we introduce a novel model reprogramming based fairness promoting method, termed FAIRREPROGRAM. Specifically, FAIRREPROGRAM considers a fixed ML model and optimize a set of vectors, named fairness trigger, concatenated on inputs to boost model fairness. We introduce an information-theoretic framework to explain the rationales of why FAIRREPROGRAM can improve model fairness. As implied by our theoretic framework as well as our empirical findings, the fairness trigger can effectively mask out the true demographic information with its strong, false demographic information. Extensive experiments demonstrate that our method could achieve better fairness improvements to retraining based methods with far-less training cost. We further empirically show fairness triggers enjoys great transferability and interpretability. We hope that FAIRREPROGRAM can inspire new fairness learning paradigms that are more feasible and flexible in practice.

# References


[1] Lucas Dixon, John Li, Jeffrey Scott Sorensen, Nithum Thain, and Lucy Vasserman, "Measuring and mitigating unintended bias in text classification," *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018.

[2] J. Park, Jamin Shin, and Pascale Fung, "Reducing gender bias in abusive language detection," in *EMNLP*, 2018.

[3] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi, "Fairness constraints: Mechanisms for fair classification," in *AISTATS*, 2017.

[4] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach, "A reductions approach to fair classification," *ArXiv*, vol. abs/1803.02453, 2018.

[5] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *ECML/PKDD*, 2012.

[6] Sina Baharlouei, Maher Nouiehed, and Meisam Razaviyayn, "Rényi fair inference," *arXiv: Learning*, 2019.

[7] Adrián Pérez-Suay, Valero Laparra, Gonzalo Mateo-García, Jordi Muñoz-Marí, Luis Gómez-Chova, and Gustau Camps-Valls, "Fair kernel learning," in *ECML/PKDD*, 2017.

[8] Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein, "Adversarial reprogramming of neural networks," *arXiv preprint arXiv:1806.11146*, 2018.

[9] Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho, "Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9614–9624.

[10] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May, "Warp: Word-level adversarial reprogramming," *arXiv preprint arXiv:2101.00121*, 2021.

[11] Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun, "Openprompt: An open-source framework for prompt-learning," *ArXiv*, vol. abs/2111.01998, 2022.

[12] Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn, "Just train twice: Improving group robustness without training group information," in *ICML*, 2021.

[13] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miroslav Dudík, and H. Wallach, "Improving fairness in machine learning systems: What do industry practitioners need?," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[14] A. Chouldechova and Aaron Roth, "The frontiers of fairness in machine learning," *ArXiv*, vol. abs/1810.08810, 2018.

[15] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth M. Belding-Royer, Kai-Wei Chang, and William Yang Wang, "Mitigating gender bias in natural language processing: Literature review," in *ACL*, 2019.

[16] Ninareh Mehrabi, Fred Morstatter, Nripsuta Ani Saxena, Kristina Lerman, and A. G. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys (CSUR)*, vol. 54, pp. 1 – 35, 2021.

[17] Anjalie Field, Su Lin Blodgett, Zeerak Waseem, and Yulia Tsvetkov, "A survey of race, racism, and anti-racism in nlp," *ArXiv*, vol. abs/2106.11410, 2021.

[18] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel, "Fairness through awareness," *ArXiv*, vol. abs/1104.3913, 2012.

[19] Karima Makhlouf, Sami Zhioua, and Catuscia Palamidessi, "Survey on causal-based machine learning fairness notions," *ArXiv*, vol. abs/2010.09553, 2020.

[20] T. Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang, "Fairness without demographics in repeated loss minimization," in *ICML*, 2018.

[21] Toon Calders and Sicco Verwer, "Three naive bayes approaches for discrimination-free classification," *Data Mining and Knowledge Discovery*, vol. 21, pp. 277–292, 2010.

[22] Moritz Hardt, Eric Price, and Nathan Srebro, "Equality of opportunity in supervised learning," in *NIPS*, 2016.

[23] Tim Räz, "Group fairness: Independence revisited," *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[24] Simon Caton and Christian Haas, "Fairness in machine learning: A survey," *ArXiv*, vol. abs/2010.04053, 2020.

[25] Faisal Kamiran and Toon Calders, "Data preprocessing techniques for classification without discrimination," *Knowledge and Information Systems*, vol. 33, pp. 1–33, 2011.

[26] Richard S. Zemel, Ledell Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork, "Learning fair representations," in *ICML*, 2013.

[27] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Eduardo Scheidegger, and Suresh Venkatasubramanian, "Certifying and removing disparate impact," *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[28] Flávio du Pin Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney, "Optimized pre-processing for discrimination prevention," in *NIPS*, 2017.

[29] Aditya Grover, Kristy Choi, Rui Shu, and Stefano Ermon, "Fair generative modeling via weak supervision," in *ICML*, 2020.

[30] Guanhua Zhang, Bing Bai, Junqi Zhang, Kun Bai, Conghui Zhu, and T. Zhao, "Demographics should not be the reason of toxicity: Mitigating discrimination in text classifications with instance weighting," in *ACL*, 2020.

[31] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh, "Fr-train: A mutual information-based approach to fair and robust training," in *ICML*, 2020.

[32] B. Zhang, Blake Lemoine, and Margaret Mitchell, "Mitigating unwanted biases with adversarial learning," *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018.

[33] Felix Petersen, Debarghya Mukherjee, Yuekai Sun, and Mikhail Yurochkin, "Post-processing for individual fairness," in *NeurIPS*, 2021.

[34] Dennis Wei, Karthikeyan Natesan Ramamurthy, and Flávio du Pin Calmon, "Optimized score transformation for fair classification," in *AISTATS*, 2020.

[35] Pranjal Awasthi, Matthäus Kleindessner, and Jamie H. Morgenstern, "Equalized odds postprocessing under imperfect group information," in *AISTATS*, 2020.

[36] Blake E. Woodworth, Suriya Gunasekar, Mesrob I. Ohannessian, and Nathan Srebro, "Learning non-discriminatory predictors," *ArXiv*, vol. abs/1702.06081, 2017.

[37] Alan Mishler and Edward H. Kennedy, "Fairness in risk assessment instruments: Post-processing to achieve counterfactual equalized odds," *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[38] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang, "Men also like shopping: Reducing gender bias amplification using corpus-level constraints," in *EMNLP*, 2017.

[39] Michael P. Kim, Amirata Ghorbani, and James Y. Zou, "Multiaccuracy: Black-box post-processing for fairness in classification," *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019.

[40] Pranay Kr. Lohia, Karthikeyan Natesan Ramamurthy, Manish Bhide, Diptikalyan Saha, Kush R. Varshney, and Ruchir Puri, "Bias mitigation post-processing for individual and group fairness," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2847–2851, 2019.

[41] Pranay Kr. Lohia, "Priority-based post-processing bias mitigation for individual and group fairness," *ArXiv*, vol. abs/2102.00417, 2021.

[42] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Mark D. M. Leiserson, "Decoupled classifiers for group-fair and efficient machine learning," in *FAT*, 2018.

[43] Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, L. Oneto, and Massimiliano Pontil, "Leveraging labeled and unlabeled data for consistent fair binary classification," in *NeurIPS*, 2019.

[44] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola, "Visual prompting: Modifying pixel space to adapt pre-trained models," *arXiv preprint arXiv:2203.17274*, 2022.

[45] Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov, and Farinaz Koushanfar, "Adversarial reprogramming of text classification neural networks," *arXiv preprint arXiv:1809.01829*, 2018.

[46] Paarth Neekhara, Shehzeen Hussain, Jinglong Du, Shlomo Dubnov, Farinaz Koushanfar, and Julian McAuley, "Cross-modal adversarial reprogramming," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2427–2435.

[47] Yang Zheng, Xiaoyi Feng, Zhaoqiang Xia, Xiaoyue Jiang, Ambra Demontis, Maura Pintor, Battista Biggio, and Fabio Roli, "Why adversarial reprogramming works, when it fails, and how to tell the difference," *arXiv preprint arXiv:2108.11673*, 2021.

[48] Tianyu Gao, Adam Fisch, and Danqi Chen, "Making pre-trained language models better few-shot learners," *ArXiv*, vol. abs/2012.15723, 2021.

[49] Xiang Lisa Li and Percy Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. abs/2101.00190, 2021.

[50] Timo Schick and Hinrich Schütze, "It's not just size that matters: Small language models are also few-shot learners," *ArXiv*, vol. abs/2009.07118, 2021.

[51] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh, "Eliciting knowledge from language models using automatically generated prompts," *ArXiv*, vol. abs/2010.15980, 2020.

[52] Shizhe Diao, Xuechun Li, Yong Lin, Zhichao Huang, and Tong Zhang, "Black-box prompt learning for pre-trained language models," *ArXiv*, vol. abs/2201.08531, 2022.

[53] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ArXiv*, vol. abs/2107.13586, 2021.

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

[55] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Wei hua Hu, Michihiro Yasunaga, Richard L. Phillips, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang, "Wilds: A benchmark of in-the-wild distribution shifts," in *ICML*, 2021.

[56] Jigsaw/Conversation AI, "Jigsaw unintended bias in toxicity classification," 2019.

[57] Tian Xu, Jennifer White, Sinan Kalkan, and Hatice Gunes, "Investigating bias and fairness in facial expression recognition," in *European Conference on Computer Vision*. Springer, 2020, pp. 506–523.

[58] Saloni Dash and Amit Sharma, "Counterfactual generation and fairness evaluation using adversarially learned inference," 2020.

[59] Sunhee Hwang, Sungho Park, Dohyung Kim, Mirae Do, and Hyeran Byun, "Fairfacegan: Fairness-aware facial image-to-image translation," *arXiv preprint arXiv:2012.00282*, 2020.

[60] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv*, vol. abs/1810.04805, 2019.

[61] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[63] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[64] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *ArXiv*, vol. abs/1308.3432, 2013.

[65] Mukund Sundararajan, Ankur Taly, and Qiqi Yan, "Axiomatic attribution for deep networks," *ArXiv*, vol. abs/1703.01365, 2017.

[66] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson, "Captum: A unified and generic model interpretability library for pytorch," 2020.

[67] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[68] Ni Zhuang, Yan Yan, Si Chen, Hanzi Wang, and Chunhua Shen, "Multi-label learning based deep transfer neural network for facial attribute classification," *Pattern Recognition*, vol. 80, pp. 225–240, 2018.

[69] Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jinfeng Yi, Mingyi Hong, Shiyu Chang, and Sijia Liu, "How to robustify black-box ml models? a zeroth-order optimization perspective," *arXiv preprint arXiv:2203.14195*, 2022.

# 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 Appendix D.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix D.
   (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? [Yes] See Appendix C.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix C.
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 Section 4.1 and Appendix A.2.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1 and Appendix A.1A.2.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (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 A.2.
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 Section 4.1.
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
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]