2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)

Provable Fairness Repair for Deep Neural Networks

Jianan Ma*†, Jingyi Wang†§, Qi Xuan‡, Zhen Wang*
*Hangzhou Dianzi University, China
†Zhejiang University, China
‡Zhejiang University of Technology, China
majiananannn@gmail.com, wangjyee@zju.edu.cn, xuanqi@zjut.edu.cn, wangzhen@hdu.edu.cn

Abstract*—Deep neural networks (DNNs) are suffering from ethical issues such as individual discrimination. In response, extensive NN repair techniques have been developed to adjust models and mitigate such undesired behaviors. However, existing fairness repair methods are typically data-centric, which often lack provable guarantees and generalization to unseen samples. To overcome these limitations, we propose PROF, a novel fairness repair framework with provable guarantees. The key intuition of PROF is to leverage interval bound propagation (a widely used NN verification technique) to soundly capture model outputs over the whole set $\mathcal{S}(\boldsymbol{x})$ around a biased sample $\boldsymbol{x}$. The derived bounds are utilized to guide fairness repair which encourages the model to produce consistent outputs on $\mathcal{S}(\boldsymbol{x})$. Specifically, we integrate fairness constraints and model modifications into a unified constraint-solving formulation, which can be transformed to a Mixed-Integer Linear Programming (MILP) problem solvable by off-the-shelf solvers. The solution to the MILP problem effectively induces a repaired model with guaranteed fairness over the whole set $\mathcal{S}(\boldsymbol{x})$. We evaluate PROF on four widely used benchmark datasets and demonstrate that it achieves provable fairness repair, with generalization of up to 95.93% on full datasets and 93.16% on the entire input space. Notably, PROF can be easily configured to support multiple sensitive attributes and more practical fairness definitions, while providing provable repair guarantees and delivering around 90% fairness improvement. Our code is available in this repository.

***Index Terms*—neural network repair, fairness, interval bound propagation**

I. Introduction

Deep neural networks (DNNs) have demonstrated impressive performance in a wide range of applications such as computer vision [1], natural language processing [2], and autonomous driving [3]. Despite the remarkable success, their increased adoption in sensitive domains (e.g., crime risk assessment [4], public policy [5], and credit scoring [6]) has raised concerns that DNNs learning from biased data can produce unfair results. Individual discrimination has become one of the most critical problems, where input pairs differing only in sensitive attributes (e.g., gender, age, etc.) receive different predictions [7]. In response to this threat, numerous studies on DNN fairness testing [8], [9], [10], [11], [12] and verification [13], [14], [15] have been proposed. These techniques aim to either uncover unfairness by generating discriminatory instances or to provide guarantees through formal methods. However, their analysis results cannot provide further solutions as they do not directly mitigate the unfairness. This raises a crucial question: *how to repair the model (ultimately with provable guarantee) once biased behavior is identified?*

A typical method for DNN fairness repair is to retrain the model with a set of discriminatory inputs. While retraining is easy to deploy, it inherently faces challenges in real-world applications due to its high costs and the necessity of accessing the original training set, which may be impractical when the model is obtained from a third party or the training data are private. For more effective and efficient repair, researchers have proposed various methods that aim to adjust the parameters of a biased NN to eliminate discriminatory behaviors. Among them, a common paradigm inspired by traditional software programs debugging is to first identify the key units in the model that are responsible for discrimination. For example, CARE [16] constructs the causal model between neurons and model outputs to pinpoint problematic neurons, followed by using a heuristic algorithm to generate neuron-level patches to modify their parameters. In addition, some other approaches achieve repair through more efficient ways: RUNNER [17] design a loss function to iteratively optimize the identified biased neurons via gradient descent, while IDNN [18] directly isolate them. NeuFair [19], on the other hand, leverages the simulated annealing algorithm to provide statistical guarantees for group fairness repair. More recently, GRFT [9] has been proposed for more effective fairness testing and repair. It achieves state-of-the-art performance by directly minimizing the distance of model outputs between discriminatory inputs and their similar instances.

Research gap - While prior repair methods have shown effectiveness in certain cases, they suffer from two fundamental limitations. First, the repair techniques they employ (e.g., heuristic algorithms, gradient-based parameter tuning, or neuron isolation) are empirical in nature and lack deterministic guarantees. Even on the discriminatory pair $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$ used in repair, these techniques may not ensure that the model's outputs satisfy fairness constraints. Second, these approaches use a set of discriminatory input pairs $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$ to analyze and repair model behavior, where $\boldsymbol{x}'$ is chosen from $\mathcal{S}(\boldsymbol{x})$ (the set of all instances similar to $\boldsymbol{x}$). As such, they only observe and correct unfairness over a limited region of $\mathcal{S}(\boldsymbol{x})$. Consequently, the repaired model may still exhibit unfair behavior on unseen or unsampled inputs, resulting in limited generalization.

Our insight - To address the above challenges, our key idea is to leverage interval bound propagation (a widely used NN

[§] Corresponding author: Jingyi Wang.

2643-1572/25/$31.00 ©2025 IEEE
DOI 10.1109/ASE63991.2025.00049

verification technique) to soundly capture model outputs over $\mathcal{S}(\boldsymbol{x})$. The derived bounds can be utilized to guide fairness repair, which encourages the model to produce consistent outputs across all similar instances. To further provide guarantees for repair, our intuition is to exploit these bounds as a bridge to integrate fairness constraints and model modifications to construct a unified constraint-solving formulation. The solution to this problem can induce a repaired model with guaranteed fairness over $\mathcal{S}(\boldsymbol{x})$.

Our solution - Based on the above insight, we propose PROF, a novel provable NN fairness repair framework. As illustrated in Fig. 2, PROF consists of two core components. Given a DNN f (which can be sliced as the first $\mathrm{L}-1$ layers $f_{1:\mathrm{L}}$ and the last layer f_{L}), the first step applies interval bound propagation to calculate the concrete bounds that soundly capture the outputs of $f_{1:\mathrm{L}}$ (i.e., outputs in feature space). These bounds define axis-aligned hyperrectangles, which enables us to directly tighten them to mitigate feature differences. In the second step, a naive approach would be to use these concrete bounds to construct a constraint-solving problem, where the parameter changes of the final layer f_{L} are treated as optimization variables. However, since concrete bounds are often overly conservative, PROF synthesizes symbolic bounds to formulate a more precise problem, thereby avoiding excessive modifications. To make the new problem solvable by off-the-shelf solvers, we introduce the dual theorem to eliminate nonlinear operations while preserving soundness. Finally, PROF establishes a Mixed-Integer Linear Programming (MILP) problem, ensuring that the solution induces a repaired model that is provably fair over the given set $\mathcal{S}(\boldsymbol{x})$.

We have implemented PROF as a self-contained toolkit and evaluated it on four popular datasets involving various sensitive attributes. The results demonstrate its effectiveness in correcting the unfairness over the given set $\mathcal{S}(\boldsymbol{x})$ with provable guarantees, and its substantial improvements in generalization over existing state-of-the-art repair methods. On average, PROF achieves 95.93% and 93.16% relative fairness improvement on the full dataset and the full input space, where the best baseline achieves only 71.44% and 72.67%. For the setting with multiple sensitive attributes and more practical fairness definition, PROF also exhibits remarkable generalization, keeping around 90% fairness improvement. Additional experiments with four state-of-the-art fairness testing frameworks further confirm its consistent effectiveness.

To summarize, this paper makes the following contributions:

- We present PROF, a novel framework for provable neural network fairness repair with three key technical innovations:
  1) We leverage interval bound propagation to soundly capture the model outputs, and design a bounds tightening process to effectively mitigate biased behavior.
  2) We synthesize symbolic bounds to precisely encode fairness constraints and model modifications into a unified constraint-solving formulation with provable guarantees.
  3) We leverage the duality theorem to eliminate nonlinearities and construct an MILP tractable by existing solvers.
- We evaluate PROF on four widely adopted benchmark datasets and demonstrate that it significantly outperforms the state-of-the-art in terms of provable guarantees and generalization to unseen samples.
- We release the code, scripts, and supplementary materials for this paper at this repository to facilitate future studies.

II. PRELIMINARY

DNNs. In this work, we focus on DNN models for binary classification tasks. A DNN can be represented as a function $f:\mathbb{R}^m\to\mathbb{R}$, which maps a high-dimensional input $\boldsymbol{x}\in\mathbb{R}^m$ to an output $f(\boldsymbol{x})\in\mathbb{R}$. It typically consists of an input layer f_1, several hidden layers $\{f_2,\cdots,f_{\mathrm{L}-1}\}$, and an output layer f_{L}. The first $\mathrm{L}-1$ layers, denoted as $f_{1:\mathrm{L}}\overset{\text{def}}{=}f_{\mathrm{L}-1}\circ\cdots\circ f_1$, transform the input into a d-dimensional feature space, i.e., $f_{1:\mathrm{L}}(\boldsymbol{x})\in\mathbb{R}^d$, while the final layer f_{L} makes the classification. The model predicts the positive class if $f(\boldsymbol{x})\geq 0$ and the negative class otherwise. Given a training dataset $\mathcal{D}_{\text{train}}$, a DNN is trained by minimizing binary cross-entropy (BCE) loss ℓ:

$$\min_f \frac{1}{|\mathcal{D}_{\text{train}}|}\sum_{(\boldsymbol{x}^i,y^i)\in\mathcal{D}_{\text{train}}}\ell(\sigma(f(\boldsymbol{x}^i)),y^i)$$

where σ and $y^i\in\{0,1\}$ denote the sigmoid function and the true label for the input $\boldsymbol{x}^i$, respectively.

Interval Bound Propagation. Owing to its efficiency in computing NN output ranges, interval bound propagation is widely used in NN verification [20], [21], [22], [23], [24], [25]. The core idea involves propagating interval bounds layer-wise through the model, starting from predefined input domains. In contrast to exact verification methods [26], [27], [28] that rely on SAT solvers, interval arithmetic offers scalable approximations by symbolizing each neuron's value as an interval derived from its predecessor layer. Here we briefly describe how it propagates the bounds through the linear layer and the activation layer (using ReLU as an example). For a neuron h after a linear transformation with weights $\mathbf{W}$ and bias b, its output bounds are computed as:

$$\begin{aligned}\underline{\mathrm{h}}&=b+\sum_i\left(\max(0,\mathrm{W}_i)\cdot\underline{\mathrm{z}}_i+\min(0,\mathrm{W}_i)\cdot\overline{\mathrm{z}}_i\right),\\ \overline{\mathrm{h}}&=b+\sum_i\left(\max(0,\mathrm{W}_i)\cdot\overline{\mathrm{z}}_i+\min(0,\mathrm{W}_i)\cdot\underline{\mathrm{z}}_i\right)\end{aligned}\tag{1}$$

where $\underline{\mathrm{z}}_i,\overline{\mathrm{z}}_i$ are the bounds of the i-th neuron in the predecessor layer. For an unstable ReLU neuron $\mathrm{h}=\mathrm{ReLU}(\mathrm{z})$ with input interval $[l,u]$ where $l<0<u$, a sound interval propagation can be derived via linear relaxation in various ways [21], [29]. Here we show a simple triangular form:

$$0\leq\mathrm{h}\leq\frac{u}{u-l}\cdot(\mathrm{z}-l),\ \left[\underline{\mathrm{h}},\overline{\mathrm{h}}\right]=[0,u]\tag{2}$$

Example 1. *Fig. 1 illustrates interval bound propagation on a simple NN, where the brown equations denote the neurons' symbolic bounds and the black square brackets are concrete bounds. The input x_1 and x_2 are bounded by $[0,8]$ and $[-1,1]$. Using Eq. (1), the interval propagation for the first layer concretizes the symbolic expressions $x_3=x_1+6x_2$ and*

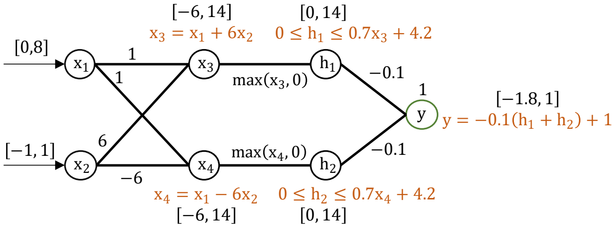

Fig. 1: Interval bound propagation on an example NN. All neurons have zero bias except the output neuron (bias=1).

$x_4 = x_1 - 6x_2$ to concrete bounds $[-6, 14]$ and $[-6, 14]$. Then the triangular relaxation from Eq. (2) *is used to calculate the bounds of h_1 and h_2, refining both to $[0, 14]$. The output y finally yields the bound $[-1.8, 1]$. Notably, this bound is usually conservative due to the input dependencies [30], [31] of interval arithmetic, which assumes that the propagation of each neuron is independent. For example, the minimum $y = -1.8$ implies $h_1 = h_2 = 14$, where no valid input (x_1, x_2) can simultaneously achieve $h_1 = 14$ and $h_2 = 14$. We analyze how this issue impacts the repair process and present our solution in Sec. III-C and Sec. III-D.*

Individual Fairness. As established in prior work [8], [12], [32], individual fairness (IF) requires a model to produce consistent predictions for similar individuals who differ only in sensitive attributes. Let $\boldsymbol{x} = (x_1, x_2, \cdots, x_m) \in \mathbb{R}^m$ be an input. We define $\mathcal{S}(\boldsymbol{x})$ as the set of all inputs similar to $\boldsymbol{x}$:

$$\mathcal{S}(\boldsymbol{x}) = \{\boldsymbol{x}' \mid \exists j \in \mathrm{P},\, x_j \neq x'_j;\, \forall j \in \mathrm{NP},\, x_j = x'_j\}$$

where x_j and x'_j denote the values of the j-th attribute in $\boldsymbol{x}$ and $\boldsymbol{x}'$, P is the set of sensitive/protected attribute indices, and NP represent the set of non-sensitive attribute indices. Recent work [13], [33] further relaxes this notion by allowing small perturbations on non-sensitive attributes. Specifically, a relaxed neighborhood $\mathcal{S}(\boldsymbol{x})$ is defined as:

$$\mathcal{S}(\boldsymbol{x}) = \{\boldsymbol{x}' \mid \exists j \in \mathrm{P},\, x_j \neq x'_j;\, \forall j \in \mathrm{NP},\, |x_j - x'_j| \leq \epsilon_j\}$$

where $\epsilon_j > 0$ limits allowable variations on non-sensitive attribute j, reflecting that small differences (e.g., age differences of a few years) may not violate the fairness relation.

Building on above notions, we define an input $\boldsymbol{x}$ as an Individual Discriminatory Instance (IDI) if there exists $\boldsymbol{x}' \in \mathcal{S}(\boldsymbol{x})$ such that: $(f(\boldsymbol{x}) \geq 0 \wedge f(\boldsymbol{x}') < 0) \vee (f(\boldsymbol{x}) < 0 \wedge f(\boldsymbol{x}') \geq 0)$. Such a pair $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$ is referred to as an IDI pair. Given a NN f, we further define $\mathcal{U}(f, \boldsymbol{x}, \mathcal{S}(\boldsymbol{x}))$ as the set of all inputs in $\mathcal{S}(\boldsymbol{x})$ that result in unfair classification under f:

$$\begin{aligned}\mathcal{U}(f, \boldsymbol{x}, \mathcal{S}(\boldsymbol{x})) = \{\boldsymbol{x}' \in \mathcal{S}(\boldsymbol{x}) \mid (f(\boldsymbol{x}) \geq 0 \wedge f(\boldsymbol{x}') < 0) \\ \vee (f(\boldsymbol{x}) < 0 \wedge f(\boldsymbol{x}') \geq 0)\}\end{aligned} \tag{3}$$

Example 2. *Let us revisit Fig. 1. Suppose we have an input $\boldsymbol{x} = (4, 0)$ with $f(\boldsymbol{x}) = 0.2$, where x_1 is the protected attribute ranging over $[0, 8]$, and x_2 is a non-sensitive attribute with $\epsilon = 1$. The neighborhood $\mathcal{S}(\boldsymbol{x})$ is thus defined as $\{\boldsymbol{x}' \mid 0 \leq x'_1 \leq 8,\ -1 \leq x'_2 \leq 1\}$. It is easy to verify that there exists $\boldsymbol{x}' \in \mathcal{S}(\boldsymbol{x})$ such that $f(\boldsymbol{x}') < 0$; for example, $\boldsymbol{x}' = (8, 1)$ yields $f(\boldsymbol{x}') = -0.6$. Therefore, $\boldsymbol{x}$ constitutes an IDI, and f violates individual fairness at this input.*

Problem Formulation. We now formalize our repair problem.

Definition 1 **(The IF repair problem).** *Given a DNN f and a repair set $\mathcal{D}_r = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^n\}$, where $\boldsymbol{x}^i$ denotes the i-th input. The goal of provable repair is to find a modified DNN $\tilde{f}$ that guarantees IF over $\mathcal{S}(\boldsymbol{x}^i)$ for every $\boldsymbol{x}^i \in \mathcal{D}_r$, that is:*

$$\forall \boldsymbol{x}^i \in \mathcal{D}_r,\ \forall \boldsymbol{x} \in \mathcal{S}(\boldsymbol{x}^i) : \mathcal{U}(\tilde{f}, \boldsymbol{x}, \mathcal{S}(\boldsymbol{x}^i)) = \emptyset \tag{4}$$

We also aim for the repair to exhibit generalization by mitigating unfairness on inputs beyond $\mathcal{D}_\mathrm{r}$. Consistent with prior repair work [18], [16], we assume access to a small set $\mathcal{D}_\mathrm{c}$ of training data to preserve the model's original performance.

III. Methodology

A. Overview

Fig. 2 presents ProF, our provable fairness repair framework for DNNs. It consists of two main components: (1) unfair feature extractor calibration via **concrete bounds tightening**, and (2) provable repair through **symbolic bounds synthesis and constraint-solving**. The first step operates in the feature space: by propagating interval bounds through the network, ProF establishes sound over-approximation for the features of all similar individuals. Then a progressively bounds tightening process is designed to rectify the biased feature extractor, promoting feature consistency on similar individuals. In the second step, symbolic bounds (which are tighter than concrete bounds) are synthesized and combined with the fairness constraints to formulate a more precise constraint-solving problem that provides guarantees for repair. To eliminate the nonlinear operations, we introduce the dual theorem and transform the original problem into an MILP, while preserving soundness.

B. Correcting Biased Feature Extractor

The objective of this phase is to calibrate the model's feature extraction part $f_{1:\mathrm{L}}$ so that it produces features as consistent as possible for all inputs in the similar set $\mathcal{S}$. The ideal scenario is that for any inputs $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}(\boldsymbol{x}^i)$, $f_{1:\mathrm{L}}$ produces identical features and thus f makes the same classification, i.e.,

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}(\boldsymbol{x}^i) : f_{1:\mathrm{L}}(\boldsymbol{x}) = f_{1:\mathrm{L}}(\boldsymbol{x}') \tag{5}$$

Some prior works [17], [18], [9] pursue similar goals by either: (1) biased neuron identification via differential analysis on given IDI pairs followed by parameter modification, or (2) retraining the models to minimize the distances between features extracted from the IDI pairs. In other words, they aim to reduce the distances between specific feature pairs (discrete point pairs in the purple region in Fig. 2, Step 1). However, these approaches rely on IDI pair $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$ generated either by sampling within the similar input set $\mathcal{S}(\boldsymbol{x})$ or existing testing methods [8], [10]. This sampling-driven nature inherently suffers from incomplete coverage of $\mathcal{S}(\boldsymbol{x})$, as finite samples cannot exhaustively represent the entire set. Consequently, some uncovered instances may still retain large feature discrepancies and thus lead to unfair classifications.

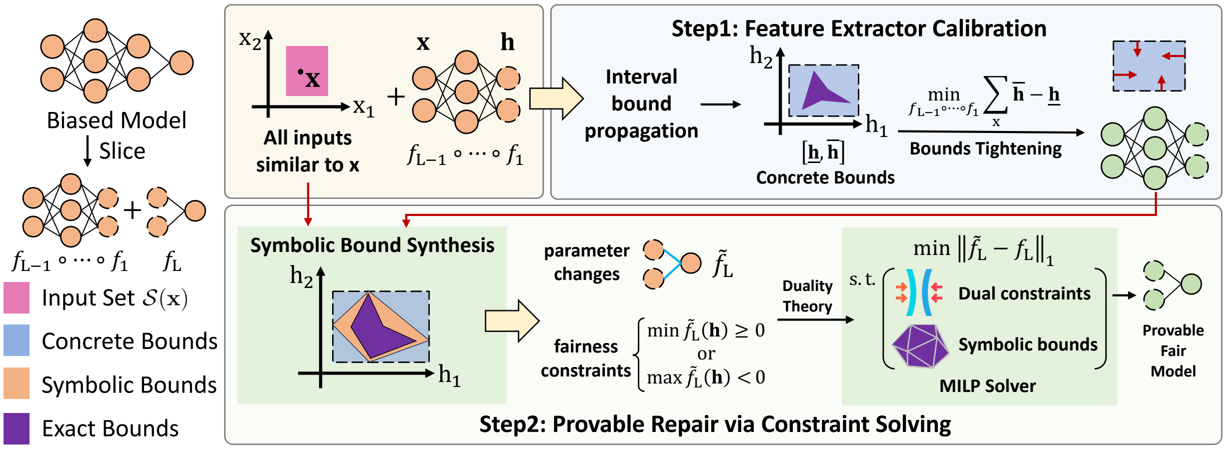
Biased Model
Slice
$f_{L-1} \circ \cdots \circ f_1$
f_L
🟪 Input Set $\mathcal{S}(x)$
🟦 Concrete Bounds
🟧 Symbolic Bounds
🟪 Exact Bounds
x_2
x
x_1
+
All inputs similar to x
x
h
$f_{L-1} \circ \cdots \circ f_1$
Step1: Feature Extractor Calibration
Interval bound propagation
h_2
h_1
$[\underline{h}, \overline{h}]$
Concrete Bounds
$\min_{f_{L-1} \circ \cdots \circ f_1} \sum_x \overline{h} - \underline{h}$
Bounds Tightening
Symbolic Bound Synthesis
h_2
h_1
parameter changes
$\tilde{f}_L$
fairness constraints
$\min \tilde{f}_L(\mathbf{h}) \geq 0$
or
$\max \tilde{f}_L(\mathbf{h}) < 0$
Duality Theory
$\min \|\tilde{f}_L - f_L\|_1$
s. t.
Dual constraints
Symbolic bounds
MILP Solver
Provable Fair Model
Step2: Provable Repair via Constraint Solving

Fig. 2: The overview of PROF framework. For notational simplicity, we use $f_{1:L}$ to denote $f_{L-1} \circ \cdots \circ f_1$ hereafter.

```
Algorithm 1: Biased Feature Calibration
  Input: Biased NN $f = f_L \circ f_{1:L}$, repair set $\mathcal{D}_r$, a small set
         of training data $\mathcal{D}_c$, maximum iteration T
  Output: Repaired feature extractor $\tilde{f}_{1:L}$
1 for $i \leftarrow 1$ to $|\mathcal{D}_r|$ do
2     $\mathcal{S}(\boldsymbol{x}^i) \overset{\text{def}}{=}$ Set of all inputs similar to $\boldsymbol{x}^i$
3     $\left[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i\right] \leftarrow \textsc{ConcreteBounds}(f_{1:L}, \mathcal{S}(\boldsymbol{x}^i))$
4     $\text{OriDiff}_i \leftarrow \left\|\overline{\mathbf{h}}^i - \underline{\mathbf{h}}^i\right\|_1$
5 for $iter \leftarrow 1$ to T do
6     for $i \leftarrow 1$ to $|\mathcal{D}_r|$ do
7         $\left[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i\right] \leftarrow \textsc{ConcreteBounds}(f_{1:L}, \mathcal{S}(\boldsymbol{x}^i))$
              /* Update the concrete bounds */
8     $\mathcal{L}_{\text{fair}} \leftarrow \frac{1}{|\mathcal{D}_r|} \sum_{i=1}^{|\mathcal{D}_r|} \frac{\left\|\overline{\mathbf{h}}^i - \underline{\mathbf{h}}^i\right\|_1}{\text{OriDiff}_i}$
9     $\mathcal{L}_{\text{bce}} \leftarrow \frac{1}{|\mathcal{D}_c|} \sum_{(\boldsymbol{x},y) \in \mathcal{D}_c} \ell(\sigma(f(\boldsymbol{x})), y)$
10    $f_{1:L} \leftarrow \textsc{GradientDescent}(f_{1:L}; \mathcal{L}_{\text{fair}} + \mathcal{L}_{\text{bce}})$
11 return $f_{1:L}$
```

To address this problem, we propose utilizing interval bound propagation to soundly *characterize* and *mitigate* the feature differences across the neighborhood set $\mathcal{S}(\boldsymbol{x})$. Unlike discrete sampling methods, our approach over-approximates the model's feature-space outputs. Specifically, we employ auto-LiRPA [34] to calculate[1] the concrete bounds $[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i] \subset \mathbb{R}^d$ for each set $\mathcal{S}(\boldsymbol{x}^i)$, such that:

$$\forall i \in [n],\ \boldsymbol{x} \in \mathcal{S}(\boldsymbol{x}^i):\ f_{1:L}(\boldsymbol{x}) \in \left[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i\right] \tag{6}$$

where $[n]$ denotes the set $\{1, 2, \ldots, n\}$. As shown in Fig. 2 (Step 1, blue region), the derived interval $[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i]$ forms an axis-aligned hyperrectangle that provably encloses the exact feature set $\mathcal{H}^i \overset{\text{def}}{=} \{f_{1:L}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{S}(\boldsymbol{x}^i)\}$ (purple region). The axis-aligned property enables us to upper bound the maximum distance L_{dis} between features in $\mathcal{H}^i$:

$$\begin{aligned} L_{dis} &\overset{\text{def}}{=} \max_{\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}(\boldsymbol{x}^i)} \|f_{1:L}(\boldsymbol{x}) - f_{1:L}(\boldsymbol{x}')\|_1 \\ &= \max_{\mathbf{h}, \mathbf{h}' \in \mathcal{H}^i} \|\mathbf{h} - \mathbf{h}'\|_1 \leq \sum_{j=1}^{d} \max_{\mathbf{h}, \mathbf{h}' \in \mathcal{H}^i} |\mathbf{h}_j - \mathbf{h}'_j| \\ &\leq \|\overline{\mathbf{h}}^i - \underline{\mathbf{h}}^i\|_1 \end{aligned} \tag{7}$$

With the derived upper bounds in hand, we design a progressive bounds tightening process to mitigate the bias in $f_{1:L}$. This procedure iteratively reduces the ℓ_1-norm of the concrete bounds, thereby contracting the maximum feature distance. As detailed in Alg. 1 (Lines 2–4), the process begins by computing initial concrete bounds for each input $\boldsymbol{x}^i \in \mathcal{D}_r$ over its neighborhood set $\mathcal{S}(\boldsymbol{x}^i)$. It then updates the concrete bounds and minimizes a normalized objective, which measures the relative ℓ_1-norm reduction between current and original bounds. This ratio, bounded within $[0, 1]$, directly quantifies the degree of bias mitigation, where zero corresponds to the ideal scenario described in Eq. (5). We aggregate these ratios across all inputs in $\mathcal{D}_r$ to construct the fairness loss $\mathcal{L}_{\text{fair}}$ (Line 8 of Alg. 1). To preserve the model's performance, the BCE loss $\mathcal{L}_{\text{bce}}$ is computed on a small set of training data $\mathcal{D}_c$. We optimize the model parameters to minimize the combined objective $\mathcal{L} = \mathcal{L}_{\text{fair}} + \mathcal{L}_{\text{bce}}$ through gradient descent.

Remark: While the above process can effectively mitigate bias in the feature extractor (see Section IV-E), the fixed-number gradient descent steps do not guarantee provable repair. The repaired feature extractor may still yield $\mathcal{L}_{\text{fair}} > 0$, indicating potential feature discrepancies within some $\mathcal{S}(\boldsymbol{x}^i)$ that could lead to unfair classifications. To address this problem, we proceed to repair the final layer f_L in the following sections. We first present a naive method that directly utilizes the concrete bounds, followed by our proposed advanced approach that incorporates symbolic bounds for enhanced precision.

[1] There are several other interval propagation tools, such as DeepPoly [20] and DeepZ [35], which vary in precision. We choose auto-LiRPA [34] due to its balanced trade-off between efficiency and precision.

C. A Naive Provable Repair Method with Concrete Bounds

In this section, we present a naive repair method for the final classification layer f_{L}. Our core idea is to formulate the repair as a constraint-solving problem and leverage existing solvers to obtain the solution with provable guarantees. Formally, the fairness repair on f_{L} is formulated as follows:

$$\min_{\Delta\mathbf{W},\Delta\mathrm{b}} \|\Delta\mathbf{W}\|_1 + \|\Delta\mathrm{b}\|_1 \tag{8}$$

$$\text{s.t. } \forall i \in [n],\ \mathcal{H}^i = \{\tilde{f}_{1:\mathrm{L}}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{S}(\boldsymbol{x}^i)\} \tag{9}$$

$$\forall i \in [n],\ \min_{\mathbf{h}\in\mathcal{H}^i} \tilde{f}_{\mathrm{L}}(\mathbf{h}) \geq 0 \vee \max_{\mathbf{h}\in\mathcal{H}^i} \tilde{f}_{\mathrm{L}}(\mathbf{h}) < 0 \tag{10}$$

Here, $\tilde{f}_{1:\mathrm{L}}$ denotes the feature extractor calibrated by Alg. 1 and $\tilde{f}_{\mathrm{L}}(\mathbf{h}) = (\mathbf{W}+\Delta\mathbf{W})\mathbf{h}+\mathrm{b}+\Delta\mathrm{b}$ is the repaired final layer, where $\mathbf{W} + \Delta\mathbf{W} \in \mathbb{R}^{1\times d}$ and $\mathbf{h} \in \mathbb{R}^d$. Eq. (8) minimizes parameter modifications while Eq. (10) and Eq. (9) enforce fairness through output consistency across all inputs in $\mathcal{S}(\boldsymbol{x}^i)$.

We identify that existing solvers cannot directly handle the formulated problem due to two inherent complexities: (1) $\tilde{f}_{1:\mathrm{L}}$ includes multiple nonlinear layers and thus the exact feature set $\mathcal{H}^i$ is highly non-convex, and (2) the constraint in Eq. (10) introduces multiplication terms $\Delta\mathbf{W}\mathbf{h}$, which are non-linear. To address the former, a straightforward method is to use the concrete bounds again to over-approximate $\mathcal{H}^i$. Specifically, we simplify Eqs. (9) and (10) as follows:

$$\begin{aligned} \forall i \in [n],\ &\mathrm{b} + \Delta\mathrm{b} + \min_{\mathbf{h}\in\left[\underline{\mathbf{h}}^i,\overline{\mathbf{h}}^i\right]} (\mathbf{W}+\Delta\mathbf{W})\mathbf{h} \geq 0 \\ &\vee\ \mathrm{b} + \Delta\mathrm{b} + \max_{\mathbf{h}\in\left[\underline{\mathbf{h}}^i,\overline{\mathbf{h}}^i\right]} (\mathbf{W}+\Delta\mathbf{W})\mathbf{h} < 0 \end{aligned} \tag{11}$$

Since $[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i]$ over-approximates $\mathcal{H}^i$, Eq.(11) consists of strengthened constraints that imply Eqs.(10) and (9). Moreover, the above simplification enables us to introduce two sets of variables $\{\mathrm{P}_{i,j}\}_{j\in[d]}$ and $\{\mathrm{Q}_{i,j}\}_{j\in[d]}$ to capture the extreme values of $(\mathbf{W} + \Delta\mathbf{W})\mathbf{h}$ in each dimension:

$$\begin{aligned} \mathrm{P}_{i,j} \leq (\mathbf{W}_j + \Delta\mathbf{W}_j)\underline{\mathbf{h}}_j^i \wedge \mathrm{P}_{i,j} \leq (\mathbf{W}_j + \Delta\mathbf{W}_j)\overline{\mathbf{h}}_j^i \\ \mathrm{Q}_{i,j} \geq (\mathbf{W}_j + \Delta\mathbf{W}_j)\underline{\mathbf{h}}_j^i \wedge \mathrm{Q}_{i,j} \geq (\mathbf{W}_j + \Delta\mathbf{W}_j)\overline{\mathbf{h}}_j^i \end{aligned} \tag{12}$$

where $\underline{\mathbf{h}}_j^i$ and $\overline{\mathbf{h}}_j^i$ are the j-th dimensional endpoints of the hyperrectangle $[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i]$. Therefore, the summations over j of $\mathrm{P}_{i,j}$ and $\mathrm{Q}_{i,j}$ can provide lower and upper bounds for $(\mathbf{W} + \Delta\mathbf{W})\mathbf{h}$ across the whole hyperrectangle $[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i]$ while avoiding the multiplication between $\mathbf{h}$ and $\Delta\mathbf{W}$, i.e.,

$$\begin{aligned} \sum_{j\in[d]} \mathrm{P}_{i,j} &\leq \min_{\mathbf{h}\in\left[\underline{\mathbf{h}}^i,\overline{\mathbf{h}}^i\right]} (\mathbf{W}+\Delta\mathbf{W})\mathbf{h} \\ \sum_{j\in[d]} \mathrm{Q}_{i,j} &\geq \max_{\mathbf{h}\in\left[\underline{\mathbf{h}}^i,\overline{\mathbf{h}}^i\right]} (\mathbf{W}+\Delta\mathbf{W})\mathbf{h} \end{aligned} \tag{13}$$

Let $\mathrm{LB}_i \overset{\text{def}}{=} \mathrm{b} + \Delta\mathrm{b} + \sum_{j\in[d]} \mathrm{P}_{i,j}$ and $\mathrm{UB}_i \overset{\text{def}}{=} \mathrm{b} + \Delta\mathrm{b} + \sum_{j\in[d]} \mathrm{Q}_{i,j}$, we then transform the repair problem as:

$$\begin{aligned} &\min_{\Delta\mathbf{W},\Delta\mathrm{b}} \|\Delta\mathbf{W}\|_1 + \|\Delta\mathrm{b}\|_1 \\ &\text{s.t. } \forall i \in [n]\ (12),\ \mathrm{LB}_i \geq 0 \vee \mathrm{UB}_i < 0 \end{aligned} \tag{14}$$

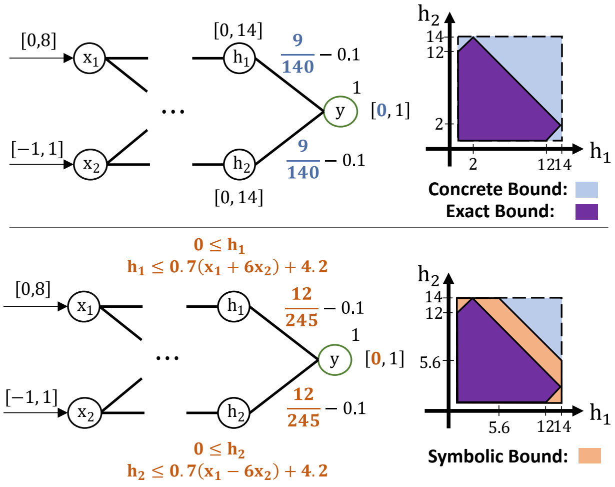

Fig. 3: Comparison of repair with concrete bounds and symbolic bounds. Purple region shows the true feature set $\mathcal{H}^i$.

Note that the disjunction ($\vee$) can be handled using Big-M method (detailed in the next section). The upper half of Fig. 3 revisits Example 1, where the blue values indicate the parameter modifications from solving Problem (14), which enforces $\forall \mathrm{h}_1 \in [0, 14], \mathrm{h}_2 \in [0, 14] : (\frac{9}{140} - 0.1) \times \mathrm{h}_1 + (\frac{9}{140} - 0.1) \times \mathrm{h}_2 + 1 \geq 0$ to ensure fairness. The provable guarantees achieved by solving Problem (14) are formally stated as:

Theorem 1. *Let $\mathcal{D}_r = \{\boldsymbol{x}^i\}_{i=1}^n$ be a set of inputs, each associated with a similarity neighborhood $\mathcal{S}(\boldsymbol{x}^i)$, and let $f = f_{\mathrm{L}} \circ \tilde{f}_{1:\mathrm{L}}$ be a DNN. Then any feasible solution $(\Delta\mathbf{W}, \Delta\mathrm{b})$ to the problem (14) induces a repaired final layer $\tilde{f}_{\mathrm{L}}$ such that the composite model $\tilde{f} = \tilde{f}_{\mathrm{L}} \circ \tilde{f}_{1:\mathrm{L}}$ provably satisfies individual fairness on all $\mathcal{S}(\boldsymbol{x}^i)$, $i \in [n]$.*

Proof. We refer the reader to the supplementary material. □

D. Provable Repair with Symbolic Bounds

We now identify a crucial problem of the proposed naive method: the concrete bound $[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i]$ is typically too conservative to capture the exact feature set $\mathcal{H}^i$ (purple region in figure). This leads to unnecessary modifications of the final layer in order to enforce fairness across the entire box $[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i]$ rather than just the exact set $\mathcal{H}^i$. As illustrated in the upper half of Fig. 3, such conservativeness results in excessive changes that account for spurious points outside $\mathcal{H}^i$. This issue is exacerbated when the repair set $\mathcal{D}_{\mathrm{r}}$ contains multiple inputs, requiring simultaneous satisfaction of fairness constraints over multiple sets $\mathcal{S}(\boldsymbol{x})$ (see Section IV-E).

As discussed in Example 1, the imprecision of concrete bounds fundamentally stems from their assumption of inter-neuron independence during propagation, which discards crucial variable dependencies. To capture $\mathcal{H}^i$ more precisely, we propose synthesizing *symbolic bounds*, where each neuron's bounds are characterized as a linear combination of preceding variables, thereby preserving the dependencies. Given a set

$\mathcal{S}(\boldsymbol{x}^i)$, we first utilize auto-LiRPA (other bound propagation tools are also applicable) to calculate the linear bounds between $\mathbf{h}$ and $\boldsymbol{x}$, and further incorporate the input bounds to synthesize the symbolic bounds for repair as:

$$\widehat{\mathcal{H}}^i \overset{\text{def}}{=} \Big\{\mathbf{h} \,\Big|\, \boldsymbol{\alpha}_i^{\leq}\boldsymbol{x} + \boldsymbol{\beta}_i^{\leq} \leq \mathbf{h} \leq \boldsymbol{\alpha}_i^{\geq}\boldsymbol{x} + \boldsymbol{\beta}_i^{\geq}, \\ \mathcal{S}_L(\boldsymbol{x}^i) \leq \boldsymbol{x} \leq \mathcal{S}_U(\boldsymbol{x}^i)\Big\} \tag{15}$$

where $\boldsymbol{\alpha}_i^{\leq}$, $\boldsymbol{\beta}_i^{\leq}$, $\boldsymbol{\alpha}_i^{\geq}$ and $\boldsymbol{\beta}_i^{\geq}$ are linear coefficients and biases, $\mathcal{S}_L(\boldsymbol{x}^i)$ and $\mathcal{S}_U(\boldsymbol{x}^i)$ denote the lower bound and upper bound of $\mathcal{S}(\boldsymbol{x}^i)$, respectively. The brown equations in Fig. 3 denote the symbolic bounds for our running example:

$$\widehat{\mathcal{H}}^i = \{\mathbf{h} \,|\, 0 \leq \mathrm{h}_1 \leq 0.7(x_1 + 6x_2) + 4.2, 0 \leq x_1 \leq 8 \\ 0 \leq \mathrm{h}_2 \leq 0.7(x_1 - 6x_2) + 4.2, -1 \leq x_2 \leq 1\}$$

We can see that the region formed by $\widehat{\mathcal{H}}^i$ (the brown area) is more precise than the one formed by concrete bounds.

Once the symbolic bounds is obtained, the repair can be formulated to guarantee that fairness constraints hold over $\widehat{\mathcal{H}}^i$:

$$\min_{\Delta\mathbf{W},\Delta\mathrm{b}} \|\Delta\mathbf{W}\|_1 + \|\Delta\mathrm{b}\|_1 \\ \text{s.t. } \min_{\mathbf{h}\in\widehat{\mathcal{H}}^i} \tilde{f}_{\mathrm{L}}(\mathbf{h}) \geq 0 \vee \max_{\mathbf{h}\in\widehat{\mathcal{H}}^i} \tilde{f}_{\mathrm{L}}(\mathbf{h}) < 0 \tag{16}$$

However, while symbolic bounds provide tighter approximations, they consist of a set of general linear constraints on $\boldsymbol{x}$ and $\mathbf{h}$, forming a polytope with fundamentally different geometric structure from hyperrectangles. Note that one of the core challenge of provable repair stems from the non-linear term $\Delta\mathbf{W}\mathbf{h}$ in fairness constraint $\min_{\mathbf{h}}(\mathbf{W}+\Delta\mathbf{W})\mathbf{h} \geq 0 \vee \max_{\mathbf{h}}(\mathbf{W}+\Delta\mathbf{W})\mathbf{h} < 0$, where $\mathbf{h} \in [\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i]$ or $\widehat{\mathcal{H}}^i$. Although Section III-C handled this issue by exploiting the axis-aligned nature of $[\underline{\mathbf{h}}^i, \overline{\mathbf{h}}^i]$, the polytope $\widehat{\mathcal{H}}^i$ lacks this nature and renders the endpoint analysis in Eq. (12) inapplicable.

We leverage Lagrangian duality to address this problem. In the following, we take $\min_{\mathbf{h}\in\widehat{\mathcal{H}}^i} \tilde{f}_{\mathrm{L}}(\mathbf{h}) \geq 0$ as an example to illustrate. The key idea is to recast the minimization over $\mathbf{h}$ as a maximization over dual variables $\boldsymbol{\lambda}$, thereby decoupling the multiplication between $\Delta\mathbf{W}$ and $\mathbf{h}$ while preserving soundness. Specifically, we first rewrite $\min_{\mathbf{h}\in\widehat{\mathcal{H}}_i}(\mathbf{W}+\Delta\mathbf{W})\mathbf{h}$ as follows:

$$\min_{\mathbf{p}} \mathbf{C}^\top\mathbf{p} \quad \text{s.t. } \mathbf{A}_i\mathbf{p} \leq \mathbf{D}_i \tag{17}$$

where $\mathbf{C} = \big[\mathbf{W} + \Delta\mathbf{W}, \mathbf{0}^{1\times m}\big]^\top \in \mathbb{R}^{(m+d)\times 1}, \mathbf{p} = \begin{pmatrix}\mathbf{h}\\ \boldsymbol{x}\end{pmatrix} \in \mathbb{R}^{m+d}$. $\mathbf{A}_i \in \mathbb{R}^{(2m+2d)\times(m+d)}$ and $\mathbf{D}_i \in \mathbb{R}^{2m+2d}$ are constant matrices that encode the definition of $\widehat{\mathcal{H}}^i$, constructed as:

$$\mathbf{A}_i = \begin{bmatrix} -\mathbf{I}_d & \boldsymbol{\alpha}_i^{\leq} \\ \mathbf{I}_d & -\boldsymbol{\alpha}_i^{\geq} \\ \mathbf{0} & -\mathbf{I}_m \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix}, \mathbf{D}_i = \begin{pmatrix} -\boldsymbol{\beta}_i^{\leq} \\ \boldsymbol{\beta}_i^{\geq} \\ -\mathcal{S}_L(\boldsymbol{x}^i) \\ \mathcal{S}_U(\boldsymbol{x}^i) \end{pmatrix}$$

It is evident that $\mathbf{A}_i\mathbf{p} \leq \mathbf{D}_i$ is equivalent to the definition of $\widehat{\mathcal{H}}^i$ given in Eq. (15).

Recalling Eq. (17), we observe that the variable $\Delta\mathbf{W}$ in $\mathbf{C}$ can be temporarily treated as a constant, since the feasible set $\{\mathbf{p} \mid \mathbf{A}_i\mathbf{p} \leq \mathbf{D}_i\}$ is independent of $\Delta\mathbf{W}$. In other words, problem (16) can be viewed as first solving the worst case $\mathbf{p}^*$ under a given $\Delta\mathbf{W}$, then optimizing $\Delta\mathbf{W}$ accordingly. Under this observation, Eq. (17) becomes a standard linear programming (LP). We define the Lagrangian function L and the dual function g to construct the dual problem of this LP:

$$L(\mathbf{p}, \boldsymbol{\lambda}_i) = \mathbf{C}^\top\mathbf{p} + \boldsymbol{\lambda}_i^\top(\mathbf{A}_i\mathbf{p} - \mathbf{D}_i),\ \boldsymbol{\lambda}_i \geq 0 \\ g(\boldsymbol{\lambda}_i) = \inf_{\mathbf{p}} L(\mathbf{p}, \boldsymbol{\lambda}_i) \tag{18}$$

where $\boldsymbol{\lambda}_i \in \mathbb{R}^{2m+2d}$ are dual variables. By maximizing the dual function, we obtain the optimality condition $\frac{\partial L}{\partial \mathbf{p}} = \mathbf{C} + \mathbf{A}_i^\top\boldsymbol{\lambda}_i = 0 \Rightarrow \mathbf{A}_i^\top\boldsymbol{\lambda}_i = -\mathbf{C}$ and construct the dual problem:

$$\begin{array}{ll} \text{(Primal Problem)} & \text{(Dual Problem)} \\ \min\limits_{\mathbf{p}} \quad \mathbf{C}^\top\mathbf{p} & \max\limits_{\boldsymbol{\lambda}_i\geq 0} \quad -\boldsymbol{\lambda}_i^\top\mathbf{D}_i \\ \text{s.t.} \quad \mathbf{A}_i\mathbf{p} \leq \mathbf{D}_i & \text{s.t.} \quad \mathbf{A}_i^\top\boldsymbol{\lambda}_i = -\mathbf{C} \end{array} \tag{19}$$

In the dual problem, the non-linear term $\Delta\mathbf{W}\mathbf{h}$ is decoupled, as only $\boldsymbol{\lambda}_i$ are variables and both $\mathbf{D}_i, \mathbf{A}_i$ are constants. We further note that the primal and dual problems attain the same optimal value due to the strong duality [36] for LP:

$$\min_{\mathbf{A}_i\mathbf{p}\leq\mathbf{D}_i} \mathbf{C}^\top\mathbf{p} = \max_{\substack{\mathbf{A}_i^\top\boldsymbol{\lambda}_i=-\mathbf{C} \\ \boldsymbol{\lambda}_i\geq 0}} -\boldsymbol{\lambda}_i^\top\mathbf{D}_i \tag{20}$$

Similar to the previous section, we now define the variable $\widehat{\mathrm{LB}}_i$ to soundly lower bound the model's output over $\widehat{\mathcal{H}}_i$:

$$\begin{aligned} \widehat{\mathrm{LB}}_i &= \mathrm{b} + \Delta\mathrm{b} + (-\boldsymbol{\lambda}_i^\top\mathbf{D}_i) \\ &\leq \mathrm{b} + \Delta\mathrm{b} + \max_{\boldsymbol{\lambda}_i} -\boldsymbol{\lambda}_i^\top\mathbf{D}_i \quad \text{s.t.} \quad \begin{array}{l} \mathbf{A}_i^\top\boldsymbol{\lambda}_i = -\mathbf{C} \\ \boldsymbol{\lambda}_i \geq 0 \end{array} \\ &= \mathrm{b} + \Delta\mathrm{b} + \min_{\mathbf{A}_i\mathbf{p}\leq\mathbf{D}_i} \mathbf{C}^\top\mathbf{p} \end{aligned} \tag{21}$$

Note that the upper bound (for $\mathrm{b}+\Delta\mathrm{b}+\max_{\mathbf{A}_i\mathbf{p}\leq\mathbf{D}_i}\mathbf{C}^\top\mathbf{p}$) can similarly be derived using another set of dual variables $\boldsymbol{\eta}_i \in \mathbb{R}^{2m+2d}$. We can now recast the problem (16) as following MILP:

$$\min_{\Delta\mathbf{W},\Delta\mathrm{b},\mathbf{Z},\boldsymbol{\lambda},\boldsymbol{\eta}} \|\Delta\mathbf{W}\|_1 + \|\Delta\mathrm{b}\|_1 \\ \text{s.t. } \mathbf{Z} \in \{0,1\}^n,\ \mathbf{C} = \Big[\mathbf{W} + \Delta\mathbf{W}, \mathbf{0}^{1\times m}\Big]^\top \\ \forall i \in [n] \begin{cases} \mathbf{A}_i = \begin{bmatrix} -\mathbf{I}_d & \boldsymbol{\alpha}_i^{\leq} \\ \mathbf{I}_d & -\boldsymbol{\alpha}_i^{\geq} \\ \mathbf{0} & -\mathbf{I}_m \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix}, \mathbf{D}_i = \begin{pmatrix} -\boldsymbol{\beta}_i^{\leq} \\ \boldsymbol{\beta}_i^{\geq} \\ -\mathcal{S}_L(\boldsymbol{x}^i) \\ \mathcal{S}_U(\boldsymbol{x}^i) \end{pmatrix} \\ \mathbf{A}_i^\top\boldsymbol{\lambda}_i = -\mathbf{C},\ \boldsymbol{\lambda}_i \geq 0 \\ \mathbf{A}_i^\top\boldsymbol{\eta}_i = \mathbf{C},\ \boldsymbol{\eta}_i \geq 0 \\ \widehat{\mathrm{LB}}_i = \mathrm{b} + \Delta\mathrm{b} + (-\boldsymbol{\lambda}_i^\top\mathbf{D}_i) \\ \widehat{\mathrm{UB}}_i = \mathrm{b} + \Delta\mathrm{b} + (\boldsymbol{\eta}_i^\top\mathbf{D}_i) \\ \widehat{\mathrm{LB}}_i \geq -\mathrm{M}\cdot(1 - \mathrm{Z}_i) \wedge \widehat{\mathrm{UB}}_i < \mathrm{M}\cdot\mathrm{Z}_i \end{cases} \tag{22}$$

In the last row we employ the Big-M method to convert the disjunction operation to conjunction, where M is a sufficiently large constant. This MILP formulation eliminates the non-linear term $\Delta\mathbf{W}\mathbf{h}$ while maintains soundness through strong duality. The provable repair guarantees achieved by solving

Problem (22), along with its superiority over Problem (14) in terms of objective value, are formally stated as follows:

Theorem 2. *Let $\mathcal{D}_r = \{\boldsymbol{x}^i\}_{i=1}^n$ be a set of inputs, each associated with a similarity neighborhood $\mathcal{S}(\boldsymbol{x}^i)$, and let $f = f_{\text{L}} \circ \tilde{f}_{1:\text{L}}$ be a DNN. Then any feasible solution $(\Delta\mathbf{W}, \Delta\text{b}, \mathbf{Z}, \boldsymbol{\lambda}, \boldsymbol{\eta})$ to the problem (22) induces a repaired final layer $\tilde{f}_{\text{L}}$ such that the composite model $\tilde{f} = \tilde{f}_{\text{L}} \circ \tilde{f}_{1:\text{L}}$ provably satisfies individual fairness on all $\mathcal{S}(\boldsymbol{x}^i)$, $i \in [n]$. Moreover, the optimal solution of problem (22) is guaranteed to be no worse than that of problem (14) (i.e., it achieves an objective value that is no larger).*

Proof. We refer the reader to the supplementary material. □

The overall algorithm of PROF is shown in Algorithm 2. It first calibrates the feature extractor $f_{1:\text{L}}$, then synthesizes symbolic bounds to formulate an MILP problem, which is solved by Gurobi [37] to achieve provable repair.

Remark: While our illustration focused on binary classification, we can naturally extend PROF to multi-class tasks by adapting the second step. For each input $\boldsymbol{x}^i$, instead of computing scalar lower/upper bounds for a single output neuron, we compute vectorized bounds ($\widehat{\text{LB}}_i, \widehat{\text{UB}}_i \in \mathbb{R}^N$) for all N classes. We then vectorize the integer variables Z_i in a one-hot manner to represent the multi-class decision, i.e., $\text{Z}_i \in \{0,1\}^N, \sum_j \text{Z}_{i,j} = 1$. These modifications enable us to soundly formulate the MILP for multi-class task.

Algorithm 2: PROF

```
Input: Biased NN $f = f_{\text{L}} \circ f_{1:\text{L}}$, repair set $\mathcal{D}_{\text{r}}$, a small set
       of training data $\mathcal{D}_{\text{c}}$, maximum iteration T
Output: Repaired model $\tilde{f}$
$\tilde{f}_{1:\text{L}} \leftarrow$ BIASEDFEATURECALIBRATION($f, \mathcal{D}_{\text{r}}, \mathcal{D}_{\text{c}}$, T)
                                   /* Repair $f_{1:\text{L}}$ by Alg. 1 */
for $i \leftarrow 1$ to $|\mathcal{D}_r|$ do
    $\mathcal{S}(\boldsymbol{x}^i) \overset{\text{def}}{=}$ Set of all inputs similar to $\boldsymbol{x}^i$
    $\widehat{\mathcal{H}}_i \leftarrow$ SYMBOLICBOUND($\tilde{f}_{1:\text{L}}, \mathcal{S}(\boldsymbol{x}^i)$)
Construct the MILP problem $\mathcal{M}$ according to Eq. (22)
$\Delta\mathbf{W}, \Delta\text{b}, \mathbf{Z}, \boldsymbol{\lambda}, \boldsymbol{\eta} \leftarrow$ Solve($\mathcal{M}$)
$\tilde{f}_{\text{L}} \leftarrow$ Linear($\mathbf{W} + \Delta\mathbf{W}, b + \Delta\text{b}$)
return $\tilde{f}_{\text{L}} \circ \tilde{f}_{1:\text{L}}$
```

IV. EVALUATION

In this section, we evaluate PROF through extensive experiments and answer the following research questions.

- **RQ1:** How effective is PROF in repairing NN unfairness?
- **RQ2:** Can PROF handle multiple protected attributes and more practical fairness definition?
- **RQ3:** How effective is PROF compared to baselines when evaluated by existing fairness testing frameworks?
- **RQ4:** How do the two key steps of PROF contribute to the repair performance?

A. Experimental Setup

1) Datasets and Models: We conduct experiments on four datasets including Adult (Census Income) [38], German Credit [39], Bank Marketing [40] and Compas [41], which are commonly used in fairness testing [8], [11], [10], [12] and repair [16], [9]. ❶ Adult is a dataset to predict whether a person earns more than $50 000. The protected attributes are gender, race, and age. ❷ Bank is a dataset for predicting if the bank client will subscribe a term deposit, with age being the sensitive attribute. ❸ German Credit is a dataset to assess creditworthiness based on personal and financial records. It has two protected attributes: gender and age. ❹ Compas is a dataset for assessing the likelihood of a criminal defendant reoffending. The protected attributes are gender, race, and age.

The biased NNs for repair are sourced from previous fairness verification works Fairify [13] and FairQuant [14]. These models consist of various number of layers and neurons.

2) Repair Setup: The original datasets $\mathcal{D}_{\text{full}}$ are divided into the training set $\mathcal{D}_{\text{train}}$ and the test set $\mathcal{D}_{\text{test}}$. Then we randomly select 100 instances from $\mathcal{D}_{\text{train}}$ to construct the repair set $\mathcal{D}_{\text{r}}$. We also provide a small set $\mathcal{D}_{\text{c}} \subset \mathcal{D}_{\text{train}}$ for all methods to preserve model's performance. Table I shows the size of these sets and the original accuracy of the DNN for each dataset.

TABLE I: Details of datasets involved in repair and evaluation.

Dataset	$\mathcal{D}_{\text{full}}$	$\mathcal{D}_{\text{train}}$	$\mathcal{D}_{\text{r}}$	$\mathcal{D}_{\text{c}}$	$\mathcal{D}_{\text{test}}$	Original Acc
Adult	48 842	38 438	100	500	6 784	85.24%
Compas	6 172	4 937	100	500	1 235	73.85%
German	1 000	700	100	100	300	72.67%
Bank	45 211	36 168	100	500	9 043	89.53%

3) Baselines: We select the following NN individual fairness repair methods as baselines for comprehensive comparison: ❶ **Flipping-based fine-tuning**. This is a basic method for flipping the protected attributes (e.g., changing the gender from female to male) in each input while maintaining the ground truth labels. Through this learning paradigm, the biased model learns to make predictions that are insensitive to variations in protected attributes. ❷ **CARE** [16]. This method utilizes the causal model to pinpoint key neurons that are responsible for unfairness, followed by the PSO algorithm to generate neuron-level patches for repairing the model. ❸ **GRFT** [9]. As the latest state-of-the-art individual fairness repair framework, GRFT designs a loss function aimed at directly reducing differences in model outputs between IDI pairs.

Our method involves two hyperparameters: the number of iterations T in Step 1, and the learning rate. We fix them to 200 and 0.001 across all scenarios.

4) Evaluation Metrics: We evaluate the repaired models from two perspectives: improvement in fairness and preservation of original performance. We also record the time costs of all methods. Details of fairness metrics are described below: **Fairness improvement**. Three metrics are considered for the fairness evaluation. The Certified Unfair Rate (CUR) assesses the percentage of inputs $\boldsymbol{x}$ in the repair set $\mathcal{D}_{\text{r}}$ for which the repaired model $\tilde{f}$ still produces biased outputs over $\mathcal{S}(\boldsymbol{x})$:

$$\text{CUR} = \frac{1}{|\mathcal{D}_{\text{r}}|} \sum_{i=1}^{|\mathcal{D}_{\text{r}}|} \mathbb{I}\left(\mathcal{U}(\tilde{f}, \boldsymbol{x}^i, \mathcal{S}(\boldsymbol{x}^i)) \neq \emptyset\right)$$

TABLE II: Fairness improvement results for single-attribute repair. The best values are highlighted in **bold**.

Metrics	Methods	Compas			Adult			German		Bank	Avg (Relative Drop)
		gender	race	age	gender	race	age	gender	age	age	
CUR	Original	5.00%	9.00%	42.00%	3.00%	2.00%	14.00%	3.00%	4.00%	1.00%	9.22% (-)
	FLIP	**0.00%**	6.00%	2.00%	2.00%	4.00%	8.00%	7.00%	1.00%	2.00%	3.56% (61.45%↓)
	CARE	**0.00%**	2.00%	17.00%	2.00%	2.00%	6.00%	**0.00%**	1.00%	**0.00%**	3.33% (63.86%↓)
	GRFT	**0.00%**	1.00%	16.00%	**0.00%**	1.00%	7.00%	**0.00%**	**0.00%**	**0.00%**	2.78% (69.88%↓)
	Ours	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00% (100.00%↓)**
IDI-D	Original	6.25%	8.77%	44.86%	2.74%	4.15%	21.64%	2.60%	2.90%	1.12%	10.56% (-)
	FLIP	1.80%	8.25%	4.83%	2.95%	5.56%	8.14%	4.30%	0.40%	0.93%	4.13% (60.91%↓)
	CARE	3.19%	5.75%	16.96%	1.89%	2.69%	5.60%	0.10%	3.30%	1.39%	4.54% (56.99%↓)
	GRFT	**0.00%**	2.37%	19.91%	0.98%	**0.64%**	2.38%	0.20%	0.10%	0.56%	3.02% (71.44%↓)
	Ours	**0.00%**	**0.00%**	**0.49%**	**0.79%**	1.48%	**0.83%**	**0.00%**	**0.00%**	**0.29%**	**0.43% (95.93%↓)**
IDI-S	Original	1.76%	2.73%	10.94%	0.65%	0.53%	3.65%	7.43%	8.77%	6.67%	4.79% (-)
	FLIP	0.63%	1.61%	3.08%	0.43%	0.33%	1.25%	2.99%	3.76%	3.06%	1.90% (60.27%↓)
	CARE	0.75%	0.93%	11.36%	0.75%	**0.28%**	3.30%	2.46%	7.14%	4.40%	3.49% (27.24%↓)
	GRFT	**0.00%**	0.50%	1.74%	0.59%	0.48%	2.51%	1.98%	2.31%	1.67%	1.31% (72.67%↓)
	Ours	**0.00%**	**0.00%**	**0.96%**	**0.12%**	0.30%	**0.26%**	**0.00%**	**0.00%**	**1.31%**	**0.33% (93.16%↓)**

where $\mathbb{I}$ is the indicator function and $\mathcal{U}$ is the set of all IDI instances in $\mathcal{S}(\boldsymbol{x})$ as defined in Eq. (3). For finite $\mathcal{S}(\boldsymbol{x}^i)$ (e.g., the protected attribute is gender and all non-sensitive attributes are fixed), we enumerate all possible items to check whether $\mathcal{U}(\tilde{f}, \boldsymbol{x}^i, \mathcal{S}(\boldsymbol{x}^i))$ is empty. For infinite cases, we employ the method from [42] that certify the model by solving an MILP. This MILP precisely computes the model output ranges so that we can verify whether $\tilde{f}$ is fair over $\mathcal{S}(\boldsymbol{x}^i)$. *A provable repair must guarantee that $\tilde{f}$ achieves zero CUR.*

An effective repair should generalize to unseen inputs. We assess this using two metrics: IDI-D, the proportion of inputs in the entire dataset $\mathcal{D}_{\text{full}}$ that are IDIs; and IDI-S, the proportion of inputs identified as IDIs in a set $\mathcal{D}_{\text{sample}}$ consisting of 100 000 instances randomly sampled from full input space, following prior work [8], [11], [12]. To compute IDI-D and IDI-S, we enumerate all similar inputs in $\mathcal{S}(\boldsymbol{x})$ for each instance $\boldsymbol{x} \in \mathcal{D}_{\text{full}}$ or $\mathcal{D}_{\text{sample}}$ when $\mathcal{S}(\boldsymbol{x})$ is finite; otherwise, we conduct uniform sampling over $\mathcal{S}(\boldsymbol{x})$ to obtain statistically significant estimates. For both metrics, *lower* values indicate better fairness improvement after repair.

In addition to the above metrics, we further incorporate four state-of-the-art fairness testing frameworks to evaluate PROF and all baselines: ADF [8], EIDIG [11], NeuronFair [12], and GRFT [9]. We configured these frameworks following their technical papers and used them to generate IDIs for both the original and the repaired models. Fewer IDIs detected on the repaired models indicate better repair effectiveness.

B. *RQ1: How effective is* PROF *in repairing NN unfairness?*

To answer this research question, we evaluate the performance of all repair methods. The results are summarized in Table II. As we can see, all methods lead to a notable reduction in the Certified Unfairness Rate (CUR), while PROF consistently achieves provable fairness guarantees, reducing CUR to 0 across all settings. In contrast, existing baselines provide no theoretical guarantees and only reduce CUR by approximately 60–70%. In terms of generalization to unseen inputs, our method also outperforms all baselines. On both the full dataset $\mathcal{D}_{\text{full}}$ and the sampled set $\mathcal{D}_{\text{sample}}$, the models repaired by PROF yield extremely low unfairness: the average IDI-D and IDI-S rates are reduced to just 0.43% and 0.33%, corresponding to relative reductions of 95.93% and 93.16%, respectively. The best-performing baseline, GRFT, only achieves 71.44% and 72.67% reductions in IDI-D and IDI-S.

TABLE III: Accuracy of repaired models, where colors denote accuracy loss: green (≤3%), orange (3–5%), red (>5%).

Dataset	Prot. Attr.	FLIP	CARE	GRFT	Ours
Compas	gender	🟩 72.39%	🟩 73.44%	🟩 72.71%	🟩 72.06%
Compas	race	🟩 72.87%	🟩 71.42%	🟩 73.60%	🟩 72.47%
Compas	age	🟧 70.53%	🟥 65.43%	🟥 63.24%	🟧 70.28%
Adult	gender	🟩 83.65%	🟩 84.39%	🟧 82.06%	🟩 82.50%
Adult	race	🟩 83.58%	🟩 84.52%	🟧 81.07%	🟧 82.05%
Adult	age	🟩 84.51%	🟧 80.78%	🟧 80.26%	🟩 83.67%
German	gender	🟧 68.67%	🟧 69.33%	🟩 70.00%	🟩 69.67%
German	age	🟩 70.33%	🟩 72.67%	🟩 70.00%	🟩 69.67%
Bank	age	🟩 89.28%	🟩 88.80%	🟩 89.33%	🟩 89.35%
Average		🟩 77.31%	🟩 76.75%	🟧 75.81%	🟩 76.86%
Accuracy Change		🟩 - 1.81%	🟩 - 2.37%	🟧 - 3.32%	🟩 - 2.27%

Regarding performance preservation, we present the accuracy of the repaired models in Table III. We find that all methods introduce some accuracy degradation. Among them, FLIP and PROF exhibit more stable performance, with average accuracy losses of 1.81% and 2.27%, respectively. By comparison, CARE and GRFT can have a notable negative impact on model performance in certain scenarios. For instance, when repairing the Age attribute on the Compas dataset, they result in accuracy drops of nearly 10%.

As for efficiency, both FLIP and GRFT introduce negligible time overheads, taking around 1 second per setting. PROF also maintains a desirable runtime of 18 seconds on average. On the other hand, CARE incurs more time consumption (averaging 244 seconds) due to its reliance on the PSO algorithm, which often demands extensive iteration for convergence.

C. *RQ2: Can* PROF *handle multiple protected attributes and more practical fairness definition?*

To answer this question, we conduct experiments on repair with multiple sensitive attributes simultaneously. CARE is

TABLE IV: Results of fairness improvements with multiple attributes and relaxed fairness properties (G, R and A denote Gender, Race and Age). Details specifications of all relaxed fairness are provided in the supplementary material.

Dataset	Attr.	CUR				IDI-D				IDI-S			
		Original	FLIP	GRFT	Ours	Original	FLIP	GRFT	Ours	Original	FLIP	GRFT	Ours
Compas	G-R	19.00%	7.00%	3.00%	**0.00%**	16.92%	7.63%	6.59%	**0.00%**	4.49%	1.94%	1.30%	**0.00%**
Compas	G-A	49.00%	13.00%	20.00%	**0.00%**	51.65%	7.57%	25.84%	**1.62%**	12.71%	3.57%	1.57%	**1.33%**
Compas	R-A	55.00%	13.00%	13.00%	**0.00%**	55.54%	7.18%	15.04%	**3.68%**	13.56%	2.91%	**1.91%**	2.56%
Adult	G-R	7.00%	1.00%	2.00%	**0.00%**	7.59%	6.80%	3.21%	**1.78%**	1.18%	0.73%	1.11%	**0.39%**
Adult	G-A	16.00%	6.00%	6.00%	**0.00%**	23.01%	10.51%	7.60%	**3.13%**	4.31%	1.16%	3.74%	**0.78%**
Adult	R-A	17.00%	8.00%	7.00%	**0.00%**	23.01%	12.62%	7.51%	**2.71%**	4.20%	1.36%	3.52%	**0.78%**
German	G-A	4.00%	13.00%	**0.00%**	**0.00%**	6.10%	8.90%	0.40%	**0.00%**	16.11%	9.91%	4.39%	**0.00%**
Average		23.86%	8.71%	7.29%	**0.00%**	26.26%	8.74%	9.46%	**1.85%**	8.08%	3.08%	2.51%	**0.83%**
Relative Drop		-	63.47%↓	69.46%↓	**100.00%**↓	-	66.70%↓	63.99%↓	**92.97%**↓	-	61.88%↓	69.00%↓	**89.67%**↓
Adult	G-ϕ_A	3.00%	6.00%	1.00%	**0.00%**	3.26%	4.97%	1.43%	**1.22%**	0.73%	0.52%	0.67%	**0.18%**
Adult	A-ϕ_A	14.00%	6.00%	5.00%	**0.00%**	21.98%	6.67%	6.68%	**0.61%**	3.73%	0.90%	3.07%	**0.32%**
Adult	R-ϕ_A	2.00%	4.00%	1.00%	**0.00%**	4.70%	6.42%	1.78%	**1.20%**	0.62%	0.39%	0.60%	**0.32%**
Bank	A-ϕ_B	2.00%	5.00%	2.00%	**0.00%**	3.56%	3.72%	2.04%	**0.11%**	23.75%	18.80%	6.82%	**2.75%**
German	G-ϕ_G	3.00%	8.00%	**0.00%**	**0.00%**	3.10%	5.30%	0.20%	**0.00%**	7.98%	4.01%	2.07%	**0.00%**
German	A-ϕ_G	4.00%	**0.00%**	**0.00%**	**0.00%**	3.50%	0.10%	0.10%	**0.00%**	9.35%	1.99%	2.39%	**0.00%**
Average		4.67%	4.83%	1.50%	**0.00%**	6.68%	4.53%	2.04%	**0.52%**	7.69%	4.44%	2.60%	**0.59%**
Relative Drop		-	-3.57%↓	67.86%↓	**100.00%**↓	-	32.22%↓	69.49%↓	**92.17%**↓	-	42.34%↓	66.18%↓	**92.27%**↓

excluded from this evaluation since its current implementation does not support this setting.

As shown in the upper panel of Table IV, the original models exhibit higher unfairness prior to repair. This is expected since multiple protected attributes can combine to form a larger set of similar individuals, making it naturally more challenging for the model to satisfy fairness constraints. Despite this increased challenge, PROF consistently achieves provable repair by completely eliminating unfair behaviors on the repair set (i.e., CUR = 0). Models repaired by FLIP and GRFT, however, often continue to exhibit discrimination, with CUR reductions under 70%. Our method also demonstrates remarkable generalization compared to existing approaches. It attains significantly lower IDI-D and IDI-S scores across all settings, with average mitigations of 92.97% and 89.67%, respectively, except for the "Compas + Race-Age" scenario, where its performance is only 0.6% below the best method.

To further examine the effectiveness of all methods under more practical fairness definitions, we conduct a set of experiments on repairing models with relaxed fairness properties. Following Fairify [13], we define small perturbations on one non-sensitive attribute so that two individuals are considered similar even if they are not exactly equal on this attribute. These individuals are still expected to be assigned to the same class. For instance, the specification ϕ_A indicates that for the Adult dataset, individuals are considered similar as long as the value of "hours-per-week" attribute differ by at most 1, regardless of their values on sensitive attributes.

The results are shown in the lower panel of Table IV. We observe that FLIP suffers a significant performance drop compared to the previous repair setting; in some cases, the unfairness of the repaired model even increases. This may be because it does not account for similarity between individuals whose non-sensitive attributes differ slightly and thus fails to enforce fairness under relaxed definitions. GRFT maintains similar performance as before, achieving around 70% reduction in unfairness metrics. In contrast, PROF consistently outperforms existing approaches across all scenarios, achieving complete unfairness elimination on the repair set, along with over 90% reduction in both IDI-D and IDI-S.

In terms of accuracy, all methods exhibit slightly increased performance degradation compared to the single-attribute repair setting. Both FLIP and PROF maintain stable performance, with average accuracy losses of 2.12% and 2.90%, respectively, which are significantly better than GRFT (5.03%).

D. RQ3: How does PROF *perform under various testing frameworks?*

To answer this research question, we employ four state-of-the-art NN fairness testing frameworks to evaluate PROF and all baselines. Specifically, we follow the original configurations of these frameworks and used them to generate IDIs for both the original and the repaired model. We denote the number of IDIs detected on the original model as N_{ori} and those detected on the repaired model as N_{repair}. To measure repair effectiveness under each testing framework, we report the ratio N_{repair}/N_{ori}, where a smaller ratio indicates that fewer IDIs are detected and thus better fairness improvement.

The results are presented in Table V. Overall, all methods reduce model bias, broadly consistent with our earlier findings. However, we find that these testing frameworks reveal model defects more precisely and effectively than evaluation based on uniform sampling, especially in certain settings. For instance, all baseline methods especially FLIP and CARE perform poorly on the German dataset, and in some cases the repaired models even exhibit more IDIs than the original ones (ratios greater than 1). In contrast, our method shows consistently better performance, effectively reducing bias across all settings. In particular, PROF achieves average ratios of 0.06, 0.12, 0.05, and 0.16 under the four testing tools, corresponding to 94%, 88%, 95%, and 84% reductions in IDIs, respectively. These results are in close agreement with our earlier sampling-

TABLE V: Results of fairness evaluation with various testing frameworks. Colors indicate repair effectiveness: green for strong unfairness mitigation (ratios near 0), orange for minor change (ratios near 1), and red for increase (ratios above 1).

Dataset	Attr.	FLIP				CARE				GRFT				PROF			
		ADF	EIDIG	NF	GRFT	ADF	EIDIG	NF	GRFT	ADF	EIDIG	NF	GRFT	ADF	EIDIG	NF	GRFT
Compas	G	0.28	0.24	0.36	0.33	0.28	0.36	0.45	0.42	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	R	0.53	0.63	0.59	0.37	0.42	0.45	0.35	0.22	0.17	0.18	0.18	0.11	**0.00**	**0.00**	**0.00**	**0.00**
	A	0.28	0.34	0.28	0.25	0.92	0.95	1.04	0.91	0.21	0.20	0.16	0.14	**0.12**	**0.13**	**0.09**	**0.08**
	G-R	0.29	0.32	0.29	0.17	0.32	0.34	0.29	0.17	0.44	0.44	0.29	0.17	**0.00**	**0.00**	**0.00**	**0.00**
	G-A	0.70	0.72	0.64	0.51	0.79	0.72	0.72	0.57	0.21	0.23	0.13	0.10	**0.15**	**0.15**	**0.10**	**0.08**
	R-A	0.30	0.38	0.38	0.31	0.48	0.47	0.36	0.29	**0.22**	**0.25**	**0.14**	**0.12**	**0.22**	0.31	0.19	0.15
Adult	G	1.07	0.68	0.41	0.79	**0.05**	**0.04**	0.04	**0.19**	0.43	0.26	0.53	0.78	0.06	0.28	**0.02**	0.31
	R	0.85	0.80	0.77	1.11	0.36	0.41	0.23	0.93	0.73	0.65	0.80	0.96	**0.18**	**0.36**	**0.09**	**0.74**
	A	0.26	0.18	0.35	0.56	1.15	1.12	0.54	0.43	1.08	0.96	0.86	0.78	**0.00**	**0.01**	**0.03**	**0.08**
	G-R	0.61	0.92	0.47	0.64	0.70	0.62	0.32	0.64	0.69	0.63	0.71	0.93	**0.09**	**0.42**	**0.04**	**0.44**
	G-A	0.35	0.29	0.49	0.61	1.05	0.97	0.84	0.95	1.03	1.01	0.97	0.67	**0.05**	**0.06**	**0.16**	**0.21**
	R-A	0.40	0.34	0.47	0.47	1.14	1.11	0.80	1.42	0.87	0.86	0.90	0.52	**0.03**	**0.08**	**0.12**	**0.28**
German	G	1.07	0.94	5.79	1.43	1.11	1.04	0.13	0.67	1.20	1.01	0.05	0.63	**0.00**	**0.00**	**0.00**	**0.00**
	A	1.20	1.13	0.02	0.70	1.08	1.15	1.40	0.63	1.16	1.04	0.02	0.68	**0.00**	**0.00**	**0.00**	**0.00**
	G-A	1.02	1.03	2.79	1.30	0.88	0.85	0.23	0.83	0.86	0.83	0.02	0.64	**0.00**	**0.00**	**0.00**	**0.00**
Bank	A	0.62	0.58	0.10	0.71	1.27	1.52	0.96	0.79	0.74	0.80	0.23	0.82	**0.04**	**0.16**	**0.03**	**0.22**
Average		0.61	0.60	0.89	0.64	0.75	0.76	0.54	0.63	0.63	0.58	0.37	0.50	**0.06**	**0.12**	**0.05**	**0.16**

based generalization evaluation (i.e., nearly 90% reduction). By comparison, the best-performing baseline, GRFT, reduces IDIs by no more than 70%, further confirming the superior generalization of our method against all baselines.

E. RQ4: How do the two key steps of PROF *impact repair?*

To answer this question, we first examine the impact of the first step, which performs a progressive bounds tightening process. Specifically, we track the dynamics of the two losses minimized during this step: $\mathcal{L}_{\text{bce}}$, the BCE loss computed on a small subset of training data $\mathcal{D}$c, and $\mathcal{L}_{\text{fair}}$, a fairness-aware loss measuring the relative ℓ_1-norm reduction between the current and original bounds. Due to space limitations, we present only a subset of the results in Figure 4; the full version is included in the supplementary material. We observe that $\mathcal{L}_{\text{fair}}$ steadily decreases as epochs progress, eventually reaching a relatively low level. This indicates that the concrete bounds after repair are significantly tightened compared to before, demonstrating Step 1 successfully calibrates and reduces model bias. Meanwhile, $\mathcal{L}_{\text{bce}}$ remains stable, showing that model's knowledge is largely preserved. These results highlight that this step effectively balances fairness enhancement and accuracy retention, with $\mathcal{L}_{\text{fair}}$ playing a pivotal role in calibrating the model without sacrificing its fidelity.

We further analyze the contribution of the second step in PROF. Figure 5 compares PROF with the naive repair method that relies on loose concrete bounds. As shown in the top panel, models repaired by PROF consistently achieve higher accuracy, suggesting that the naive method leads to excessive modifications of the final layer as it requires fairness constraints are satisfied across the box formed by concrete bounds. This observation is further supported by the bottom panel, where we report the optimal values of the respective MILP formulations solved by Gurobi. PROF consistently yields smaller solutions, indicating that the tighter symbolic bounds enable less adjustment to the model.

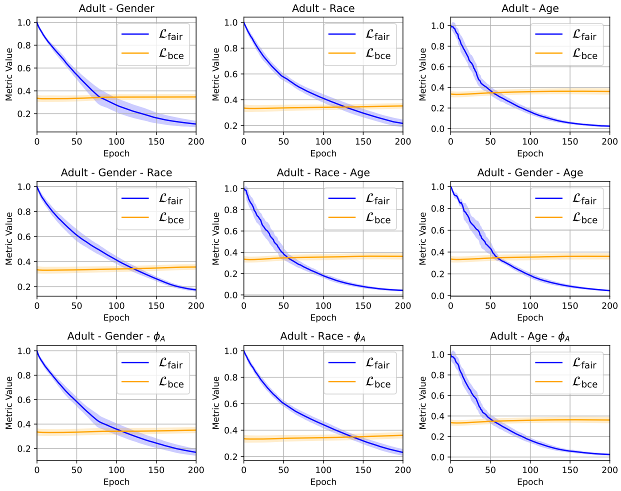

Fig. 4: Loss dynamics during the bounds tightening in Step 1 of PROF. Shaded areas denote the variance across 10 runs.

V. DISCUSSION

Scalability to more complex models: PROF can scale to more complex models since it only requires the final layer to be linear, which holds for most DNNs. A potential limitation when scaling to these models is computational cost, since MILP is theoretically NP-hard and its complexity grows exponentially with the number of integer variables. However, this number is irrelevant to the model size in our formulation, which could mitigate the exponential explosion issue. The remaining variables and constraints form a standard linear program solvable in polynomial time. Nevertheless, we agree that the overall MILP complexity cannot be guaranteed to be polynomial due to its inherent NP-hardness.

Relation to robustness repair: PROF achieves provable fairness repair through two carefully designed steps. The first

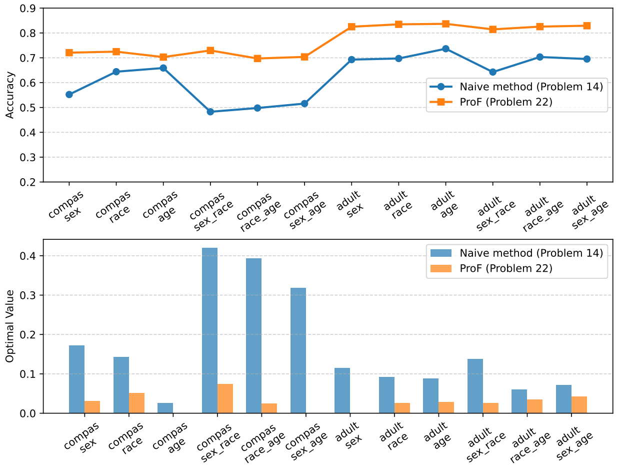

Fig. 5: Comparison between the naive repair method and PROF. Top: Accuracy of models repaired by the naive method and PROF. Bottom: Optimal values of the MILP formulated by the naive method (Problem 14) and PROF (Problem 22).

step mitigates bias by tightening concrete bounds to promote feature consistency among similar individuals, regardless of the ground truth label. This method is tailored for fairness and cannot apply to robustness repair directly, which requires all perturbed inputs to be classified correctly. In the second step, we formulate a unified constraint solving problem. The key challenge here is the presence of non-linear terms in the MILP, which we address by introducing the dual theorem. This technique can also be adapted to address a similar problem in robustness repair. Beyond this issue, fairness repair exhibits another distinctive challenge: it involves disjunctive constraints that cannot be handled by existing solvers. To address this, we carefully design and introduce integer variables to transform disjunctive constraints into linear form. Overall, while some techniques in the second step can be adapted to robustness repair, the first step is fairness-specific, which makes PROF currently unique to fairness.

VI. Related Work

NN Fairness. Fairness for DNNs is commonly categorized into *group fairness* and *individual fairness*. Group fairness considers whether different demographic groups receive statistically similar outcomes [43], [44]. Despite its simplicity in metric design, it only captures statistical parity and may result in unfairness at the individual level. In contrast, individual fairness [45], [46] stipulates that similar individuals should be treated similarly by the model, typically formalized as the constraint that predictions remain consistent when only the sensitive attributes are changed. It enables finer, instance-level analysis and broader use of testing and verification techniques.

Various methods have been developed to test and verify individual fairness in DNNs. ADF [8] and EIDIG [11] search for IDIs near the decision boundary by leveraging the loss function gradient. NeuronFair [12] further optimizes test generation by computing gradients only for biased neurons identified via sensitivity analysis. More recently, [47] introduces extreme value theory to fairness testing. It models the worst case counterfactual bias and develops a randomized test-case generation algorithm to collect tail samples with statistical guarantees. Beyond testing, several works have been proposed to formally verify NN fairness. For example, [15] learns Markov Chains from a given model to formally verify group fairness with probabilistic guarantees. On the other hand, DeepGemini [48] encodes individual fairness constraints into SMT formulas but is limited in scalability. Fairify [13] improves verification efficiency by decomposing the problem and pruning neurons. FairQuant [14] further enhances scalability and precision via abstraction-refinement, and additionally quantifies the proportion of inputs that are certifiably fair or unfair.

Regarding unfairness mitigation, several works [18], [9], [16], [49] use heuristic algorithms to correct model bias but lack formal guarantees. Shifty [50] presents a fairness training method, offering high-confidence fairness guarantees under demographic shift. NeuFair [19] and AutoRIC [51] are recently proposed methods that target group fairness repair: the former leverages simulated annealing to optimize repair solutions with statistical guarantees, while the latter uses constraint solving. Unlike these methods focusing on group fairness, this work aims to provide *deterministic*, *provable* guarantees for *individual fairness repair*.

NN Repair. There exists a broader line of work on general NN repair that targets other correctness properties. Among them, non-provable methods [16], [52], [53], [54], [55], [56] typically follow a two-step pipeline: first localizing faulty neurons or parameters, then applying heuristic techniques to adjust them. These methods often rely heavily on sufficient data and lack rigorous guarantees. To provide guarantees, several recent works [57], [58], [59] leverage constraint solvers to calculate parameter changes ensuring property satisfaction. However, these methods are not specifically designed for fairness and thus cannot be directly applied to fairness repair.

VII. Conclusion

We present PROF, a provable fairness repair framework for DNNs. It leverages interval bound propagation to calculate concrete bounds that soundly capture the model behavior in feature space, and iteratively tightens these bounds to calibrate the biased model. PROF further synthesizes symbolic bounds to formulate a precise constraint-solving problem and introduces the duality theorem to eliminate non-linear operations to construct an MILP that provides provable guarantees for repair. Extensive experiments demonstrate that PROF significantly outperforms the state-of-the-art in terms of both effectiveness and generalization. Moreover, PROF can handle multiple sensitive attributes and relaxed fairness definitions.

Acknowledgments

This work is supported by the NSFC (No. U21B2001, No. 62402150) and the Key R&D Program of Zhejiang under Grant (No. 2025C01083).

References

[1] M. Badar, M. Haris, and A. Fatima, “Application of deep learning for retinal image analysis: A review,” *Computer Science Review*, vol. 35, p. 100203, 2020.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[3] Y. Deng, T. Zhang, G. Lou, X. Zheng, J. Jin, and Q.-L. Han, “Deep learning-based autonomous driving systems: A survey of attacks and defenses,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 7897–7912, 2021.

[4] T. Brennan, W. Dieterich, and B. Ehret, “Evaluating the predictive validity of the compas risk and needs assessment system,” *Criminal Justice and behavior*, vol. 36, no. 1, pp. 21–40, 2009.

[5] K. T. Rodolfa, H. Lamba, and R. Ghani, “Empirical observation of negligible fairness–accuracy trade-offs in machine learning for public policy,” *Nature Machine Intelligence*, vol. 3, no. 10, pp. 896–904, 2021.

[6] A. E. Khandani, A. J. Kim, and A. W. Lo, “Consumer credit-risk models via machine-learning algorithms,” *Journal of Banking & Finance*, vol. 34, no. 11, pp. 2767–2787, 2010.

[7] A. Aggarwal, P. Lohia, S. Nagar, K. Dey, and D. Saha, “Black box fairness testing of machine learning models,” in *Proceedings of the 2019 27th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*, 2019, pp. 625–635.

[8] P. Zhang, J. Wang, J. Sun, G. Dong, X. Wang, X. Wang, J. S. Dong, and T. Dai, “White-box fairness testing through adversarial sampling,” in *Proceedings of the ACM/IEEE 42nd international conference on software engineering*, 2020, pp. 949–960.

[9] L. Quan, T. Li, X. Xie, Z. Chen, S. Chen, L. Jiang, and X. Li, “Dissecting global search: A simple yet effective method to boost individual discrimination testing and repair,” in *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 2025, pp. 771–771.

[10] Z. Wang, M. Zhang, J. Yang, B. Shao, and M. Zhang, “Maft: Efficient model-agnostic fairness testing for deep neural networks via zero-order gradient search,” in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–12.

[11] L. Zhang, Y. Zhang, and M. Zhang, “Efficient white-box fairness testing through gradient search,” in *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021, pp. 103–114.

[12] H. Zheng, Z. Chen, T. Du, X. Zhang, Y. Cheng, S. Ji, J. Wang, Y. Yu, and J. Chen, “Neuronfair: Interpretable white-box fairness testing through biased neuron identification,” in *Proceedings of the 44th International Conference on Software Engineering*, 2022, pp. 1519–1531.

[13] S. Biswas and H. Rajan, “Fairify: Fairness verification of neural networks,” in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2023, pp. 1546–1558.

[14] B. H. Kim, J. Wang, and C. Wang, “Fairquant: Certifying and quantifying fairness of deep neural networks,” in *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 2024, pp. 191–203.

[15] B. Sun, J. Sun, T. Dai, and L. Zhang, “Probabilistic verification of neural networks against group fairness,” in *Formal methods: 24th international symposium, FM 2021, virtual event, November 20–26, 2021, proceedings 24*. Springer, 2021, pp. 83–102.

[16] B. Sun, J. Sun, L. H. Pham, and J. Shi, “Causality-based neural network repair,” in *Proceedings of the 44th International Conference on Software Engineering*, 2022, pp. 338–349.

[17] T. Li, Y. Cao, J. Zhang, S. Zhao, Y. Huang, A. Liu, Q. Guo, and Y. Liu, “Runner: Responsible unfair neuron repair for enhancing deep neural network fairness,” in *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, 2024, pp. 1–13.

[18] J. Chen, J. Wang, Y. Sun, P. Cheng, and J. Chen, “Isolation-based debugging for neural networks,” in *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2024, pp. 338–349.

[19] V. A. Dasu, A. Kumar, S. Tizpaz-Niari, and G. Tan, “Neufair: Neural network fairness repair with dropout,” in *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2024, pp. 1541–1553.

[20] G. Singh, T. Gehr, M. Püschel, and M. Vechev, “An abstract domain for certifying neural networks,” *Proceedings of the ACM on Programming Languages*, vol. 3, no. POPL, pp. 1–30, 2019.

[21] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, “Efficient neural network robustness certification with general activation functions,” *Advances in neural information processing systems*, vol. 31, 2018.

[22] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, “Ai2: Safety and robustness certification of neural networks with abstract interpretation,” in *2018 IEEE symposium on security and privacy (SP)*. IEEE, 2018, pp. 3–18.

[23] B. Paulsen and C. Wang, “Example guided synthesis of linear approximations for neural network verification,” in *International Conference on Computer Aided Verification*. Springer, 2022, pp. 149–170.

[24] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, “Formal security analysis of neural networks using symbolic intervals,” in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1599–1614.

[25] P. Yang, R. Li, J. Li, C. Huang, J. Wang, J. Sun, B. Xue, and L. Zhang, “Improving neural network verification through spurious region guided refinement,” in *TACAS 2021*, ser. Lecture Notes in Computer Science, vol. 12651. Springer, 2021, pp. 389–408.

[26] R. Bunel, P. Mudigonda, I. Turkaslan, P. Torr, J. Lu, and P. Kohli, “Branch and bound for piecewise linear neural network verification,” *Journal of Machine Learning Research*, vol. 21, no. 2020, 2020.

[27] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljić *et al.*, “The marabou framework for verification and analysis of deep neural networks,” in *Computer Aided Verification: 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I 31*. Springer, 2019, pp. 443–452.

[28] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, “Reluplex: An efficient smt solver for verifying deep neural networks,” in *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*. Springer, 2017, pp. 97–117.

[29] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, “Efficient formal safety analysis of neural networks,” *Advances in neural information processing systems*, vol. 31, 2018.

[30] L. H. De Figueiredo and J. Stolfi, “Affine arithmetic: concepts and applications,” *Numerical algorithms*, vol. 37, pp. 147–158, 2004.

[31] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to interval analysis*. SIAM, 2009.

[32] Z. Chen, J. M. Zhang, M. Hort, M. Harman, and F. Sarro, “Fairness testing: A comprehensive survey and analysis of trends,” *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 5, pp. 1–59, 2024.

[33] P. G. John, D. Vijaykeerthy, and D. Saha, “Verifying individual fairness in machine learning models,” in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 749–758.

[34] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh, “Automatic perturbation analysis for scalable certified robustness and beyond,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1129–1141, 2020.

[35] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. T. Vechev, “Fast and effective robustness certification,” in *NeurIPS 2018*, Montréal, Canada, 2018, pp. 10 825–10 836.

[36] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[37] Gurobi Optimization, LLC, “Gurobi Optimizer Reference Manual,” 2025. [Online]. Available: https://www.gurobi.com

[38] B. Becker and R. Kohavi, “Adult,” UCI Machine Learning Repository, 1996, DOI: https://doi.org/10.24432/C5XW20.

[39] H. Hofmann, “Statlog (German Credit Data),” UCI Machine Learning Repository, 1994, DOI: https://doi.org/10.24432/C5NC77.

[40] S. Moro, P. Rita, and P. Cortez, “Bank Marketing,” UCI Machine Learning Repository, 2012, DOI: https://doi.org/10.24432/C5K306.

[41] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, “Machine bias,” 2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[42] V. Tjeng, K. Y. Xiao, and R. Tedrake, “Evaluating robustness of neural networks with mixed integer programming,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=HyGIdiRqtm

[43] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 259–268.

[44] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," *Advances in neural information processing systems*, vol. 29, 2016.

[45] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International conference on machine learning*. PMLR, 2013, pp. 325–333.

[46] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012, pp. 214–226.

[47] V. Monjezi, A. Trivedi, V. Kreinovich, and S. Tizpaz-Niari, "Fairness testing through extreme value theory," in *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 2025, pp. 1501–1513.

[48] X. Xie, F. Zhang, X. Hu, and L. Ma, "Deepgemini: verifying dependency fairness for deep neural network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 12, 2023, pp. 15 251–15 259.

[49] T. Li, Q. Guo, A. Liu, M. Du, Z. Li, and Y. Liu, "Fairer: fairness as decision rationale alignment," in *International Conference on Machine Learning*. PMLR, 2023, pp. 19 471–19 489.

[50] S. Giguere, B. Metevier, Y. Brun, B. C. Da Silva, P. S. Thomas, and S. Niekum, "Fairness guarantees under demographic shift," in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

[51] X. Sun, W. Liu, S. Wang, T. Chen, Y. Tao, and X. Mao, "Autoric: Automated neural network repairing based on constrained optimization," *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 2, pp. 1–29, 2025.

[52] M. Usman, D. Gopinath, Y. Sun, Y. Noller, and C. S. Păsăreanu, "Nn repair: constraint-based repair of neural network classifiers," in *Computer Aided Verification: 33rd International Conference, CAV 2021, Virtual Event, July 20–23, 2021, Proceedings, Part I 33*. Springer, 2021, pp. 3–25.

[53] P. Henriksen, F. Leofante, and A. Lomuscio, "Repairing misclassifications in neural networks using limited data," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, 2022, pp. 1031–1038.

[54] J. Sohn, S. Kang, and S. Yoo, "Arachne: Search-based repair of deep neural networks," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 4, pp. 1–26, 2023.

[55] Z. Chen, J. Zhou, Y. Sun, J. Wang, Q. Xuan, and X. Yang, "Interpretability based neural network repair," in *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2024, pp. 908–919.

[56] J. Ma, P. Yang, J. Wang, Y. Sun, C.-C. Huang, and Z. Wang, "Vere: Verification guided synthesis for repairing deep neural networks," in *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, 2024, pp. 1–13.

[57] M. Sotoudeh and A. V. Thakur, "Provable repair of deep neural networks," in *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, 2021, pp. 588–603.

[58] F. Fu and W. Li, "Sound and complete neural network repair with minimality and locality guarantees," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[59] Z. Tao, S. Nawas, J. Mitchell, and A. V. Thakur, "Architecture-preserving provable repair of deep neural networks," *Proceedings of the ACM on Programming Languages*, vol. 7, no. PLDI, pp. 443–467, 2023.