LLMs Are Not a Silver Bullet: A Case Study on Software Fairness

Anonymous Author(s)

Abstract

Fairness is a critical requirement for human-related, high-stakes software systems, motivating extensive research on bias mitigation. Prior work has largely focused on tabular data settings using traditional Machine Learning (ML) methods. With the rapid rise of Large Language Models (LLMs), recent studies have begun to explore their use for bias mitigation in the same setting. However, it remains unclear whether LLM-based methods offer advantages over traditional ML methods, leaving software engineers without clear guidance for practical adoption. To address this gap, we present a large-scale study comparing state-of-the-art ML- and LLM-based bias mitigation methods. We find that ML-based methods consistently outperform LLM-based methods in both fairness and predictive performance, with even strong LLMs failing to surpass established ML baselines. To understand why prior LLM-based studies report favorable results, we analyze their evaluation settings and show that these gains are largely driven by artificially balanced test data rather than realistic imbalanced distributions. We further observe that existing LLM-based methods primarily rely on in-context learning and thus fail to leverage all available training data. Motivated by this, we explore supervised fine-tuning on the full training set and find that, while it achieves competitive results, its advantages over traditional ML methods remain limited. These findings suggest that LLMs are not a silver bullet for software fairness.

CCS Concepts

• **Software and its engineering → Software reliability**.

Keywords

Software Fairness, Bias Mitigation, Large Language Models

ACM Reference Format:

Anonymous Author(s). 2026. LLMs Are Not a Silver Bullet: A Case Study on Software Fairness. In *Proceedings of IEEE/ACM International Conference on Automated Software Engineering*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Software fairness has become a critical requirement [14, 15, 18, 24] for human-related, high-stakes software systems across socially sensitive domains, such as credit assessment [1], hiring [42], and criminal justice [29]. It requires software to provide equal opportunities or achieve comparable predictive performance across social groups defined by sensitive attributes such as sex, race, and age [18, 24]. Violations of software fairness can lead to severe societal and legal consequences [46, 51].

From the Software Engineering (SE) perspective, unfairness issues are commonly regarded as fairness bugs [18, 24]. Addressing these bugs has therefore become an important responsibility for software researchers and engineers [24], motivating a growing body of work on bias mitigation in the SE community [26]. This body of work has largely focused on tabular data, where sensitive attributes are explicitly defined, fairness-sensitive applications are prevalent, and mitigation effects can be systematically evaluated. As a result, tabular data has become the most established and widely adopted setting for software fairness research [16, 33, 49].

Within this setting, prior bias mitigation research has been dominated by traditional Machine Learning (ML) methods [20, 33, 44]. Existing ML-based methods are typically categorized into pre-processing, in-processing, and post-processing methods, which mitigate bias by modifying the training data, incorporating fairness constraints during model training, or adjusting model outputs after training, respectively [26, 33]. These methods have been extensively studied and shown effective across diverse tabular datasets, establishing ML-based mitigation as the dominant paradigm for software fairness [20, 23, 24, 48].

Recently, with the rapid rise of Large Language Models (LLMs), software engineers are increasingly adopting LLMs as a general-purpose solution for a wide range of software tasks, often replacing traditional ML pipelines with LLM-based alternatives [39]. This trend has naturally extended to fairness-sensitive tabular prediction settings, where a growing body of work has begun to explore LLM-based bias mitigation in the same tabular setting [27, 34, 41, 50, 52]. These methods typically convert tabular data into natural language descriptions and use in-context learning, where a small set of demonstrations selected from the training data is included in the prompt alongside the test instances. The LLM then makes predictions conditioned on these demonstrations without updating its parameters. Prior studies [34, 41, 50] report that carefully selected demonstrations can improve both predictive performance and fairness, suggesting that LLM-based methods may offer a promising alternative to traditional ML-based bias mitigation.

However, it remains unclear whether LLM-based methods offer advantages over traditional ML methods, leaving software engineers without clear guidance for practical adoption. This gap is partly due to a lack of direct comparisons between the two paradigms. For instance, recent LLM-based studies [27, 34, 41] propose in-context learning approaches for bias mitigation but do not compare against traditional ML-based methods. Similarly, recent work on ML-based bias mitigation [22, 23, 55] rarely considers LLM-based approaches.

To address this gap, we conduct a large-scale empirical study comparing state-of-the-art ML- and LLM-based bias mitigation methods on widely used real-world tabular datasets. Specifically, we evaluate eight representative ML-based methods across four common ML models, and eight LLM-based methods under a unified

IEEE/ACM International Conference on Automated Software Engineering, German

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

experimental setting. Our results show that LLM-based methods are generally less effective than traditional ML-based approaches. On average, ML-based methods achieve both better fairness (48.3%–59.6% improvement across different fairness metrics) and higher predictive performance (e.g., 11.1% higher accuracy and 9.3% higher F1-score). This trend persists even when stronger LLMs are used.

However, existing LLM-based bias mitigation studies often report promising results, motivating us to investigate this discrepancy. We find that these studies [34, 41, 50] typically evaluate on artificially balanced test sets with equal proportions across demographic groups and labels, whereas we follow real-world imbalanced distributions. To examine the impact of this difference, we compare LLM-based methods under both settings and observe that balanced distributions can improve fairness by 36.2%–138.4%. Such improvements, however, may not generalize to real-world settings, and thus provide limited guidance for software engineers in practice.

Another potential explanation for the performance gap between LLM- and ML-based methods is that existing LLM-based methods [34, 50] primarily rely on in-context learning, which uses only a small subset of the training data as demonstrations. To examine whether broader access to training data can improve LLM-based methods, we investigate supervised fine-tuning on the full training set. We consider both standard fine-tuning and fine-tuning combined with traditional data-level pre-processing. Our results show that fine-tuning substantially improves over in-context learning, but its advantages over traditional ML-based methods remain limited: no significant gains are observed in any fairness evaluations, and gains are observed in only 37.5% of predictive performance comparisons.

Overall, this paper makes the following contributions:

- A large-scale empirical study comparing state-of-the-art ML- and LLM-based bias mitigation methods on real-world tabular datasets, demonstrating that ML-based methods consistently outperform LLM-based methods.
- An in-depth analysis of LLM-based methods, showing that their effectiveness is sensitive to evaluation settings and learning paradigms: balanced test distributions can inflate fairness, while fine-tuning still yields limited gains over traditional ML methods.
- An open-source replication package [13], including all scripts and data used in this study, to facilitate future research.

2 Related Work

This section summarizes existing work highly relevant to this study.

Software Fairness. Fairness has become an important requirement for modern software systems, motivating extensive research in the SE community. Existing software fairness research primarily focuses on tabular classification, the most widely studied setting in this area [26, 33, 44]. Tabular classification supports decision-making across domains such as finance, healthcare, and criminal justice, where prediction outcomes directly affect individuals and social groups [15, 18, 24]. Tabular datasets typically include sensitive attributes (e.g., sex, race, and age) that define privileged and unprivileged groups [44]. Because historical data may encode societal biases, learned models can produce systematically different outcomes across demographic groups, leading to fairness violations in intelligent software systems [49]. Ensuring fairness in this setting is therefore essential for ethical decision-making.

Traditional ML-Based Bias Mitigation. There have been extensive bias mitigation methods based on traditional ML methods [33]. These methods are commonly categorized into three types: pre-processing, in-processing, and post-processing. *Pre-processing methods* modify the training data before learning, for example through reweighting, resampling, and synthetic data generation [35, 38, 45, 54]. LTDD [38], for instance, uses linear regression to identify non-sensitive features and feature values that are strongly associated with sensitive attributes, and excludes the biased parts while preserving as much unbiased information as possible. *In-processing methods* mitigate bias during model training by modifying the learning objective or training procedure [56]. *Post-processing methods* mitigate bias after training by adjusting prediction outputs [31, 36]. Prior ML-based methods have also explored ensemble techniques [25, 55] that combine multiple mitigation strategies or models to leverage complementary strengths.

Emerging LLM-Based Bias Mitigation. With the rapid advancement of LLMs, recent work has begun to explore their use for bias mitigation in tabular classification tasks [41, 50]. In this setting, tabular instances are typically converted into natural language descriptions and provided to LLMs for prediction. LLM-based methods usually operate through in-context learning, where the model conditions on task instructions and a small set of labeled examples (namely *demonstrations*) included in the prompt. Unlike traditional ML-based methods, which intervene on training data, learning objectives, or model outputs, existing LLM-based bias mitigation methods primarily influence fairness through prompt design and demonstration selection.

LLM-based bias mitigation methods can be broadly categorized based on how demonstrations are constructed and used. *Zero-shot* classification uses only task instructions and the input instance, serving as a non-mitigated baseline that reflects the model's inherent behavior [47, 50]. In contrast, *few-shot* strategies mitigate bias by controlling the demonstrations included in the prompt. Prior work [41, 50] shows that fairness outcomes are highly sensitive to the composition and distribution of demonstrations. Existing methods therefore mainly improve fairness by balancing demographic groups and labels in the prompt, modifying demonstration labels, or selecting fairness-aware demonstration sets [41, 50]. More advanced methods search for fairness-aware demonstration sets that better balance fairness and predictive performance. For example, *Fairness via Clustering-Genetic (FCG)* [34] clusters training data to identify representative samples and applies evolutionary search to optimize demonstration selection.

Despite advances in both ML- and LLM-based bias mitigation methods, direct comparisons between the two paradigms remain limited. Existing studies typically evaluate methods within a single paradigm rather than across paradigms. For example, recent LLM-based work [34] compares different in-context learning strategies for bias mitigation but does not benchmark them against traditional ML-based approaches. As a result, it remains unclear how these two paradigms compare, leaving software engineers without clear guidance for practical adoption. To address this gap, this paper presents a large-scale empirical study comparing state-of-the-art methods from both paradigms.

3 Experimental Setup

This section describes the evaluation settings for this study.

3.1 Datasets

We evaluate bias mitigation methods on six tasks derived from three widely adopted real-world tabular datasets.

Table 1 summarizes the three datasets used in our study, include Adult [3], Compas [2], and Credit [1]. All of them are well-established benchmarks in prior fairness research [25, 33, 45, 55]. They consist of real-world data collected from fairness-critical decision domains, including income prediction, recidivism risk assessment, and credit scoring. They also cover the three most commonly studied sensitive attributes in the literature, namely sex, race, and age [33]. Based on the sensitive attributes available in each dataset, we define six fairness evaluation tasks, where each task corresponds to a specific dataset paired with a designated sensitive attribute. Specifically, the tasks include *Adult-Sex*, *Adult-Race*, *Compas-Sex*, *Compas-Race*, *Credit-Sex*, and *Credit-Age*.

3.2 ML-Based Bias Mitigation Methods

We evaluate one default baseline and eight traditional ML-based bias mitigation methods. We select representative and well-performing methods across all intervention stages, including pre-processing, in-processing, post-processing, and ensemble methods. Our selection encompasses both widely adopted methods in prior top-tier SE studies [23, 31, 33, 36, 44, 45, 56] and recently proposed state-of-the-art methods [38, 54, 55], ensuring a comprehensive assessment that covers both classic foundations and the latest advancements in the field.

- **Default baseline.** This configuration uses the original training data and the default ML pipeline without any bias mitigation.
- **Pre-processing methods.** These methods mitigate bias by modifying the training data before model training. (i) **FairMask** [45] uses an extrapolation model to adjust protected-attribute information before classification, thereby mitigating potential discrimination; (ii) **LTDD** (Linear-regression based Training Data Debugging) [38] mitigates bias by identifying and excluding "biased components" of features, thereby cleaning the training data to support fairer predictions; (iii) **CoT** (Correlation Tuning) [54] is a statistical intervention that adjusts data correlations via the Phi-coefficient and decouples sensitive attributes from the decision-making process to improve the fairness performance trade-off.
- **In-processing methods.** These methods mitigate bias by incorporating fairness-aware constraints into the model training objective. **ADV** [56] applies adversarial learning to reduce the dependence between model predictions and sensitive attributes during training. It jointly optimizes a predictor and an adversary, discouraging discriminatory signals while preserving predictive performance.
- **Post-processing methods.** These methods mitigate bias by adjusting model predictions after training without modifying the model. (i) **EOP** [31] modifies output labels to equalize error rates across demographic groups, typically by aligning false positive and false negative rates; (ii) **ROC** [36] adjusts predictions for instances near the decision boundary, altering labels in uncertain regions to improve fairness while preserving overall accuracy.
- **Ensemble methods.** These methods combine multiple models or mitigation stages to balance fairness and predictive performance. (i) **MAAT** [25] trains separate models optimized for predictive performance and fairness, and combines their predictions to balance the two objectives; (ii) **MirrorFair** [55] constructs a counterfactual dataset by flipping sensitive attributes, trains models on both the original and counterfactual data, and adaptively combines their predictions to produce fairer decisions.

Model Selection. For the traditional ML paradigm, we select four models widely studied in fairness research [22, 23, 25, 33, 55] for training: Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), and Deep Neural Network (DNN). For the DNN architecture, we utilize a fully connected network with five hidden layers consisting of 64, 32, 16, 8, and 4 units, respectively.

3.3 LLM-Based Bias Mitigation Methods

We evaluate zero-shot prompting as the default baseline, along with eight LLM-based mitigation methods via in-context learning (ICL). We select these methods because they are representative prompt-level strategies studied in prior LLM fairness literature [34, 41, 50] and cover the major intervention types in ICL-based bias mitigation, including baseline prompting, random demonstration selection, label manipulation, distribution-controlled demonstration construction, and fairness-aware demonstration search.

Let k denote the number of demonstrations included in a prompt. Let $A \in \{0, 1\}$ denote the sensitive attribute (1: privileged, 0: unprivileged) and $Y \in \{0, 1\}$ denote the original class label, where $Y = 1$ represents the favorable label. For a given prompt, we define r_A as the proportion of demonstrations drawn from the unprivileged group ($A = 0$), and r_Y as the proportion of demonstrations assigned the favorable label ($Y = 1$).

- **Zero Shot [41, 50].** The LLM performs classification based solely on task instructions and the target instance, without demonstrations. This reflects the model's inherent behavior without contextual calibration.
- **Random [41, 50].** A fixed set of k labeled demonstrations is randomly sampled from the training data and included in the prompt.
- **Label Flipping [41].** An intentional contextual intervention in which the labels of selected demonstrations are systematically flipped to weaken the empirical correlation between sensitive attributes and favorable outcomes. For example, a demonstration originally labeled as $(A = 0, Y = 0)$ is modified to $(A = 0, Y = 1)$, therefore increasing the representation of favorable outcomes for the unprivileged group within the prompt.
- **Distribution-controlled Few-Shot [34, 50].** Demonstrations are selected under three distinct distribution (S1, S2, S3) to analyze the impact of group and label representation.
  (i) **balanced (S1)**, where $r_A = 0.5$ and $r_Y = 0.5$, meaning half of the demonstrations are drawn from the unprivileged group and half are assigned the favorable label;
  (ii) **minority-balanced (S2)**, where $r_A = 1.0$ and $r_Y = 0.5$, meaning all demonstrations are drawn from the unprivileged group and half are assigned the favorable label;

Table 1: Benchmark datasets.

Name	Size	Sensitive attr(s)	Favorable label	Description
Adult [3]	48,842	sex, race	income > 50K	Predict whether an individual's income exceeds 50K.
Compas [2]	6,172	sex, race	no recidivism	Predicting criminal defendant recidivism.
Credit [1]	30,000	sex, age	no default	Predicting whether a customer will default on payment.

(iii) **minority-unbalanced (S3)**, where $r_A = 1.0$ and $r_Y = 1.0$, meaning all demonstrations are drawn from the unprivileged group and all are assigned the favorable label.

- **FCG Variants [34].** Following the FCG strategy proposed in previous literature, candidate demonstration subsets are first constructed from clustered training samples and then optimized with a genetic search procedure based on both fairness and predictive performance. From the resulting optimized candidate pool $\mathcal{P}$, we select the final k-shot demonstration sets that satisfy the corresponding distribution requirements (S1, S2, S3) defined in the Few-Shot configurations. This gives three variants in our evaluation: **FCG-S1**, **FCG-S2**, and **FCG-S3**.

Model Selection. We use GPT-4o-mini [4] as the primary LLM to implement these methods. We select GPT-4o-mini because it is a cost-effective model from a major AI vendor and has been widely used in prior LLM evaluation studies [21, 30, 40]. Its relatively low inference cost also makes it suitable for our large-scale evaluation across multiple mitigation methods, six fairness evaluation tasks, and repeated runs.

3.4 Evaluation Metrics

Bias mitigation methods often involve a trade-off between fairness and predictive performance [23, 25, 33]; therefore, we evaluate their effectiveness using both fairness and performance metrics.

3.4.1 Fairness Metrics. We adopt group fairness metrics widely utilized in the previous literature to measure software bias [16, 17, 19, 25, 44, 57]. As defined in Section 3.3, let A be a protected attribute, with 1 representing the privileged group and 0 the unprivileged group. Let Y be the original class label and $\hat{Y}$ the predicted label, where 1 denotes the favorable class and 0 the unfavorable class; let P denote the probability. Following previous work [22, 25, 33, 44, 55], we use the absolute values of these metrics, where a value of 0 indicates perfect fairness and larger values indicate higher levels of bias.

- **Statistical Parity Difference (SPD)** measures the disparity in the probability of receiving a favorable prediction between unprivileged and privileged groups:

$$SPD = |P(\hat{Y} = 1|A = 0) - P(\hat{Y} = 1|A = 1)| \tag{1}$$

- **Average Odds Difference (AOD)** evaluates the average of the absolute differences in false-positive rates and true-positive rates between the two demographic groups:

$$\begin{aligned} AOD = \frac{1}{2}(&|P(\hat{Y} = 1|A = 0, Y = 0) - P(\hat{Y} = 1|A = 1, Y = 0)| \\ &+|P(\hat{Y} = 1|A = 0, Y = 1) - P(\hat{Y} = 1|A = 1, Y = 1)|) \end{aligned} \tag{2}$$

- **Equal Opportunity Difference (EOD)** measures the true-positive rate difference between unprivileged and privileged groups:

$$EOD = |P(\hat{Y} = 1|A = 0, Y = 1) - P(\hat{Y} = 1|A = 1, Y = 1)| \tag{3}$$

3.4.2 Performance Metrics. To measure predictive performance, we utilize traditional classification metrics, including precision (Prec.), recall (Rec.), F1-score (F1), and accuracy (Acc.). For a given class, precision is defined as the proportion of samples predicted as that class that actually belong to it, while recall denotes the proportion of samples belonging to a class that are correctly predicted. F1-score is calculated as the harmonic mean of precision and recall. Following established practices in SE research [22, 25, 33, 57], we report the macro-average values for precision, recall, and F1-score to enable a balanced comparison of the overall performance across both favorable and unfavorable classes. This involves calculating each metric for each class individually and then averaging the results. Accuracy, which measures the frequency of correct predictions, remains a standard metric in fairness literature [22, 24, 25]. For all four metrics, larger values indicate superior predictive performance, with 1 representing perfect prediction.

3.5 Implementation Details

We briefly describe the implementation details used in our study.

Experimental environment. All experiments are implemented in Python 3.11.9. For the traditional ML paradigm, we use IBM AIF360 [5] for mitigation methods and fairness metrics, *Scikit-learn* for traditional classifiers, and *TensorFlow Keras* for the DNN model. LLM inference is performed via API calls [10, 11], using the default temperature for each evaluated model to reflect real-world application scenarios, where users typically interact with these models in their out-of-the-box configurations. Fine-tuning experiments are conducted on a machine with one NVIDIA RTX 5090 GPU (32GB) and 90GB RAM.

Data preprocessing and split. Before partitioning the datasets, we perform a standard preprocessing step by removing all instances containing missing values (NA) to ensure data quality and consistency across different models. To mitigate the impact of randomness and ensure the soundness of our comparisons, we then adopt a consistent evaluation protocol across both traditional ML and LLM paradigms. In each experimental run, the cleaned dataset is randomly split into 80% training data and 20% testing data using different random seeds. We do not employ cross-validation to maintain alignment with the standard prompting and fine-tuning procedures used in recent LLM studies. Following established practices in software fairness literature [22, 25, 55] to reduce random variation, each bias mitigation task is repeated 3 times with different random seeds, and we report the average predictive performance and fairness metrics across these runs.

Prompt serialization. To facilitate LLM processing of tabular data, we serialize each structured record into a natural language string. As in prior studies [34, 41], each instance is represented as a concatenation of feature-value pairs (e.g., "f_1 is $x_1, \ldots, f_d$ is x_d"). As illustrated in Figure 1, each prompt consists of a task description,

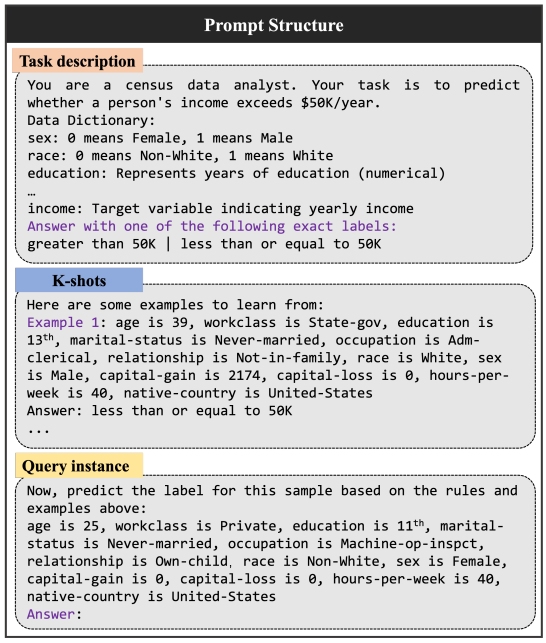

Figure 1: Prompt structure for the LLM paradigm.

k-shot demonstrations, and a query instance. Following previous studies [23, 34, 50], we set $k = 16$; in the zero-shot setting, we use the same prompt format without demonstrations.

4 Research Questions and Results

This section presents our research questions (RQs) and answers them through experimental results.

4.1 RQ1: How do LLM-based and ML-based bias mitigation methods compare?

4.1.1 Motivation and Methodology. In this RQ, we compare the overall effectiveness of the traditional ML and LLM paradigms for tabular bias mitigation across six fairness evaluation tasks. Table 2 reports the detailed results. Each value is averaged over three repeated runs with different random train-test splits. For traditional ML, each method result is further averaged across four classification models (i.e., SVM, RF, LR, and DNN), while all results for the LLM paradigm are obtained using GPT-4o-mini [4]. In Table 2, *Default* and *Zero-Shot* denote the baseline settings for the ML and LLM paradigms, respectively, and *Average* denotes the mean result of the eight bias mitigation methods under each paradigm.

To assess whether the predictive-performance and fairness differences between the two paradigms are statistically significant, we conduct the non-parametric Mann-Whitney U-test [43] on the raw outcomes. For each task, we compare the raw outcomes of the two paradigms on the four predictive metrics reported in Table 2 (Accuracy, Recall, Precision, and F1), resulting in 24 predictive-performance comparisons (24 = 6 tasks ×4 metrics), and on the three fairness metrics (SPD, EOD, and AOD), resulting in 18 fairness comparisons (18 = 6 tasks ×3 metrics). Both groups aggregate raw results across eight mitigation methods and repeated runs, with the ML group further incorporating the corresponding base models. A difference is considered statistically significant when the resulting p-value is below 0.05. Since higher values indicate better predictive performance, a predictive case is considered to favor ML when the ML group has a higher mean value and the Mann-Whitney test produces $p < 0.05$. Since lower values indicate better fairness, a fairness case is considered to favor ML when the ML group has a lower mean value and the Mann-Whitney test produces $p < 0.05$.

4.1.2 Results. We analyze the results from both fairness and predictive performance.

Fairness. Regarding fairness, the traditional ML paradigm consistently achieves lower bias levels than the LLM paradigm across all datasets. Under identical experimental settings across six tasks, traditional ML reduces SPD, EOD, and AOD by 48.3%, 59.6%, and 51.6%, respectively, compared to the LLM paradigm. The gap is especially large on *Compas-Race*, where ML reduces EOD by 80.7% relative to the LLM average. Even when advanced demonstration strategies such as FCG-S3 improve fairness within the LLM paradigm, the remaining bias is still substantial compared with traditional ML methods. For example, on *Compas-Race*, FCG-S3 reduces EOD by 76.2% relative to the Zero-Shot baseline, but traditional ML still achieves a further 88.1% reduction relative to FCG-S3. Consistent with these observations, ML is significantly better in 13 out of 18 fairness comparisons (72.2%).

Predictive performance. Overall, the traditional ML paradigm consistently demonstrates superior predictive performance compared to the LLM paradigm across all six tasks. As shown in Table 2, ML achieves higher average Accuracy, Precision, and F1-score in every configuration, while Recall is also higher on average overall. Across all six tasks, traditional ML achieves 11.1% higher Accuracy, 9.3% higher F1-score, 11.3% higher Precision, and 4.6% higher Recall than the LLM paradigm. This advantage is statistically significant in 20 out of 24 predictive-performance comparisons (83.3%). Overall, these results confirm that traditional ML-based methods remain more effective than current LLM-based mitigation methods in both preserving predictive performance and reducing fairness disparities under practical tabular settings.

> **Ans. to RQ1:** Under identical real-world experimental settings, the traditional ML paradigm consistently outperforms the LLM paradigm in both fairness and predictive performance. Across six tasks, traditional ML reduces SPD, EOD, and AOD by 48.3%, 59.6%, and 51.6%, respectively, while achieving 11.1% higher Accuracy, 9.3% higher F1-score, 11.3% higher Precision, and 4.6% higher Recall on average. Moreover, ML is significantly better in 72.2% of the fairness comparisons and 83.3% of the predictive comparisons. Therefore, traditional ML remains the more effective and practical paradigm for tabular bias mitigation.

4.2 RQ2: Does the comparison depend on the choice of LLM?

4.2.1 Motivation and Methodology. This RQ examines whether the overall comparison between the ML and LLM paradigms depends on the choice of LLM. To answer this question, we conduct an additional targeted validation using four advanced LLMs from major AI vendors: GPT-5 [9], Gemini-2.5-flash [8], DeepSeek-v3.2 [7],

Table 2: (RQ1) Predictive performance and fairness of ML- and LLM-based bias mitigation methods across six tasks. Each paradigm includes one baseline (*Default* for ML and *Zero-Shot* for LLM) and eight mitigation methods. Metrics labeled with ↑ indicate that higher values are better, while those labeled with ↓ indicate that lower values are better. Results for ML-based methods are averaged over SVM, RF, LR, and DNN. The *Average* rows report the mean results of the eight mitigation methods within each paradigm.

Paradigm	Method	Adult-Sex							Compas-Sex							Credit-Sex						
		Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓
ML	Default	0.849	0.784	0.808	0.768	0.181	0.096	0.086	0.646	0.641	0.644	0.641	0.284	0.221	0.259	0.813	0.648	0.757	0.628	0.021	0.011	0.018
	COT	0.845	0.772	0.807	0.751	0.133	0.028	0.037	0.651	0.643	0.647	0.643	0.060	0.046	0.051	0.812	0.649	0.753	0.629	0.009	0.004	0.015
	LTDD	0.844	0.775	0.801	0.758	0.133	0.120	0.082	0.652	0.643	0.648	0.643	0.073	0.078	0.089	0.813	0.647	0.756	0.628	0.010	0.005	0.016
	FairMask	0.849	0.783	0.808	0.766	0.176	0.084	0.079	0.645	0.640	0.645	0.642	0.142	0.097	0.106	0.813	0.652	0.755	0.631	0.011	0.005	0.011
	ADV	0.841	0.762	0.806	0.739	0.048	0.218	0.117	0.650	0.645	0.647	0.646	0.348	0.269	0.329	0.819	0.672	0.758	0.647	0.008	0.007	0.023
	MAAT	0.844	0.758	0.823	0.730	0.102	0.060	0.043	0.651	0.641	0.648	0.641	0.149	0.090	0.119	0.813	0.645	0.757	0.625	0.010	0.006	0.013
	MirrorFair	0.847	0.772	0.814	0.749	0.130	0.043	0.043	0.648	0.635	0.647	0.636	0.145	0.090	0.115	0.813	0.648	0.757	0.628	0.010	0.005	0.011
	EOP	0.826	0.752	0.772	0.739	0.105	0.037	0.026	0.609	0.589	0.605	0.593	0.040	0.025	0.028	0.809	0.641	0.744	0.623	0.010	0.007	0.014
	ROC	0.776	0.737	0.726	0.781	0.036	0.157	0.118	0.647	0.642	0.645	0.643	0.057	0.042	0.046	0.787	0.689	0.692	0.689	0.027	0.018	0.017
	Average	**0.834**	**0.764**	**0.795**	**0.752**	**0.108**	**0.093**	**0.068**	**0.644**	**0.635**	**0.642**	**0.636**	**0.127**	**0.092**	**0.111**	**0.810**	**0.655**	**0.746**	**0.637**	**0.012**	**0.007**	**0.015**
LLM	Zero-Shot	0.768	0.728	0.717	0.765	0.262	0.179	0.169	0.642	0.641	0.661	0.655	0.236	0.149	0.182	0.779	0.557	0.644	0.557	0.087	0.194	0.124
	Random	0.775	0.717	0.710	0.734	0.145	0.079	0.065	0.591	0.580	0.584	0.581	0.213	0.196	0.193	0.644	0.593	0.605	0.648	0.016	0.028	0.019
	LF	0.703	0.662	0.664	0.710	0.210	0.155	0.142	0.516	0.512	0.513	0.513	0.250	0.243	0.227	0.670	0.595	0.608	0.632	0.031	0.039	0.038
	S1	0.780	0.719	0.715	0.732	0.156	0.096	0.080	0.640	0.631	0.638	0.632	0.015	0.026	0.028	0.711	0.617	0.615	0.634	0.025	0.033	0.030
	S2	0.771	0.708	0.709	0.725	0.152	0.076	0.070	0.596	0.593	0.593	0.594	0.070	0.055	0.072	0.628	0.562	0.585	0.610	0.025	0.033	0.032
	S3	0.797	0.727	0.729	0.726	0.128	0.057	0.048	0.634	0.627	0.631	0.629	0.183	0.158	0.207	0.713	0.575	0.577	0.576	0.080	0.096	0.089
	FCG-S1	0.817	0.686	0.809	0.663	0.088	0.028	0.026	0.638	0.588	0.665	0.611	0.156	0.222	0.148	0.639	0.588	0.605	0.647	0.014	0.017	0.018
	FCG-S2	0.804	0.737	0.740	0.739	0.220	0.196	0.158	0.651	0.613	0.672	0.627	0.105	0.093	0.076	0.660	0.606	0.614	0.658	0.018	0.021	0.024
	FCG-S3	0.829	0.725	0.807	0.697	0.132	0.136	0.089	0.646	0.605	0.669	0.621	0.054	0.052	0.029	0.688	0.590	0.587	0.604	0.087	0.103	0.096
	Average	**0.785**	**0.710**	**0.735**	**0.716**	**0.154**	**0.103**	**0.085**	**0.614**	**0.594**	**0.621**	**0.601**	**0.131**	**0.131**	**0.122**	**0.669**	**0.591**	**0.600**	**0.626**	**0.037**	**0.046**	**0.043**

Paradigm	Method	Adult-Race							Compas-Race							Credit-Age						
		Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓
ML	Default	0.849	0.784	0.808	0.768	0.098	0.077	0.057	0.646	0.641	0.644	0.641	0.287	0.236	0.257	0.813	0.648	0.757	0.628	0.053	0.030	0.047
	COT	0.847	0.783	0.803	0.769	0.055	0.040	0.025	0.643	0.640	0.641	0.641	0.080	0.058	0.054	0.812	0.648	0.754	0.629	0.026	0.015	0.033
	LTDD	0.848	0.783	0.805	0.767	0.057	0.099	0.065	0.650	0.643	0.647	0.643	0.097	0.115	0.114	0.812	0.645	0.755	0.626	0.029	0.014	0.044
	FairMask	0.849	0.783	0.807	0.767	0.092	0.060	0.047	0.639	0.635	0.637	0.636	0.192	0.151	0.161	0.813	0.650	0.756	0.629	0.052	0.028	0.043
	ADV	0.848	0.781	0.808	0.764	0.021	0.107	0.062	0.653	0.646	0.650	0.646	0.137	0.113	0.134	0.818	0.665	0.760	0.641	0.018	0.007	0.029
	MAAT	0.847	0.771	0.817	0.746	0.062	0.021	0.017	0.644	0.638	0.640	0.638	0.161	0.112	0.129	0.812	0.644	0.756	0.626	0.039	0.021	0.029
	MirrorFair	0.850	0.783	0.811	0.765	0.088	0.054	0.042	0.644	0.636	0.641	0.636	0.202	0.147	0.173	0.813	0.650	0.756	0.629	0.048	0.025	0.038
	EOP	0.841	0.769	0.797	0.752	0.061	0.046	0.027	0.605	0.594	0.600	0.595	0.024	0.022	0.030	0.804	0.642	0.722	0.624	0.027	0.018	0.027
	ROC	0.805	0.769	0.754	0.813	0.041	0.073	0.048	0.638	0.638	0.640	0.641	0.035	0.041	0.048	0.779	0.683	0.683	0.686	0.045	0.028	0.048
	Average	**0.842**	**0.778**	**0.800**	**0.768**	**0.060**	**0.062**	**0.042**	**0.640**	**0.634**	**0.637**	**0.634**	**0.116**	**0.095**	**0.105**	**0.808**	**0.653**	**0.743**	**0.636**	**0.035**	**0.019**	**0.036**
LLM	Zero-Shot	0.767	0.728	0.717	0.765	0.213	0.161	0.159	0.643	0.642	0.661	0.655	0.608	0.778	0.623	0.779	0.554	0.643	0.555	0.018	0.023	0.014
	Random	0.789	0.729	0.722	0.741	0.124	0.076	0.075	0.630	0.616	0.627	0.619	0.393	0.524	0.405	0.674	0.610	0.613	0.653	0.100	0.087	0.083
	LF	0.739	0.693	0.685	0.731	0.156	0.086	0.100	0.438	0.434	0.448	0.450	0.235	0.253	0.228	0.676	0.609	0.609	0.646	0.073	0.046	0.064
	S1	0.764	0.713	0.706	0.741	0.151	0.091	0.098	0.608	0.606	0.616	0.613	0.551	0.675	0.562	0.679	0.608	0.612	0.643	0.067	0.070	0.056
	S2	0.785	0.721	0.725	0.734	0.108	0.055	0.058	0.612	0.611	0.615	0.615	0.505	0.631	0.516	0.675	0.612	0.615	0.656	0.104	0.076	0.082
	S3	0.814	0.733	0.756	0.721	0.076	0.045	0.037	0.615	0.608	0.626	0.619	0.553	0.684	0.565	0.676	0.611	0.614	0.654	0.104	0.091	0.086
	FCG-S1	0.810	0.681	0.809	0.672	0.068	0.054	0.042	0.594	0.504	0.642	0.563	0.155	0.224	0.161	0.645	0.597	0.613	0.659	0.014	0.036	0.040
	FCG-S2	0.814	0.736	0.761	0.727	0.112	0.106	0.082	0.621	0.619	0.638	0.633	0.609	0.755	0.622	0.653	0.595	0.604	0.644	0.006	0.029	0.033
	FCG-S3	0.804	0.659	0.792	0.644	0.056	0.047	0.035	0.604	0.502	0.683	0.567	0.111	0.185	0.118	0.723	0.579	0.586	0.580	0.077	0.146	0.092
	Average	**0.790**	**0.708**	**0.744**	**0.714**	**0.106**	**0.070**	**0.066**	**0.590**	**0.563**	**0.612**	**0.585**	**0.389**	**0.491**	**0.397**	**0.675**	**0.603**	**0.608**	**0.642**	**0.068**	**0.073**	**0.067**

and Qwen3-Max-Thinking [12]. These models cover both open-source and closed-source architectures from leading AI vendors (OpenAI, Google, DeepSeek, Alibaba), which are widely adopted in real-world applications [28, 58]. We focus this validation on *Adult-Sex* and *Adult-Race* because Adult is one of the most widely used and representative benchmark datasets in fairness research [33, 41]. Table 3 reports the detailed results. For each LLM and each configuration, we average results over the eight mitigation methods to obtain the overall LLM results, and then compare them with the corresponding ML results from RQ1. We further assess statistical significance over the 56 displayed advanced-LLM comparisons (56 = 2 tasks ×4 LLMs ×7 metrics).

4.2.2 Results. The overall conclusion remains consistent with RQ1. Across both *Adult-Sex* and *Adult-Race*, traditional ML still outperforms the overall LLM averages on all three fairness metrics and all four predictive metrics. Specifically, ML reduces SPD, EOD, and AOD by 48.6%, 35.4%, and 48.9% on *Adult-Sex*, and by 60.3%, 38.6%, and 58.0% on *Adult-Race*, respectively. It also improves Accuracy, F1-score, Precision, and Recall by 8.9%, 6.4%, 9.1%, and 0.5% on *Adult-Sex*, and by 7.7%, 6.1%, 8.3%, and 0.9% on *Adult-Race*, respectively. This advantage is statistically significant in 46 of the 56 comparisons (82.1%). Therefore, even after replacing GPT-4o-mini with stronger models, the overall conclusion of RQ1 remains unchanged.

Table 3: (RQ2) Predictive performance and fairness results of LLM-based mitigation methods across four advanced LLMs (QW: Qwen3-Max-Thinking, GPT: GPT-5, DS: DeepSeek-v3.2, and GM: Gemini-2.5-Flash) on *Adult-Sex* and *Adult-Race*. Even with stronger LLMs, the overall conclusion remains unchanged.

Method	Acc. ↑				F1 ↑				Prec. ↑				Rec. ↑				SPD ↓				EOD ↓				AOD ↓			
	QW	GPT	DS	GM	QW	GPT	DS	GM	QW	GPT	DS	GM	QW	GPT	DS	GM	QW	GPT	DS	GM	QW	GPT	DS	GM	QW	GPT	DS	GM
Adult-Sex																												
Zero-Shot	0.824	0.833	0.750	0.777	0.762	0.771	0.712	0.743	0.764	0.777	0.704	0.732	0.759	0.766	0.760	0.796	0.250	0.262	0.445	0.290	0.289	0.316	0.492	0.142	0.210	0.227	0.414	0.166
Random	0.803	0.829	0.811	0.841	0.743	0.756	0.744	0.779	0.738	0.780	0.747	0.791	0.751	0.743	0.743	0.770	0.193	0.204	0.168	0.207	0.129	0.227	0.096	0.185	0.111	0.160	0.084	0.138
LF	0.618	0.391	0.465	0.646	0.602	0.386	0.463	0.610	0.651	0.510	0.586	0.636	0.696	0.502	0.596	0.666	0.325	0.178	0.171	0.302	0.246	0.129	0.132	0.277	0.254	0.154	0.137	0.261
S1	0.780	0.805	0.787	0.836	0.733	0.759	0.737	0.783	0.721	0.745	0.725	0.780	0.763	0.785	0.761	0.786	0.206	0.251	0.205	0.220	0.116	0.157	0.095	0.166	0.111	0.151	0.102	0.135
S2	0.758	0.775	0.775	0.829	0.722	0.743	0.727	0.782	0.715	0.734	0.717	0.771	0.774	0.801	0.757	0.799	0.271	0.345	0.198	0.271	0.160	0.249	0.090	0.198	0.167	0.243	0.097	0.176
S3	0.722	0.720	0.755	0.790	0.692	0.693	0.709	0.754	0.699	0.702	0.699	0.742	0.759	0.767	0.745	0.801	0.179	0.046	0.078	0.222	0.053	0.079	0.058	0.100	0.069	0.081	0.042	0.107
FCG-S1	0.818	0.833	0.801	0.836	0.746	0.732	0.727	0.777	0.758	0.814	0.749	0.782	0.740	0.708	0.729	0.775	0.125	0.140	0.125	0.220	0.069	0.106	0.016	0.166	0.052	0.079	0.030	0.138
FCG-S2	0.772	0.818	0.818	0.828	0.734	0.776	0.760	0.782	0.723	0.764	0.756	0.770	0.780	0.805	0.765	0.801	0.335	0.332	0.182	0.289	0.252	0.266	0.086	0.230	0.243	0.240	0.084	0.200
FCG-S3	0.795	0.827	0.803	0.828	0.746	0.767	0.731	0.779	0.737	0.769	0.737	0.770	0.772	0.766	0.729	0.792	0.218	0.163	0.119	0.239	0.147	0.074	0.089	0.167	0.130	0.066	0.062	0.145
Average	0.758	0.750	0.752	0.804	0.715	0.702	0.700	0.756	0.718	0.727	0.714	0.755	0.754	0.735	0.728	0.774	0.231	0.207	0.156	0.246	0.147	0.161	0.083	0.186	0.142	0.147	0.080	0.162
Overall	**0.766**				**0.718**				**0.729**				**0.748**				**0.210**				**0.144**				**0.133**			
Adult-Race																												
Zero-Shot	0.824	0.834	0.746	0.777	0.761	0.772	0.708	0.743	0.765	0.779	0.701	0.732	0.758	0.767	0.757	0.797	0.149	0.163	0.305	0.193	0.165	0.219	0.336	0.098	0.124	0.153	0.287	0.119
Random	0.819	0.827	0.813	0.842	0.749	0.753	0.747	0.780	0.760	0.777	0.752	0.792	0.741	0.741	0.745	0.771	0.130	0.141	0.122	0.125	0.139	0.189	0.096	0.117	0.104	0.130	0.082	0.089
LF	0.707	0.536	0.569	0.790	0.674	0.523	0.558	0.746	0.679	0.600	0.623	0.736	0.735	0.624	0.659	0.780	0.207	0.167	0.128	0.167	0.142	0.100	0.040	0.100	0.153	0.139	0.074	0.106
S1	0.745	0.797	0.757	0.831	0.707	0.754	0.715	0.778	0.710	0.747	0.714	0.774	0.757	0.785	0.758	0.784	0.163	0.174	0.163	0.140	0.069	0.107	0.067	0.111	0.094	0.113	0.094	0.094
S2	0.705	0.778	0.770	0.828	0.681	0.741	0.729	0.780	0.703	0.741	0.726	0.771	0.763	0.789	0.771	0.796	0.228	0.172	0.152	0.147	0.086	0.094	0.053	0.114	0.140	0.106	0.080	0.098
S3	0.725	0.796	0.784	0.830	0.700	0.757	0.738	0.784	0.710	0.746	0.728	0.773	0.779	0.798	0.768	0.802	0.185	0.134	0.120	0.139	0.054	0.060	0.028	0.075	0.099	0.071	0.050	0.076
FCG-S1	0.800	0.821	0.810	0.841	0.750	0.760	0.744	0.779	0.741	0.776	0.751	0.789	0.771	0.766	0.747	0.770	0.141	0.135	0.116	0.131	0.088	0.111	0.087	0.141	0.087	0.093	0.075	0.102
FCG-S2	0.768	0.828	0.812	0.830	0.728	0.767	0.747	0.780	0.725	0.779	0.753	0.772	0.774	0.769	0.751	0.793	0.255	0.159	0.133	0.151	0.242	0.182	0.094	0.112	0.217	0.135	0.087	0.100
FCG-S3	0.800	0.829	0.786	0.834	0.751	0.767	0.725	0.779	0.742	0.771	0.717	0.777	0.775	0.765	0.737	0.782	0.166	0.101	0.091	0.135	0.121	0.072	0.041	0.112	0.114	0.061	0.042	0.091
Average	0.759	0.776	0.763	0.828	0.718	0.728	0.713	0.776	0.721	0.742	0.720	0.773	0.762	0.755	0.742	0.785	0.184	0.148	0.128	0.142	0.118	0.114	0.063	0.110	0.126	0.106	0.073	0.095
Overall	**0.782**				**0.733**				**0.739**				**0.761**				**0.151**				**0.101**				**0.100**			

Another notable observation is that even the largest reasoning-oriented model in our evaluation does not show clear mitigation advantages over traditional ML. Although Qwen3-Max-Thinking is the largest model we study, it still lags behind ML. On *Adult-Sex*, ML reduces SPD, EOD, and AOD by 48.3%, 35.5%, and 47.7%, respectively, while achieving 8.9%, 6.0%, 7.8%, and 0.4% higher Accuracy, F1-score, Precision, and Recall. A similar pattern also appears on *Adult-Race*. This suggests that simply scaling up LLM size or reasoning capability is insufficient to address the challenges of tabular bias mitigation.

> **Ans. to RQ2:** Even when stronger advanced LLMs are considered, the overall conclusion remains unchanged. Traditional ML is significantly better in 82.1% of the comparisons, and the advanced LLMs do not overturn ML's overall advantage in either fairness or predictive performance. This shows that the advantage of the ML paradigm remains consistent across different advanced LLMs.

4.3 RQ3: How do evaluation settings affect the effectiveness of LLM-based methods?

4.3.1 Motivation and Methodology. RQ1 and RQ2 show that LLM-based mitigation methods underperform traditional ML-based methods under our evaluation setting, and that this conclusion remains the same across different LLMs. However, existing LLM-based bias mitigation studies [34, 41, 50] often report promising results. Furthermore, we find that these studies [34, 41, 50] typically conduct evaluations on artificially balanced test sets, where samples are evenly distributed across demographic groups and labels. In contrast, our evaluation follows real-world data distributions, which are inherently imbalanced. This observation leads us to hypothesize that evaluation settings, particularly the use of balanced versus imbalanced test sets, may substantially influence the observed fairness performance. Therefore, in this RQ, we systematically investigate the impact of evaluation settings on the effectiveness of LLM-based mitigation methods.

To answer this question, we focus on *Adult-Sex* and *Adult-Race* and compare the same LLM-based mitigation methods under two test-set settings. The **balanced-test setting** follows previous LLM evaluation settings and samples 512 test instances with balanced demographic and label proportions. The **random-test setting** samples 512 test instances uniformly from the Adult test set and thus preserves the natural distribution. Both test-set settings are repeated three times with different random seeds. To assess whether this overall pattern is statistically significant, we aggregate the raw results across the nine LLM-based methods and three repeated runs, giving 27 raw values for each fairness metric under each task. A difference is considered statistically significant when the resulting p-value is below 0.05, and the balanced-test setting is considered better when it yields a lower mean fairness value.

4.3.2 Results. Table 4 shows that the balanced-test setting usually reports better fairness than the random-test setting. Specifically, 77.8% of the method-level fairness comparisons numerically favor the balanced-test setting. The balanced-test setting is significantly better in 66.7% of the fairness comparisons. Averaged over the eight mitigation methods, SPD, EOD, and AOD increase by 138.4%, 49.7%,

Table 4: (RQ3) Comparison of LLM-based mitigation methods under the balanced-test setting and the random-test setting on Adult. The *balanced-test setting* uses balanced demographic and label proportions, while the *random-test setting* preserves the original test distribution. Each value is averaged over three runs.

Method	Adult-Sex														Adult-Race													
	Balanced							Random							Balanced							Random						
	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓
Zero-Shot	0.737	0.735	0.743	0.737	0.143	0.148	0.143	0.785	0.732	0.720	0.754	0.258	0.158	0.159	0.760	0.758	0.771	0.760	0.175	0.203	0.175	0.785	0.732	0.719	0.755	0.192	0.256	0.186
Random	0.745	0.741	0.761	0.745	0.081	0.112	0.083	0.793	0.727	0.722	0.733	0.152	0.042	0.054	0.763	0.762	0.769	0.763	0.037	0.076	0.068	0.788	0.731	0.721	0.748	0.121	0.220	0.147
LF	0.727	0.726	0.730	0.727	0.009	0.029	0.020	0.741	0.685	0.675	0.711	0.102	0.028	0.025	0.754	0.753	0.758	0.754	0.039	0.065	0.057	0.783	0.727	0.715	0.746	0.095	0.162	0.106
S1	0.757	0.757	0.757	0.757	0.035	0.039	0.043	0.749	0.702	0.692	0.738	0.132	0.060	0.054	0.740	0.739	0.745	0.740	0.043	0.081	0.074	0.776	0.721	0.710	0.742	0.104	0.067	0.067
S2	0.746	0.746	0.746	0.746	0.068	0.065	0.068	0.753	0.708	0.699	0.749	0.169	0.086	0.084	0.760	0.760	0.761	0.760	0.030	0.055	0.056	0.769	0.722	0.710	0.756	0.111	0.159	0.105
S3	0.744	0.742	0.751	0.744	0.043	0.047	0.059	0.796	0.735	0.727	0.747	0.165	0.031	0.054	0.729	0.723	0.752	0.729	0.060	0.107	0.065	0.803	0.733	0.734	0.731	0.114	0.201	0.119
FCG-S1	0.673	0.637	0.772	0.673	0.042	0.070	0.047	0.829	0.728	0.804	0.703	0.120	0.161	0.101	0.697	0.665	0.769	0.697	0.048	0.076	0.048	0.813	0.744	0.747	0.740	0.133	0.208	0.132
FCG-S2	0.741	0.733	0.771	0.741	0.159	0.206	0.159	0.810	0.745	0.750	0.747	0.215	0.151	0.132	0.739	0.735	0.755	0.739	0.046	0.078	0.074	0.784	0.721	0.713	0.733	0.137	0.198	0.133
FCG-S3	0.707	0.685	0.776	0.707	0.083	0.117	0.083	0.842	0.749	0.825	0.721	0.144	0.080	0.066	0.627	0.576	0.744	0.627	0.035	0.070	0.043	0.797	0.644	0.761	0.625	0.029	0.080	0.049
Average	**0.731**	**0.722**	**0.757**	**0.731**	**0.074**	**0.093**	**0.078**	**0.789**	**0.723**	**0.735**	**0.734**	**0.162**	**0.089**	**0.081**	**0.730**	**0.719**	**0.758**	**0.730**	**0.057**	**0.090**	**0.073**	**0.789**	**0.719**	**0.725**	**0.731**	**0.115**	**0.172**	**0.116**

and 36.2%, respectively, when evaluation shifts from the balanced-test setting to the random-test setting.

The balanced-test setting removes much of the demographic skew in the original data, which makes fairness disparities appear smaller than under the natural distribution. Once evaluation returns to the random-test setting, this apparent advantage weakens substantially. This helps explain why prior LLM studies can report more favorable fairness outcomes under balanced evaluation, while our results in RQ1 and RQ2 remain much less favorable under more realistic test distributions.

> **Ans. to RQ3:** Fairness in the LLM paradigm is highly sensitive to the test setting. Across Adult-Sex and Adult-Race, 77.8% of the method-level fairness comparisons numerically favor the balanced-test setting, and 66.7% of the fairness comparisons show statistically significant advantages for the balanced-test setting. Under the random-test setting, SPD, EOD, and AOD increase by 138.4%, 49.7%, and 36.2%, respectively. This suggests that the favorable fairness reported in previous LLM studies may rely on balanced evaluation and may not generalize to more realistic test distributions.

4.4 RQ4: Can fine-tuning on the full training data improve LLM-based methods?

4.4.1 Motivation and Methodology. Beyond the influence of test data, we further investigate whether the performance gap between LLM- and ML-based methods can be attributed to differences in training data utilization. Existing LLM-based bias mitigation methods [34, 50] primarily rely on in-context learning, which leverages only a small subset of training data as demonstrations, whereas ML-based methods are trained on the full dataset. This discrepancy raises the question of whether limited access to training data constrains the effectiveness of LLM-based approaches. To examine this, we move beyond in-context learning and instead adopt supervised fine-tuning, which allows LLMs to fully leverage the entire training dataset. Specifically, we fine-tune LLMs on the full training set and evaluate whether such broader data exposure can improve their effectiveness.

Based on this motivation, we study both regular fine-tuning and fine-tuning combined with traditional data-level pre-processing, and examine whether these gains are enough to challenge the traditional ML paradigm.

Fine-tuning-based strategies. We evaluate four fine-tuning-based strategies using supervised LLM fine-tuning. Unlike in-context learning, which relies on limited demonstrations without updating model parameters, fine-tuning adapts the LLM using the full training dataset.

- **Regular Fine-tuning.** The LLM is fine-tuned in a standard supervised manner using the original tabular training data without any fairness intervention.
- **Pre-processed Fine-tuning.** Traditional pre-processing bias mitigation techniques are first applied to the training data, after which the LLM is fine-tuned on the modified dataset. Specifically, we evaluate fine-tuning in combination with COT, LTDD, and FairMask to examine whether data-level debiasing can influence fairness outcomes in the fine-tuned LLM.

We evaluate these strategies on all six tasks using two lightweight and economical LLMs, GPT-4o-mini [4] and Qwen2.5-7B [6]. For each setting, we keep the train-test split fixed for fine-tuning and repeat evaluation three times on the same test set. Table 5 reports the detailed results. To assess statistical significance, we compare the raw outcomes of fine-tuning with the in-context learning results in RQ1 and with the traditional ML results in RQ1. Across all six tasks, this gives 24 predictive-performance comparisons and 18 fairness comparisons.

4.4.2 Results. **Compared with in-context learning.** Compared with the in-context learning results in RQ1, fine-tuning is more effective. Averaged over the pre-processed variants and the two lightweight models across all six tasks, SPD, EOD, and AOD decrease by 42.3%, 54.0%, and 50.4%, respectively. Fine-tuning is significantly better in 72.2% of fairness comparisons against the in-context learning results in RQ1. Meanwhile, Accuracy, F1-score, Precision, and Recall improve by 11.2%, 4.4%, 15.0%, and 0.8%, respectively, and fine-tuning is significantly better in 79.2% of predictive-performance comparisons. These results indicate that giving the model access to the full training set is much more effective than relying on demonstrations alone.

Table 5: (RQ4) Fine-tuning-based mitigation results. *Regular* denotes standard supervised fine-tuning on the original training data. *COT*, *LTDD*, and *FairMask* denote pre-processed fine-tuning, where the corresponding pre-processing method is applied before fine-tuning. Each value is averaged over three repeated tests.

Model	Method	Adult-Sex							Compas-Sex							Credit-Sex						
		Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓
GPT-4o-mini	Regular	0.871	0.814	0.844	0.794	0.168	0.055	0.055	0.675	0.671	0.671	0.670	0.224	0.199	0.183	0.821	0.674	0.766	0.648	0.013	0.006	0.017
	COT	0.871	0.815	0.841	0.796	0.161	0.031	0.041	0.672	0.656	0.675	0.658	0.043	0.010	0.014	0.812	0.652	0.747	0.630	0.008	0.034	0.020
	LTDD	0.787	0.568	0.856	0.574	0.017	0.069	0.035	0.666	0.647	0.671	0.650	0.090	0.048	0.050	0.814	0.663	0.746	0.640	0.008	0.031	0.021
	FairMask	0.867	0.803	0.847	0.777	0.133	0.016	0.024	0.666	0.653	0.666	0.653	0.148	0.132	0.112	0.821	0.679	0.759	0.654	0.016	0.024	0.016
	Average	0.849	0.750	0.847	0.735	0.120	0.043	0.039	0.670	0.657	0.671	0.658	0.126	0.097	0.090	0.817	0.667	0.755	0.643	0.011	0.024	0.018
Qwen2.5-7B	Regular	0.871	0.821	0.834	0.811	0.203	0.119	0.099	0.644	0.591	0.686	0.615	0.134	0.060	0.119	0.808	0.601	0.782	0.590	0.016	0.008	0.018
	COT	0.872	0.817	0.841	0.800	0.176	0.067	0.064	0.650	0.603	0.686	0.622	0.085	0.035	0.059	0.818	0.653	0.770	0.629	0.006	0.002	0.014
	LTDD	0.799	0.634	0.793	0.617	0.054	0.022	0.016	0.614	0.581	0.617	0.592	0.054	0.058	0.042	0.790	0.500	0.791	0.530	0.003	0.002	0.007
	FairMask	0.870	0.815	0.840	0.797	0.182	0.083	0.074	0.619	0.535	0.690	0.583	0.061	0.030	0.041	0.817	0.648	0.774	0.625	0.009	0.004	0.009
	Average	0.853	0.772	0.827	0.756	0.154	0.073	0.063	0.632	0.577	0.670	0.603	0.084	0.046	0.065	0.808	0.601	0.779	0.594	0.009	0.004	0.012
Overall		**0.851**	**0.761**	**0.837**	**0.746**	**0.137**	**0.058**	**0.051**	**0.651**	**0.617**	**0.670**	**0.630**	**0.105**	**0.071**	**0.078**	**0.813**	**0.634**	**0.767**	**0.618**	**0.010**	**0.014**	**0.015**

Model	Method	Adult-Race							Compas-Race							Credit-Age						
		Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓	Acc.↑	F1↑	Prec.↑	Rec.↑	SPD↓	EOD↓	AOD↓
GPT-4o-mini	Regular	0.871	0.814	0.844	0.794	0.093	0.059	0.043	0.675	0.670	0.671	0.670	0.432	0.618	0.448	0.821	0.673	0.766	0.647	0.057	0.025	0.051
	COT	0.870	0.814	0.840	0.796	0.093	0.052	0.040	0.673	0.664	0.671	0.663	0.117	0.051	0.079	0.821	0.666	0.771	0.640	0.053	0.023	0.048
	LTDD	0.834	0.754	0.792	0.732	0.058	0.021	0.019	0.572	0.554	0.631	0.598	0.087	0.152	0.110	0.804	0.586	0.775	0.579	0.043	0.014	0.085
	FairMask	0.857	0.767	0.872	0.730	0.052	0.008	0.007	0.674	0.667	0.671	0.667	0.209	0.156	0.170	0.822	0.674	0.768	0.648	0.059	0.026	0.055
	Average	0.858	0.787	0.837	0.763	0.074	0.035	0.028	0.649	0.639	0.661	0.650	0.211	0.244	0.202	0.817	0.650	0.770	0.629	0.053	0.022	0.060
Qwen2.5-7B	Regular	0.871	0.821	0.834	0.811	0.122	0.126	0.086	0.645	0.591	0.688	0.615	0.229	0.117	0.212	0.808	0.601	0.780	0.590	0.022	0.007	0.017
	COT	0.872	0.817	0.841	0.800	0.104	0.074	0.055	0.648	0.600	0.682	0.620	0.206	0.115	0.185	0.818	0.654	0.772	0.630	0.051	0.024	0.045
	LTDD	0.756	0.607	0.653	0.597	0.043	0.046	0.033	0.620	0.618	0.618	0.619	0.092	0.137	0.123	0.781	0.450	0.748	0.505	0.004	0.001	0.008
	FairMask	0.859	0.774	0.868	0.738	0.066	0.040	0.026	0.623	0.544	0.689	0.588	0.106	0.053	0.088	0.788	0.496	0.775	0.528	0.013	0.007	0.017
	Average	0.840	0.755	0.799	0.736	0.084	0.072	0.050	0.634	0.588	0.669	0.610	0.158	0.106	0.152	0.799	0.550	0.769	0.563	0.023	0.010	0.022
Overall		**0.849**	**0.771**	**0.818**	**0.750**	**0.079**	**0.053**	**0.039**	**0.641**	**0.613**	**0.665**	**0.630**	**0.185**	**0.175**	**0.177**	**0.808**	**0.600**	**0.769**	**0.596**	**0.038**	**0.016**	**0.041**

Effect of pre-processing. Pre-processing further improves fine-tuning. Compared with regular fine-tuning, the pre-processed variants reduce SPD, EOD, and AOD by 40.9%, 40.3%, and 41.0% on average for GPT-4o-mini, and by 39.7%, 39.0%, and 45.1% for Qwen2.5-7B. These improvements come with only modest drops in predictive performance, with Accuracy and F1-score decreasing by 2.1% and 5.2% for GPT-4o-mini, and by 2.4% and 6.1% for Qwen2.5-7B.

Compared with traditional ML. Relative to traditional ML, the gains from fine-tuning remain limited. Compared with the ML results in RQ1, the overall fine-tuning results achieve better mean values in 50.0% of predictive-performance comparisons and 61.1% of fairness comparisons. However, fine-tuning is significantly better in only 37.5% of predictive-performance comparisons and in none of the fairness comparisons. In addition, fine-tuning introduces extra monetary cost (on average $8.32 per task and per strategy) and training time (on average 1.10 hours per task and per strategy), which further limits its practical advantage over traditional ML in tabular bias mitigation. These results suggest that although fine-tuning strengthens the LLM paradigm, its advantage over traditional ML remains limited in both effectiveness and practical cost.

> **Ans. to RQ4:** Fine-tuning-based methods improve over in-context learning, and pre-processing further helps. However, gains over traditional ML remain limited. The overall fine-tuning results achieve better mean values in 61.1% of fairness comparisons and 50.0% of predictive-performance comparisons, but significant advantages appear in none of the fairness comparisons and in only 37.5% of predictive-performance comparisons. Fine-tuning also introduces extra monetary cost and training time. These results suggest that although fine-tuning strengthens the LLM paradigm, its advantage over traditional ML remains limited in both effectiveness and practical cost.

5 Implications

This section derives implications for future research and practice, based on our findings.

(1) LLMs are not a silver bullet. Although LLMs are currently receiving substantial attention, our findings (RQ1 and RQ2) show that, in tabular bias mitigation, traditional ML-based methods achieve stronger fairness and predictive performance than LLM-based methods. RQ4 further shows that, although fine-tuning on the full training data improves LLM-based methods, it still does not consistently outperform traditional ML-based mitigation. These suggest that LLMs should not be assumed to be a universal replacement for traditional bias mitigation techniques. For software engineers, this implies that adopting LLM-based methods should not be treated as a default upgrade in fairness-critical tabular applications; instead, method selection should remain evidence-driven and task-specific. More broadly, our findings highlight a general lesson for SE: the adoption of emerging paradigms such as LLMs should be guided

by empirical evidence rather than prevailing trends. Even highly capable foundation models may not uniformly outperform well-established techniques in domain-specific tasks, underscoring the importance of rigorous evaluation when integrating new paradigms into software systems.

(2) Real-world adoption requires evidence from realistic evaluation settings. Our findings from RQ3 show that the effectiveness of LLM-based bias mitigation methods is highly sensitive to evaluation settings. In particular, artificially balanced test distributions can substantially inflate fairness improvements, compared to evaluations under realistic, imbalanced data distributions, and may therefore create a misleading impression of real-world effectiveness. This has an important implication for both research and practice. For practitioners, fairness improvements demonstrated only under balanced test settings may not translate to real-world deployments, where class and demographic distributions are inherently skewed. Relying on such results may therefore lead to suboptimal or even misleading deployment decisions. For researchers, these results highlight that evaluation settings should be treated as an integral part of fairness claims. Methods intended for real-world use should be evaluated primarily under realistic data distributions, and results obtained under controlled or balanced settings should be interpreted with caution. More broadly, this finding underscores a general lesson for SE: the validity of empirical conclusions depends not only on the methods being evaluated, but also on how closely the evaluation setting reflects real-world operating conditions.

(3) Cross-paradigm evaluation is essential. Our study demonstrates that cross-paradigm comparisons are necessary to obtain a complete understanding of method effectiveness. Existing research on bias mitigation typically evaluates methods within a single paradigm (e.g., ML-based or LLM-based), which may lead to an incomplete or overly optimistic view of their effectiveness. By directly comparing ML- and LLM-based methods in a unified experimental setting, we uncover limitations that are not apparent in within-paradigm evaluations, including the relative underperformance of LLM-based methods (RQ1) and their sensitivity to evaluation settings (RQ3). For researchers, this suggests that evaluations confined to a single paradigm may provide an incomplete or even misleading picture of method effectiveness. Future work should therefore incorporate cross-paradigm comparisons to better position new approaches against established alternatives. More broadly, this finding points to a general lesson for SE: meaningful evaluation requires not only strong baselines, but also comparisons across fundamentally different solution paradigms.

6 Threats to Validity

This section discusses potential threats to the validity of our empirical study.

Selection of datasets. The choice of datasets may threaten the validity of our results. To mitigate this threat, we use three widely studied real-world datasets spanning multiple high-stakes domains and a diverse range of sensitive attributes [24, 25, 33, 44]. This setup provides broad coverage of application settings that are central to fairness research and enables a controlled comparison between the ML and LLM paradigms.

Selection of mitigation methods. Our results may also be affected by the mitigation methods included in the study. To reduce this threat, we include representative traditional ML methods spanning pre-processing, in-processing, post-processing, and ensemble learning, as well as representative LLM-based mitigation methods under in-context learning, which are widely adopted in prior studies [23, 33, 34, 38, 44, 50, 55]. For RQ4, we additionally construct fine-tuning-based methods to examine whether exposing LLMs to the full training data changes the overall conclusion. These methods cover the main stages of bias mitigation in tabular classification.

Selection of models. Another threat lies in the choice of models. To mitigate this threat, we evaluate four widely used ML classifiers and use GPT-4o-mini as the primary under-test LLM. We also validate the main findings on four additional advanced LLMs in RQ2 and examine fine-tuning in RQ4 with two lightweight and economical LLMs. This design reduces the risk that our findings are tied to a single model family and allows us to assess whether the main conclusion remains stable across different model choices.

Selection of evaluation settings and metrics. The evaluation settings and metrics used in this study may also influence the results. To alleviate this threat, we align datasets, metrics, and evaluation settings across paradigms, and we explicitly compare balanced-test and random-test settings in RQ3. In addition, we adopt widely used [22, 23, 33] metrics for both predictive performance and fairness, and report both types of metrics instead of relying on a single criterion.

Data leakage in LLMs. The use of both closed-source and open-source LLMs introduces a threat related to potential data leakage from public data sources [32, 37, 53]. To mitigate this threat, we evaluate multiple advanced LLMs in RQ2 and report repeated results rather than single runs. Although we cannot fully rule out data leakage, such leakage would be more likely to favor the LLM paradigm than disadvantage it. Despite this possibility, the zero-shot and few-shot LLM results in our study still remain clearly below traditional ML in many settings.

7 Conclusion

This paper presents a large-scale empirical comparison of state-of-the-art ML- and LLM-based bias mitigation methods for tabular classification. Our results show that traditional ML-based methods consistently outperform advanced LLM-based methods in both predictive performance and fairness, while the favorable fairness reported in prior LLM-based studies is highly sensitive to evaluation settings. Although supervised fine-tuning improves over in-context learning, it does not consistently outperform traditional ML-based methods. These findings suggest that LLM-based methods should not be treated as a default choice for fairness-critical tabular applications. They also indicate that the advantages of LLM-based methods over traditional approaches may not be consistent in broader SE tasks. More broadly, our study highlights that adopting emerging techniques in SE requires rigorous, cross-paradigm evaluation.

8 Data Availability

We have publicly released the scripts, data, and our analysis results as a replication package [13].

References

[1] 1994. The Credit Dataset. https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients.

[2] 2016. The Compas dataset. https://github.com/propublica/compas-analysis.

[3] 2017. The Adult Census Income dataset. https://archive.ics.uci.edu/ml/datasets/adult.

[4] 2024. GPT-4o mini. https://platform.openai.com/docs/models/gpt-4o-mini.

[5] 2024. IBM AIF360. https://ai-fairness-360.org/.

[6] 2024. Qwen2.5-7B-Instruct. https://huggingface.co/Qwen/Qwen2.5-7B-Instruct.

[7] 2025. DeepSeek-v3.2. https://huggingface.co/deepseek-ai/DeepSeek-V3.2.

[8] 2025. Gemini-2.5-Flash. https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash.

[9] 2025. GPT-5. https://developers.openai.com/api/docs/models/gpt-5-chat-latest.

[10] 2026. openai. https://developers.openai.com/api/reference/overview.

[11] 2026. openrouter. https://openrouter.ai/.

[12] 2026. Qwen3-Max-Thinking. https://openrouter.ai/qwen/qwen3-max-thinking.

[13] 2026. Replication package. https://doi.org/10.5281/zenodo.19244975.

[14] Razieh Alidoosti. 2021. Ethics-driven software architecture decision making. In *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 90–91.

[15] Fatma Başak Aydemir and Fabiano Dalpiaz. 2018. A roadmap for ethics-aware software engineering. In *Proceedings of the international workshop on software fairness*. 15–21.

[16] Sumon Biswas and Hridesh Rajan. 2020. Do the machine learning models on a crowd sourced platform exhibit bias? an empirical study on model fairness. In *Proceedings of the 28th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*. 642–653.

[17] Sumon Biswas and Hridesh Rajan. 2021. Fair preprocessing: towards understanding compositional fairness of data transformers in machine learning pipeline. In *Proceedings of the 29th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*. 981–993.

[18] Yuriy Brun and Alexandra Meliou. 2018. Software fairness. In *Proceedings of the 2018 26th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*. 754–759.

[19] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. 2021. Bias in machine learning software: Why? how? what to do?. In *Proceedings of the 29th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*. 429–440.

[20] Joymallya Chakraborty, Suvodeep Majumder, Zhe Yu, and Tim Menzies. 2020. Fairway: a way to build fair ML software. In *Proceedings of the 28th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*. 654–665.

[21] Evan Chen, Run-Jun Zhan, Yan-Bai Lin, and Hung-Hsuan Chen. 2025. More Women, Same Stereotypes: Unpacking the Gender Bias Paradox in Large Language Models. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 4639–4643.

[22] Zhenpeng Chen, Xinyue Li, Jie M. Zhang, Federica Sarro, and Yang Liu. 2025. Diversity Drives Fairness: Ensemble of Higher Order Mutants for Intersectional Fairness of Machine Learning Software. In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering*. 743–755.

[23] Zhenpeng Chen, Xinyue Li, Jie M Zhang, Weisong Sun, Ying Xiao, Tianlin Li, Yiling Lou, and Yang Liu. 2025. Software Fairness Dilemma: Is Bias Mitigation a Zero-Sum Game? *Proceedings of the ACM on Software Engineering* 2, FSE (2025), 1780–1801.

[24] Zhenpeng Chen, Jie M Zhang, Max Hort, Mark Harman, and Federica Sarro. 2024. Fairness testing: A comprehensive survey and analysis of trends. *ACM Transactions on Software Engineering and Methodology* 33, 5 (2024), 1–59.

[25] Zhenpeng Chen, Jie M Zhang, Federica Sarro, and Mark Harman. 2022. MAAT: a novel ensemble approach to addressing fairness and performance bugs for machine learning software. In *Proceedings of the 30th ACM joint european software engineering conference and symposium on the foundations of software engineering*. 1122–1134.

[26] Zhenpeng Chen, Jie M Zhang, Federica Sarro, and Mark Harman. 2023. A comprehensive empirical study of bias mitigation methods for machine learning classifiers. *ACM transactions on software engineering and methodology* 32, 4 (2023), 1–30.

[27] Valeriia Cherepanova, Chia-Jung Lee, Nil-Jana Akpinar, Riccardo Fogliato, Martin Bertran Lopez, Michael Kearns, and James Zou. 2025. Improving llm group fairness on tabular data via in-context learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, Vol. 8. 579–590.

[28] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Proceedings of the Forty-first International Conference on Machine Learning, ICML 2024*.

[29] Michael E Donohue. 2018. A replacement for Justitia's scales: Machine learning's role in sentencing. *Harv. JL & Tech.* 32 (2018), 657.

[30] Atmika Gorti, Aman Chadha, and Manas Gaur. 2024. Unboxing occupational bias: Debiasing llms with us labor data. In *Proceedings of the AAAI Symposium Series*, Vol. 4. 48–55.

[31] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).

[32] Naila Shafirni Hidayat, Muhammad Dehan Al Kautsar, Alfan Farizki Wicaksono, and Fajri Koto. 2025. Simulating training data leakage in multiple-choice benchmarks for llm evaluation. In *Proceedings of the 5th Workshop on Evaluation and Comparison of NLP Systems*. 21–39.

[33] Max Hort, Zhenpeng Chen, Jie M Zhang, Mark Harman, and Federica Sarro. 2024. Bias mitigation for machine learning classifiers: A comprehensive survey. *ACM Journal on Responsible Computing* 1, 2 (2024), 1–52.

[34] Jingyu Hu, Weiru Liu, and Mengnan Du. 2024. Strategic demonstration selection for improved fairness in llm in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 7460–7475.

[35] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems* 33, 1 (2012), 1–33.

[36] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. 2012. Decision theory for discrimination-aware classification. In *2012 IEEE 12th international conference on data mining*. IEEE, 924–929.

[37] Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024. An open-source data contamination report for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 528–541.

[38] Yanhui Li, Linghan Meng, Lin Chen, Li Yu, Di Wu, Yuming Zhou, and Baowen Xu. 2022. Training data debugging for the fairness of machine learning software. In *Proceedings of the 44th International Conference on Software Engineering*. 2215–2227.

[39] Junwei Liu, Kaixin Wang, Yixuan Chen, Xin Peng, Zhenpeng Chen, Lingming Zhang, and Yiling Lou. 2026. Large Language Model-Based Agents for Software Engineering: A Survey. *ACM Transactions on Software Engineering and Methodology* (2026).

[40] Yang Liu and Chenhui Chu. 2025. Do LLMs Align Human Values Regarding Social Biases? Judging and Explaining Social Biases with LLMs. *arXiv preprint arXiv:2509.13869* (2025).

[41] Yanchen Liu, Srishti Gautam, Jiaqi Ma, and Himabindu Lakkaraju. 2024. Confronting LLMs with traditional ML: Rethinking the fairness of large language models in tabular classifications. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 3603–3620.

[42] Ali A Mahmoud, Tahani AL Shawabkeh, Walid A Salameh, and Ibrahim Al Amro. 2019. Performance predicting in hiring process and performance appraisals using machine learning. In *2019 10th international conference on information and communication systems (ICICS)*. IEEE, 110–115.

[43] Henry B Mann and Donald R Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics* (1947), 50–60.

[44] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)* 54, 6 (2021), 1–35.

[45] Kewen Peng, Joymallya Chakraborty, and Tim Menzies. 2022. Fairmask: Better fairness via model-based rebalancing of protected attributes. *IEEE Transactions on Software Engineering* 49, 4 (2022), 2426–2439.

[46] Data Protection. 2018. General data protection regulation. *Intersoft Consulting, Accessed in October* 24, 1 (2018).

[47] Omar Shaikh, Hongxin Zhang, William Held, Michael Bernstein, and Diyi Yang. 2023. On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 4454–4470.

[48] Yining She, Sumon Biswas, Christian Kästner, and Eunsuk Kang. 2025. FairSense: Long-Term Fairness Analysis of ML-Enabled Systems. In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering*. 782–794.

[49] Ezekiel Soremekun, Mike Papadakis, Maxime Cordy, and Yves Le Traon. 2025. Software fairness: An analysis and survey. *Comput. Surveys* 58, 3 (2025), 1–38.

[50] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2023. DecodingTrust: A Comprehensive Assessment of Trustworthiness in {GPT} Models. (2023).

[51] Michael Wick, Jean-Baptiste Tristan, et al. 2019. Unlocking fairness: a trade-off revisited. *Advances in neural information processing systems* 32 (2019).

[52] Noam Wies, Yoav Levine, and Amnon Shashua. 2023. The learnability of in-context learning. *Advances in Neural Information Processing Systems* 36 (2023), 36637–36651.

[53] Yonghao Wu, Zheng Li, Jie M Zhang, and Yong Liu. 2024. Condefects: A complementary dataset to address the data leakage concern for llm-based fault localization and program repair. In *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*. 642–646.

[54] Ying Xiao, Shangwen Wang, Sicen Liu, Dingyuan Xue, Xian Zhan, Yepang Liu, and Jie M Zhang. 2025. Fairness Is Not Just Ethical: Performance Trade-Off

via Data Correlation Tuning to Mitigate Bias in ML Software. *arXiv preprint arXiv:2512.21348* (2025).

[55] Ying Xiao, Jie M Zhang, Yepang Liu, Mohammad Reza Mousavi, Sicen Liu, and Dingyuan Xue. 2024. MirrorFair: Fixing fairness bugs in machine learning software via counterfactual predictions. *Proceedings of the ACM on Software Engineering* 1, FSE (2024), 2121–2143.

[56] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 335–340.

[57] Jie M Zhang and Mark Harman. 2021. " Ignorance and Prejudice" in Software Fairness. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 1436–1447.

[58] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR* abs/2303.18223 (2023).