Sailing in the Location-Based Fairness-Bias Sphere

Erhu He[†]
University of Pittsburgh
erh108@pitt.edu

Yiqun Xie[†]
University of Maryland
xie@umd.edu

Xiaowei Jia
University of Pittsburgh
xiaowei@pitt.edu

Weiye Chen
University of Maryland
weiyec@umd.edu

Han Bao, Xun Zhou
University of Iowa
{han-bao,xun-zhou}@uiowa.edu

Zhe Jiang
University of Florida
zhe.jiang@ufl.edu

Rahul Ghosh
University of Minnesota
ghosh128@umn.edu

Praveen Ravirathinam
University of Minnesota
pravirat@umn.edu

ABSTRACT

As the adoption of machine learning continues to thrive, fairness of the algorithms has become a key factor determining their long-term success and sustainability. Among them, location-based fairness – or spatial fairness – is critical for a variety of essential societal applications that commonly rely on spatial data, including agriculture, disaster response, urban planning, etc. Spatial biases incurred by learning, if left unattended, may cause or exacerbate unfair distribution of resources, spatial disparity, social division, etc. However, very limited understanding has been developed on location-based fairness and bias in machine learning. Compared to traditional fairness-preserving techniques, the spatial consideration introduces two major layers of complication: (1) Space is continuous with no well-defined categories (e.g., categories by race or gender); and (2) Categorizations given by space-partitionings are known to be subject to high statistical sensitivity (e.g., gerrymandering). Under these challenges, we formally explore and demonstrate the fragility of learning methods in the spatial fairness-bias sphere. Specifically, we present a set of techniques that can maneuver the training process towards various targeted fairness-bias outcomes, while maintaining the same level of overall prediction performance (i.e., for "free"). Extensive experiments are carried out on two real-world problems: crop monitoring in the US and palm oil plantation mapping in Indonesia. The results demonstrate the effectiveness of the manipulation algorithms and the importance of explicitly regulating location-based fairness using a diverse set of criteria.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Information systems** → **Spatial-temporal systems**.

[†]Equal contribution. Corresponding author: Yiqun Xie.

SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9529-8/22/11…$15.00
https://doi.org/10.1145/3557915.3560976

KEYWORDS

Fairness, location, deep learning

ACM Reference Format:
Erhu He, Yiqun Xie, Xiaowei Jia, Weiye Chen, Han Bao, Xun Zhou, Zhe Jiang, Rahul Ghosh, and Praveen Ravirathinam. 2022. Sailing in the Location-Based Fairness-Bias Sphere. In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3557915.3560976

1 INTRODUCTION

As the adoption of machine learning continues to thrive and inspire major interests across broad applications (e.g., driving assistance or automation, face recognition, healthcare), fairness in the data-driven algorithms has drawn serious attention and becomes a key factor for the sustained success in the long term.

This paper focuses on location-based fairness (a.k.a., spatial fairness), which is critical for a variety of essential societal applications, where location information is heavily used in decision and policy-making. Spatial biases incurred by learning, if left unattended, may cause or exacerbate unfair distribution of resources, social division, spatial disparity, etc. In agriculture, for example, population growth has caused immense pressure on food production and supply across the globe, which is worsened by climate change and its consequences (e.g., extreme events and frequent disturbances). The pressure has resulted in multiple initiatives in large-scale crop monitoring, including NASA Harvest and G20's GEOGLAM global agriculture monitoring [1]. As the size of the satellite imagery that these types of projects commonly rely on is reaching far beyond the capacity of manual processing, they heavily rely on learning methods to assist the generation of crop maps [12, 15]. Major derived products such as acreage estimates [21] are further used to inform critical actions such as the distribution of subsidies [3, 4, 18] and other resources, to allow resilience against disturbances and long-term sustainability. However, existing monitoring frameworks largely ignored fairness issues, including location-related fairness. To illustrate the potential implications, Fig. 1 shows the spatial distributions of the F1 scores achieved by a deep learning model for real-world cotton classification using satellite imagery. The study area has a size of 80km by 80km, and is partitioned into

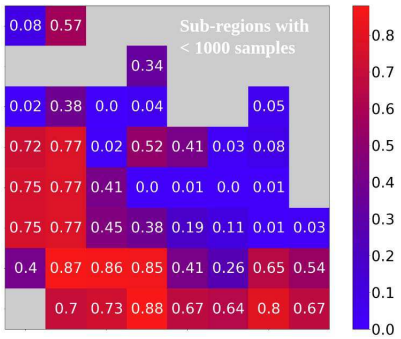


Figure 1: Examples of spatial bias on cotton classification.

10km by 10km local regions. As we can see, the results across locations present clear spatial bias with large differences between local F1-scores (the patterns of spatial bias may differ between models trained from two separate runs). Such unattended bias in crop mapping can lead to unfair resource distribution. For example, it can hurt small holders representing the main production force behind minor crops [6, 26, 27]. Similarly, maps generated by machine learning have been increasingly applied in other critical decision-making processes, where ignorance of location-based bias can cause unfair estimations in disaster or insurance management (e.g., real-time satellite-based maps of floods, damages, or risks), unfair allocation of essential resources to the population in poverty (e.g., urban slums in Africa), unfair carbon tax, and many more.

Fairness is a relatively new topic in machine learning but has been widely studied in recent years given its importance. However, existing fairness-preserving techniques have largely focused on problems where fairness can be well-defined on classes in certain categorical attributes such as race, gender, or income level.

Related work has explored a variety of techniques. The most common and generally-applicable strategy is regularization-based approach, which includes additional fairness-related losses during the training process [13, 23, 31, 33]. Another major direction of methods aims to learn group-invariant features [2], in which additional discriminators are included in the training to penalize learned features that can reveal the identity of a group (e.g., gender) in an adversarial manner. Sensitive category de-correlation also employs the adversarial learning regime. However, instead of learning group-invariant features, it tries to learn features that do not lead to polarization of predictions (e.g., the sentiment of a phrase) for each category (e.g., a language) [2, 25, 34]. From the data perspective, strategies have also been developed for data collection and filtering to reduce bias in downstream learning tasks [11, 24, 32]. More variations have also been discussed in a recent survey [16]. These methods have been applied to tasks where groups are well-defined by categorical attributes (e.g., face detection [23], text analysis [25], online bidding [19]). For spatial data, location-explicit frameworks [28, 30] have been developed to improve prediction performance over locations, but they do not consider fairness.

Compared to the traditional fairness-preserving techniques designed for categorical groups, evaluation and enforcement of spatial fairness introduce two major layers of complication. First, space is continuous with no well-defined categories (e.g., categories by race or gender). Second, location groups are commonly created by space-partitioning. However, statistics evaluated from groups or categories given by space-partitionings are known to be sensitive to changes in partitionings. In other words, a result map determined to be fair on one partitioning can easily get an opposite conclusion from another. In statistics, this is known as the Modifiable Areal Unit Problem (MAUP; Def. 2), which shows the fragility of statistical conclusions under the manipulation of partitionings. The lack of consideration of MAUP has created major public concerns. A high-profile example is gerrymandering, which refers to the partitioning-manipulation practice used by political parties to gain favor during an election. The growing concerns have raised the issue to the US Supreme Court in 2019 [20] and state courts [9].

With the MAUP challenge in mind, we propose a set of techniques to maneuver in the sphere of location-based fairness and bias during the training process. Our goal is to explicitly manipulate the fairness/bias of a learning model to reach a variety of targeted outcomes (e.g., fair, biased, controlled mix with multiple fairness criteria), all operating under the condition to maintain the same level of overall performance (e.g., global F1-score) as that of a base model (e.g., an LSTM trained with no fairness consideration). Our techniques demonstrate the fragility of fairness in the spatial setting and the feasibility of manipulating the results with high degrees of freedom, revealing the importance of explicit and thorough consideration of location-based fairness in learning.

We carry out extensive experiments on two real-world problems: crop monitoring in US and palm oil plantation mapping in Indonesia. For each problem, we evaluate the proposed fairness-bias maneuvering techniques on top of two base neural network architectures, i.e., densely-connected neural network (DNN) and recurrent LSTM. The results confirm the effectiveness of the proposed algorithms under a variety of objectives and constraints.

2 PROBLEM

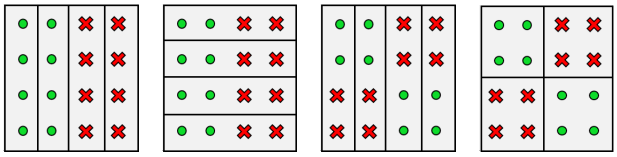
We first introduce the basic concepts for location-based fairness, and then discuss the general formulation of the fairness-aware learning problem. Finally, we present key instances of the problem, representing different learning outcomes in the fairness-bias sphere.

2.1 Concepts

Definition 1. ***Partitioning $\mathcal{P}$ and partition p.*** *A partitioning $\mathcal{P}$ splits an input space into K non-overlapping partitions $\{p_1, ..., p_K\}$ that together cover the entire space.*

Definition 2. ***Modifiable Areal Unit Problem (MAUP).*** *MAUP states that statistical results and conclusions are sensitive to the choice of space partitioning $\mathcal{P}$. Specifically, given a statistic τ that aggregates information inside a partition p, MAUP entails that the distribution of τ or conclusions based on it varies as $\mathcal{P}$ changes. This is often considered as a dilemma as statistical results are expected to vary if different aggregations or groupings of locations are used.*

Statistical sensitivity by MAUP has been commonly exploited in practice, including examples of gerrymandering [9, 20]. In the context of this work, MAUP means that the conclusion on "fair vs. biased" is fragile to variations in $\mathcal{P}$. For example, if F1-score or error rate is used as τ, then one can easily manipulate the fairness result by altering the partitioning as shown in Fig. 2; here different partitionings lead to opposite conclusions in fairness evaluation



(a) Dist. 1: Unfair (b) Dist. 1: Fair (c) Dist. 2: Fair (d) Dist. 2: Unfair

Figure 2: Fairness on two example distributions Dist. 1 and 2. (green: correct predictions; red: wrong predictions).

(e.g., (a) is unfair as two partitions have 100% accuracy whereas the other two have 0%, and (b) is fair as all are at 50%).

Definition 3. ***Performance measure*** $m_{\mathcal{F}}$**.** *A measure that evaluates the solution quality (not related to fairness) of a trained model $\mathcal{F}_{\Theta}$ with parameters Θ. For example, $m_{\mathcal{F}}$ can be F1-score (or a loss function during training), mean squared errors, etc. In the rest of the paper, $m_{\mathcal{F}}(\mathcal{F}_{\Theta})$ is used to denote the general performance of $\mathcal{F}_{\Theta}$, and $m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p)$ or $m_{\mathcal{F}}(\mathcal{F}_{\Theta}, \mathcal{P})$ specifically denotes the performance of $\mathcal{F}_{\Theta}$ on data samples in space covered by a partition $p \in \mathcal{P}$ or an entire partitioning $\mathcal{P}$ (equivalent to the entire dataset in this case).*

Definition 4. ***Fairness measure*** m_{fair}**.** *A statistic used to evaluate the fairness of a learning model's performance across several mutually-exclusive groups of individual locations. An example m_{fair} is the variance of F1-scores across groups. In this paper, groups are defined by partitions $p \in \mathcal{P}$, and m_{fair} we use is:*

$$m_{fair}(\mathcal{F}_{\Theta}, m_{\mathcal{F}}, \mathcal{P}) = \sum_{i=1}^{K} \frac{|m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p_i) - m_{\mathcal{F}}(\mathcal{F}_{\Theta}, \mathcal{P})|}{K} \quad (1)$$

where $\mathcal{F}_{\Theta}$ is a learning model (e.g., a deep network) with parameters Θ; K is the number of partitions $p \in \mathcal{P}$; $m_{\mathcal{F}}(\mathcal{F}_{\Theta}, \mathcal{P})$ represents the global performance across all partitions, which is equivalent to the expectation $E_{p \in \mathcal{P}}(m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p))$.

Definition 5. ***MAUP-aware fairness measure*** M_{fair}**.** *A fairness measure that explicitly considers multiple partitionings $\{\mathcal{P}\}$ during evaluation, which can be defined as:*

$$M_{fair}(\mathcal{F}_{\Theta}, m_{\mathcal{F}}, \{\mathcal{P}\}) = \frac{1}{|\{\mathcal{P}\}|} \cdot \sum_{i=1}^{|\{\mathcal{P}\}|} m_{fair}(\mathcal{F}_{\Theta}, m_{\mathcal{F}}, \mathcal{P}_i) \quad (2)$$

where $|\{\mathcal{P}\}|$ is the cardinality of partitionings used for MAUP-aware fairness evaluation.

2.2 General formulation

The inputs to the problem, in general, follow a typical learning formulation: features $\mathbf{X}$ and labels $\mathbf{y}$, split into training, validation and test sets (detailed in Sec. 4). The main difference here is that we additionally include a machine learning model $\mathcal{F}$ of user's choice, as well as its parameters Θ_0, which are trained without considering any fairness criterion. Θ_0 is important as it sets the expected level of the overall model performance $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})$ such as F1-scores in the free-training scenario, which can be used as a reference point for fairness-engaged training on $\mathcal{F}$.

The output is a trained model $\mathcal{F}$ with parameters Θ that maintains the same level of overall performance $m_{\mathcal{F}}(\mathcal{F}_{\Theta})$ as $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})$ (e.g., $|m_{\mathcal{F}}(\mathcal{F}_{\Theta}) - m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})| \leq \alpha$), while aiming for one or more of the following fairness-bias objectives:

- Fairness criteria C_{fair}: These criteria can be evaluated based on results from a MAUP-aware fairness measure M_{fair}. The goal is to improve location-based fairness.
- Bias injection C_{bias}: To understand location-based fairness in the learning context, it is important to know if, where, and how bias may be included in the model under different fairness conditions. In practice, bias may exist due to various reasons including malicious acts, manipulation, fairness-unaware training (e.g., Fig. 1), etc. The representation of bias is more diverse than fairness, i.e., a fair model can be made unfair in multiple different ways. For example, C_{bias} can be represented by a high M_{fair} value or a single partition $p \in \mathcal{P}$ that has a low solution quality $m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p)$.

While C_{fair} and C_{bias} appear to be on the opposite sides of an objective, they are not necessarily in conflict with each other and can co-exist in, or be co-expressed by, a trained model for the location-based fairness-bias problem as we will discuss next.

Finally, the scope of this problem focuses on the model-provider side, and the scenario where manipulation often cannot be done on the test data (e.g., in agriculture monitoring, satellite imagery is often published by trusted sources).

2.3 Key instances

In the following, we present three key instances of the general problem formulation in Sec. 2.2.

2.3.1 Pure fairness-driven learning. This instance focuses only on fairness-related objectives defined by the MAUP-aware fairness measure M_{fair}:

$$\min_{\Theta} M_{fair}(\mathcal{F}_{\Theta}, m_{\mathcal{F}}, S_{\mathcal{P}}),\ s.t.\ |m_{\mathcal{F}}(\mathcal{F}_{\Theta}) - m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})| \leq \alpha \quad (3)$$

where $S_{\mathcal{P}} = \{\mathcal{P}\}$ is a set of user-selected partitionings used for MAUP-aware fairness training, Θ_0 is the set of parameters trained without fairness consideration (Sec. 2.2), $m_{\mathcal{F}}(\mathcal{F}_{\Theta})$ and $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})$ evaluate the global model performance on the entire data (during training, we use validation data as a proxy of test data), and $\alpha \in \mathbb{R}^{+}$.

2.3.2 Pure bias-injection learning. Opposite to the previous instance, this aims to purely inject bias into a model. Here we consider two different forms of bias injection: (1) A high M_{fair} value on a target partitioning $\mathcal{P}$ (here $|\{\mathcal{P}\}| = 1$ for M_{fair}, making M_{fair} equivalent to its special case m_{fair}); and (2) A low model performance $m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p)$ on one specific partition $p \in \mathcal{P}$. The two forms are shown in Eq. (4).

$$\begin{aligned} &\max_{\Theta} M_{fair}(\mathcal{F}_{\Theta}, m_{\mathcal{F}}, \mathcal{P}) \textbf{ or } \min_{\Theta} m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p) \\ &s.t.\ |m_{\mathcal{F}}(\mathcal{F}_{\Theta}) - m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})| \leq \alpha \end{aligned} \quad (4)$$

2.3.3 False fairness-preserving learning. The first two instances are relatively easier for training as they have a pure objective, either fairness- or bias-based. This instance deals with a more complex

Location information (i.e., geo-coordinates of data samples) is not included as part of the features $\mathbf{X}$ for two reasons: (1) Locations are sensitive attributes in location-based fairness-aware learning, and a model should not treat a sample differently because of sensitive attributes. Removal of sensitive attributes is a standard practice in fair learning [7, 8, 14, 17]. and (2) Including geo-coordinates as features will make the trained model not applicable at a different spatial region, with or without fairness consideration. This is because different regions tend to have different ranges of geo-coordinates and rely on different coordinate systems/projections.

scenario, which hides biases under a seemingly fair model:

$$\min_{\Theta} \begin{bmatrix} \beta^{bias} \\ \beta^{fair} \end{bmatrix}^T \left(\begin{bmatrix} -M_{fair}(\mathcal{F}_\Theta, m_\mathcal{F}, \mathcal{P}^{bias}) \\ M_{fair}(\mathcal{F}_\Theta, m_\mathcal{F}, S^{fair}_\mathcal{P}) \end{bmatrix} \textbf{ or } \begin{bmatrix} m_\mathcal{F}(\mathcal{F}_\Theta, p^{bias}) \\ M_{fair}(\mathcal{F}_\Theta, m_\mathcal{F}, S^{fair}_\mathcal{P}) \end{bmatrix} \right)$$
$$s.t.\ |m_\mathcal{F}(\mathcal{F}_\Theta) - m_\mathcal{F}(\mathcal{F}_{\Theta_0})| \leq \alpha$$
$$\mathcal{P}^{bias} \notin S^{fair}_\mathcal{P} \tag{5}$$

As we can see, the objective includes both a fairness objective from Eq. (3) and a bias-injection objective from Eq. (4); again, the bias can be expressed in the same two forms in Eq. (4). For the first form of bias (partitioning-level), we do need an additional constraint, which requires that $\mathcal{P}^{bias}$ is not a member of $S^{fair}_\mathcal{P}$. Finally, weights β^{fair} and β^{bias} are used to combine the objectives; in this work, we set β^{bias} to 1 and β^{fair} to $|S^{fair}_\mathcal{P}|$. If biases can be injected under the coverage of the fairness objectives, it can become much more challenging to recognize or detect them in practice. Thus, it is important to understand the interactions to design more robust mechanisms to avoid the bias risks.

3 METHOD

The scope of methods in this paper focuses on general deep learning models (i.e., model-agnostic to deep networks).

3.1 Preliminaries: SPAD-based Training

As discussed in Def. 2 and 5, the statistical sensitivity caused by the MAUP problem needs to be explicitly considered when incorporating location-based fairness in the training process. Thus, we adopt the SPace-As-a-Distribution (SPAD) representation and bi-level training framework from [29] as our base framework. Note that [29] does not consider issues related to bias-injection. Here we briefly summarize the key components of the SPAD framework.

3.1.1 SPAD representation and stochastic training. As statistical conclusions from a single partitioning can hardly remain unchanged on different space partitionings $\mathcal{P}$, SPAD considers a set or distribution of partitionings $S_\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, ...\}$ and uses the aggregated fairness scores across all $\mathcal{P} \in S_\mathcal{P}$ as the final score. Without loss of generality, in this paper, the MAUP-aware fairness measure M_{fair} in Def. 5 is an implementation of the SPAD-based measure, where each $\mathcal{P}$ in the collection is given the same weight in the aggregation.

As it is computationally expensive to evaluate the fairness scores using all the partitionings in each iteration, the training process uses a stochastic strategy in which each epoch only considers one random sample of $\mathcal{P}$ from the collection. This allows a significant reduction of training time while keeping the performance of the model similar and sometimes better because of the improved ability to jump out of local minima without the averaging effects [29]. Following the recommendations from SPAD, all fairness-engaged training epochs start from a base model that is trained without any fairness consideration. This prevents the model from enforcing fairness at a premature stage (e.g., low global accuracy) that constrains its prediction quality. This is the $\mathcal{F}_{\Theta_0}$ in Sec. 2.2.

3.1.2 Bi-level training. A common training strategy in fairness-aware learning is to add the fairness score as a regularization term to the overall loss function [13, 31]. However, this suffers from three major limitations in the spatial fairness context: (1) The samples from each training batch are often not representative of all partitions $p \in \mathcal{P}$ and lead to inaccurate estimations of the fairness measure m_{fair} (Def. 5. This is different from traditional fairness evaluation where there is only a small number of groups to consider (e.g., genders); (2) Exact measures such as F1-scores cannot be used to measure fairness during training as they are not differentiable, and the approximation further decreases the quality of fairness evaluation; and (3) Additional hyper-parameters are needed to combine the loss. As a result, the regularization-based approach often has unsatisfying outcomes, which will be shown in Sec. 4.

In the bi-level strategy [29], the loss function remains unchanged (i.e., no addition of fairness-based regularization term) during the training phase, concentrating on prediction performance $m_\mathcal{F}$. The fairness is enforced by a referee, which is used at the beginning of each epoch to set the learning rates for data samples in different partitions $p \in \mathcal{P}$ (an epoch uses one random sample of $\mathcal{P}$ as discussed in Sec. 3.1.1) by evaluating the current level of bias across the partitions $p \in \mathcal{P}$. Intuitively, partitions p with higher-than-expected performance (e.g., a positive $m_\mathcal{F}(\mathcal{F}_\Theta, p_i) - m_\mathcal{F}(\mathcal{F}_\Theta, \mathcal{P})$; see Eq. (2)) will be assigned with lower learning rates, while p not meeting expectations will be given higher rates:

$$\eta_i = \frac{\Delta_i - \Delta_{min}}{\Delta_{max} - \Delta_{min}} \cdot \eta_{max} \tag{6}$$

where $\Delta_i = m_\mathcal{F}(\mathcal{F}_\Theta, p_i) - m_\mathcal{F}(\mathcal{F}_\Theta, \mathcal{P})$, and $p_i \in \mathcal{P}$.

Since the fairness evaluation assignment is performed at the beginning of each epoch, it can use representative samples from all partitions $p \in \mathcal{P}$. Moreover, as the fairness evaluation is used to assign learning rates rather than calculating gradients, fairness can be calculated directly with exact performance measures. Finally, there is no need for an extra hyper-parameter as there is no regularization term.

3.2 Fairness-preserving learning

Pure fairness-preserving learning (Sec. 2.3.1) can be achieved by applying the training strategies from Sec. 3.1. Since it is in general difficult to apply hard constraints during the back-propagation process, in our solution we model the constraints in Eqs. (3) to (5) as soft constraints. In order to minimize the deviation from the overall performance (e.g., global F1-score) achieved by the unconstrained model $\mathcal{F}_{\Theta_0}$ (no fairness consideration), we use the following tactics to keep the training of $\mathcal{F}_\Theta$ maneuvering around $m_\mathcal{F}(\mathcal{F}_{\Theta_0})$.

First, when evaluating the fairness result on a single partitioning $\mathcal{P}$ using m_{fair} in Eq. (1), we substitute $m_\mathcal{F}(\mathcal{F}_\Theta, \mathcal{P})$ with $m_\mathcal{F}(\mathcal{F}_{\Theta_0}, \mathcal{P})$. This allows m_{fair} to not only balance the performance on different partitions $p \in \mathcal{P}$, but also encourage their performances to converge towards $m_\mathcal{F}(\mathcal{F}_{\Theta_0}, \mathcal{P})$, which helps keep the overall performance at a similar level. Together with this substitution, we initiate the parameters for $\mathcal{F}_\Theta$ using Θ_0 as a guidance.

Second, we introduce a new performance-reconditioning epoch:

Definition 6. *A performance-reconditioning epoch temporarily ignores the fairness (or bias) criteria and focuses only on overall performance $m_\mathcal{F}$, as a mitigation strategy to move closer to $m_\mathcal{F}(\mathcal{F}_{\Theta_0}, \mathcal{P})$. In this context, this means the learning rates will be the same for all partitions $p \in \mathcal{P}$. One performance-reconditioning epoch is executed whenever the constraint $|m_\mathcal{F}(\mathcal{F}_\Theta) - m_\mathcal{F}(\mathcal{F}_{\Theta_0})| \leq \alpha$ is violated.*

Validation of the constraint in Def. 6 is performed by evaluation on the training dataset with exact metrics (e.g., F1-score instead of approximation by loss functions). The evaluation is delayed for t_{wait} epochs (e.g., $t_{wait} = 5$ in the paper) if the condition is met to save computation, and otherwise performed immediately after each epoch so that more execution of the reconditioning epoch may be used to maneuver back to a similar level of overall performance. The algorithm is summarized in Alg. 1.

Algorithm 1 Fairness-preserving learning

Require:
- Choice of architecture $\mathcal{F}$ and Θ_0 (trained without fairness)
- Set of partitionings $S_{\mathcal{P}}$
- Feature $\mathbf{X}$ and label $\mathbf{y}$
- Learning rate bound η_{max}

1: init($\mathcal{F}_{\Theta}$, $\Theta = \Theta_0$)
2: $t_{wait} = 0$
3: **while** not done **do**
4: $\mathcal{P}$ = get_one_partitioning($S_{\mathcal{P}}$)
5: S_η = get_fairness_learning_rates($\mathcal{F}_{\Theta}$, $\mathcal{F}_{\Theta_0}$, $\mathcal{P}$, $m_{\mathcal{F}}$, η_{max})
6: **for** (p, η_i) in ($\mathcal{P}$, S_η) **do**
7: **for** ($\mathbf{X}_{batch}$, $\mathbf{y}_{batch}$) in ($\mathbf{X}_p$, $\mathbf{y}_p$) **do**
8: $\Theta = \Theta - \eta_i \cdot \nabla \mathcal{L}_{\mathcal{F}_{\Theta}}(\mathbf{X}_{batch}, \mathbf{y}_{batch})$
9: **end for**
10: **end for**
11: **if** $t_{wait} \leq 0$ **then**
12: $m_{\mathcal{F}}(\mathcal{F}_{\Theta})$ = eval($\mathcal{F}_{\Theta}$, $\mathbf{X}$, $\mathbf{y}$)
13: **if** $|m_{\mathcal{F}}(\mathcal{F}_{\Theta}) - m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})| \leq \alpha$ **then**
14: $t_{wait} = 5$
15: **else**
16: exec_reconditioning_epoch($\mathcal{F}_{\Theta}$, $\mathbf{X}$, $\mathbf{y}$)
17: $t_{wait} = 0$
18: **end if**
19: **end if**
20: $t_{wait} = t_{wait} - 1$
21: **end while**
22: **return** $\mathcal{F}_{\Theta}$

3.3 Bias-injection learning

3.3.1 Partitioning-level bias injection. Pure bias-injection learning for a target partitioning, i.e., $\max_{\Theta} M_{fair}(\mathcal{F}_{\Theta}, m_{\mathcal{F}}, \mathcal{P})$ in Eq. (4) can adopt the same high-level training process in Alg. 1. The key difference is that the learning rates will be assigned in a different way. Instead of pushing performances on different partitions $p \in \mathcal{P}$ towards $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0}, \mathcal{P})$, here their discrepancies are increased by only providing a learning rate (η_{max}) to partitions with performances above $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})$. As allocating positive learning rates to partitions with lower performances may narrow their distances to $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})$, we set their rates to 0 in the bias-injection epochs.

In practice, the reconditioning epoch is often not activated much during pure fairness-preserving learning. However, it becomes important when Alg. 1 is applied to bias-injection. The main reason is that partitions with higher $m_{\mathcal{F}}(\mathcal{F}_{\Theta})$, in general, have less space for further growth. This is different from the scenario in pure fairness-preserving learning, where the training process tries to increase the $m_{\mathcal{F}}(\mathcal{F}_{\Theta})$ on lower-performing partitions. On the other hand, during bias-injection, lower-performing partitions that are not assigned learning rates often have a faster decrease in $m_{\mathcal{F}}(\mathcal{F}_{\Theta})$. Combined together, this makes it easy for $m_{\mathcal{F}}(\mathcal{F}_{\Theta}) - m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})$ to be smaller than $-\alpha$. Thus, having the new reconditioning epoch in Def. 6 is necessary during bias-injection, which was not considered in Sec. 3.1. Based on our experiments, the reconditioning epoch is often executed for more than 50% of the epochs (lines 4-10 in Alg. 1).

3.3.2 Partition-level bias injection. As discussed in Sec. 2.3.2, partition-level bias-injection targets performance decrease on only a single partition $p \in \mathcal{P}$, i.e., $\min_{\Theta} m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p)$ in Eq. (4). The constraint $|m_{\mathcal{F}}(\mathcal{F}_{\Theta}) - m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})| \leq \alpha$ is the same. We further discuss two related scenarios for the single partition (p) level:

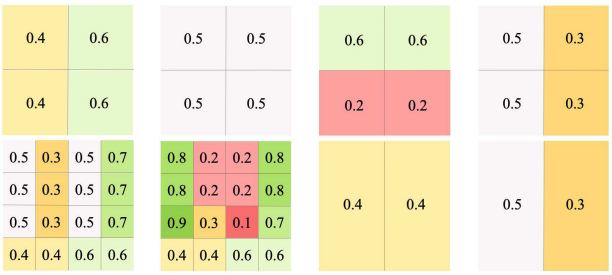
- Uncontrolled decrease on $m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p)$, where the only bias-injection purpose is to reduce the performance on p;
- Controlled decrease on $m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p)$, where the prediction is manipulated towards a user-specified target (e.g., from "oil palm plantation area" to "forest").

The training strategy for both scenarios can be embarrassingly simple. For the uncontrolled scenario, we can simply leave out data samples from the partition during training. One may also apply more aggressive tactics such as gradient ascent, i.e., $\Theta = \Theta + \eta \cdot \nabla \mathcal{L}_{\mathcal{F}_{\Theta}}(\mathbf{X}_p, \mathbf{y}_p)$, but based on our experiments the left-out strategy is self-sufficient in most scenarios. For the controlled scenario, we swap the training labels in p to the target labels. Note that for both scenarios, the reconditioning epoch is still needed to keep the model performance at the level of $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})$. Moreover, currently we only target on partitions with relatively small sizes (e.g., <10% of the entire study area), and this may not be a feasible problem with major changes in labels. For example, depending on the original $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})$, a certain proportion of change may result in a bounded performance of $m_{\mathcal{F}}(\mathcal{F}_{\Theta})$ which is below $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0}) - \alpha$. In the future, we will explore strategies to control only a learned/optimized subset of labels to inject location-based bias.

3.4 False fairness-preserving learning

In this section, we target the final problem defined in Sec. 2.3.3, where the goal is to simultaneously preserve fairness and inject bias during the training process. Such manipulations in opposite directions are often infeasible for traditional fairness problems, where the groups (e.g., race, gender) are pre-defined. In the location-based fairness problem, due to the existence of non-stationary groupings (i.e., different partitionings), we will show that it is possible for a model to have "hidden bias" under the cover of "fair results", which may be more easily unnoticed or undetected in practice.

3.4.1 Partitioning-level. Compared to the first two problems with pure objectives, false fairness-preserving learning is much more challenging due to the conflicts that often exist between the objectives in $\min_{\Theta}[\beta^{bias} \cdot (-M_{fair}(\mathcal{F}_{\Theta}, m_{\mathcal{F}}, \mathcal{P}^{bias})) + \beta^{fair} \cdot M_{fair}(\mathcal{F}_{\Theta}, m_{\mathcal{F}}, S_{\mathcal{P}}^{fair})]$; as a reminder, lower M_{fair} values correspond to fairer results. Although we have included an additional constraint that $\mathcal{P}^{bias} \notin S_{\mathcal{P}}^{fair}$, different partitionings are not independent and they often share certain level of overlaps. For this reason, the attempt to directly combine the training strategies in Sec. 3.2 and 3.3.1, as we tested, often gets stuck in a middle ground with little progress either on $S_{\mathcal{P}}^{fair}$ or $\mathcal{P}^{bias}$.



(a) Global F1: 0.5 (b) Global F1: 0.5 (c) Global F1: 0.4 (d) Global F1: 0.4

Figure 3: Examples showing the feasibility of improving fairness on one partitioning while injecting bias in another.

Thus, we propose an Agreement-driven simultaneous Fairness-preserving And Bias-injection (A-FAB) training approach to achieve the two goals for the same model $\mathcal{F}_{\Theta}$. In the following, we first demonstrate the feasibility of the task and then present the A-FAB algorithm.

Feasibility: Fig. 3 shows two illustrative examples of changes in performance distributions, which make results in one partitioning fairer while the other more biased. The grids represent different examples of space-partitionings, and the numbers in the partitions show the accuracy values achieved by a model. For simplicity of illustration, we assume each partition has an equal number of data samples. The first example consists of Fig. 3 (a) and (b), where all four partitionings share the same overall performance (i.e., global accuracy at 0.5). The changes from (a) to (b) make the location-based fairness improve (perfectly fair) for the partitionings at the top. However, they introduce more bias into the partitionings in the second row, i.e., the values move further away from the global mean at 0.5. Fig. 3 (c) and (d) show the second example, where similar patterns appear when changes are made from (c) to (d). Similarly, all partitionings share the same global accuracy at 0.4. In both cases, the fairness results get better after the change for the partitionings in the top row, but deteriorate for the partitionings at the bottom. The two examples demonstrate that it is feasible to simultaneously incur improvements and decreases on fairness.

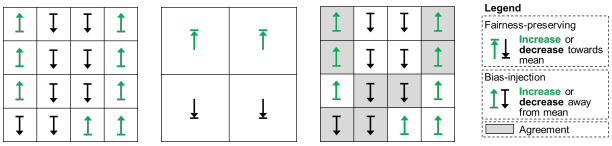
A-FAB algorithm: To realize the feasible scenarios in Fig. 3, the A-FAB training process executes in a paired-fashion, where each pair $(\mathcal{P}^{fair}, \mathcal{P}^{bias})$ is a combination of a partitioning in $S_{\mathcal{P}}^{fair}$ (the goal is to improve fairness for partitionings in the set) and the target partitioning for bias-injection $\mathcal{P}^{bias}$. The general sequence of training is that each epoch uses one pair from the set, and continues to loop over it till convergence.

The key step during the training of each pair $(\mathcal{P}^{fair}, \mathcal{P}^{bias})$ is to identify agreement between them. Specifically, A-FAB uses directional agreement (Def. 7 and 8) to determine whether a partition should be trained in the current epoch.

Definition 7. ***Desired direction.*** *A desired direction of performance change for a partition $p \in \mathcal{P}$ is the direction that moves its performance $m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p)$ in order to help the model to improve its objective function value. The directions are different for fairness preservation and bias-injection. For fairness preservation, a desired direction of p will be to increase if its score is below the global mean $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0}, \mathcal{P})$, and to decrease if its score is above the mean, which helps reduce M_{fair}:*

$$dir^{fair}(p) = \begin{cases} \uparrow \text{ or } 1, & \text{if } m_{\mathcal{F}}(\mathcal{F}_{\Theta}, p) \leq m_{\mathcal{F}}(\mathcal{F}_{\Theta_0}, \mathcal{P}) \\ \downarrow \text{ or } -1, & \text{otherwise} \end{cases} \quad (7)$$

For bias-injection, the directions are the opposite in order to increase M_{fair}.



(a) $\mathcal{P}^{bias}$ directions (b) $S_{\mathcal{P}}^{fair}$ directions (c) Agreement

Figure 4: Illustrative example of directional agreement.

Definition 8. ***Directional agreement.*** *Given two overlapping partitions $p^{fair} \in \mathcal{P}^{fair}$ and $p^{bias} \in \mathcal{P}^{bias}$, a directional agreement between them means that their desired directions of performance change are identical for the current epoch. Note that the directional agreements vary over epochs due to the continued updates on model parameters.*

Fig. 4 shows an example of directional agreement between a pair $(\mathcal{P}^{fair}, \mathcal{P}^{bias})$ in an epoch. Directional agreement is important as it identifies common grounds between two seemingly "conflicting" objectives. For the training of each epoch, we only carry out training on partitions from $\mathcal{P}^{bias}$ (or $\mathcal{P}^{fair}$) that agreed on the directions of intersecting partitions from $\mathcal{P}^{fair}$ (or $\mathcal{P}^{bias}$). The partitioning to choose partitions from is determined based on the average number of overlapping partitions $\bar{O}$, e.g., $\bar{O}^{fair} = |\mathcal{P}^{fair}|^{-1} \sum_{i=0}^{|\mathcal{P}^{fair}|} |p_i^{fair} \cap \mathcal{P}^{bias}|$. The partitioning with the smaller $\bar{O}$ will be selected.

If a partition p overlaps with multiple partitions in the other partitioning, we use the majority vote to determine if it will be included in training or not (ties are broken in favor of "agreement"). The reconditioning epoch is also employed here to maintain overall performance. The A-FAB algorithm is illustrated in Alg. 2.

3.4.2 Partition-level. The solution at the partition-level is much simpler. The training process is a combination of Alg. 1 for fairness-preserving learning in Sec. 3.2 and the partition-level bias-injection learning in Sec. 3.3.2. Specifically, for the uncontrolled bias-injection, we perform Alg. 1 as regular, and the only difference is that, the intersection between any partitioning $\mathcal{P}^{fair} \in S_{\mathcal{P}}^{fair}$ and the single partition p^{bias} is skipped in training. For the controlled bias-injection (altering prediction labels), instead of skipping the samples in p^{bias} for training, we use manipulated samples with label changes for the training of the partition, following the strategy in Sec. 3.3.2.

4 EXPERIMENTS

4.1 Dataset description

California crop mapping: Accurate mapping of crops is critical for estimating crop areas and yield, which are often used for distributing subsidies and providing farm insurance over space. Our input

Algorithm 2 A-FAB algorithm

Require:
- Architecture $\mathcal{F}$ and Θ_0 (no fairness consideration)
- Set of partitionings $S_{\mathcal{P}}^{fair}$ for fairness preservation
- Target partitioning $\mathcal{P}^{bias}$ for fairness injection
- Feature $\mathbf{X}$ and label $\mathbf{y}$

1: init($\mathcal{F}_\Theta$, $\Theta = \Theta_0$)
2: $pairs$ = construct_partitioning_pairs($S_{\mathcal{P}}^{fair}$, $\mathcal{P}^{bias}$)
3: **while** not done **do**
4: **for** ($\mathcal{P}^{fair}$, $\mathcal{P}^{bias}$) in $pairs$ **do**
5: $m_{\mathcal{F}}^{fair}$, $m_{\mathcal{F}}^{bias}$ = eval_partition_level($\mathcal{F}_\Theta$, $\mathbf{X}$, $\mathbf{y}$, $\mathcal{P}^{fair}$, $\mathcal{P}^{bias}$)
6: d^{fair}, d^{bias} = get_directions($m_{\mathcal{F}}^{fair}$, $m_{\mathcal{F}}^{bias}$, $m_{\mathcal{F}}(\mathcal{F}_{\Theta_0})$)
7: S_p^{agree} = get_agreement(d^{fair}, d^{bias})
8: $\mathbf{X}^{agree}$, $\mathbf{y}^{agree}$ = get_training_data($\mathbf{X}$, $\mathbf{y}$, S_p^{agree})
9: **for** ($\mathbf{X}_{batch}$, $\mathbf{y}_{batch}$) in ($\mathbf{X}^{agree}$, $\mathbf{y}^{agree}$) **do**
10: $\Theta = \Theta - \eta \cdot \nabla \mathcal{L}_{\mathcal{F}_\Theta}(\mathbf{X}_{batch}, \mathbf{y}_{batch})$
11: **end for**
12: **end for**
{Comment: Skipped the code for the reconditioning epoch, which is the same as lines 11-20 in Alg. 1.}
13: **end while**
14: **return** $\mathcal{F}_\Theta$

$\mathbf{X}$ for crop and land cover classification is the multi-spectral remote sensing data from Sentinel-2 in Central Valley, California, and the study region has a size of 4096×4096 (~6711 km^2 at 20m resolution). We use the multi-spectral data captured in August, 2018 for the mapping, and each location has reflectance values from 10 spectral bands, which are used as input features. The label $\mathbf{y}$ is from the USDA Crop Data Layer (CDL) [5].

Mapping palm oil plantations in Indonesia: We also validate our framework in detecting oil palm plantations, which is a key driver for deforestation in Indonesia. Plantations have similar greenness levels to tropical forests. Our ground truth labels are created in Kalimantan, Indonesia in 2014 based on manually created plantation mapping products RSPO [10] and Tree Plantation [22]. Each location is labeled as one of the categories from {plantation, non-plantation, unknown}, where the "unknown" class represents the locations with inconsistent labels between the RSPO and Tree Plantation dataset. We do not consider the "unknown" class in the classification. We utilize the 500-meter resolution multi-spectral MODIS satellite image, which consists of 7 reflectance bands (620-2155 nm) collected by MODIS instruments onboard NASA's satellites, and is collected in January, 2014.

For both problems, we randomly select 20%, 20%, and 60% locations for training, validation and testing, respectively.

4.2 Candidate methods

- **Base:** The base deep learning model (fully-connected DNN and LSTM) without consideration of spatial fairness.
- **REG:** Regularization is a main strategy used for general fairness-aware learning [13, 23, 31, 33]. Following this strategy, spatial fairness is enforced using the base model with a regularization term to the loss function. This regularization encourages the fairness for the partitionings under protection and the bias for the partitionings under intervention. As F1-score is not differentiable, we use standard approximation via the threshold-based approach, which amplifies softmax predictions $\hat{y}$ over a threshold γ to 1 to suppresses others to 0 using $1 - \text{ReLU}(1 - A \cdot \text{ReLU}(\hat{y} - \gamma))$, where A is a sufficiently large number ($A = 10000$ in our tests). The weight of the regularizer is set to 10.
- **Adversarial Discriminating-based learning (ADL):** This baseline is an extension of the discriminator-based fairness enforcing approach [2]. We include a separate discriminator for each partitioning in $S_{\mathcal{P}}^{fair}$, and $\mathcal{P}^{bias}$. For fairness preservation, the model aims to learn group-invariant (or fair) features that make it difficult for a discriminator to identify the partition $p \in \mathcal{P}$ from which data samples come. For bias-injection, we do the opposite to reward features that are in favor of the discriminator.
- **FAIR:** The proposed pure fairness-preserving learning approach described in Sec. 3.2.
- **BI:** The proposed pure bias-injection learning approach described in Sec. 3.3. There are three variants of BI: (1) BI$_{\mathcal{P}}$: partitioning-level bias-injection; (2) BI$_p$: partition-level bias-injection (without label control); and (3) BI$_p^*$: partition-level bias-injection with target label control.
- **A-FAB:** The proposed method that simultaneously performs fairness-preservation and bias-injection in Sec. 3.4. Similarly, depending on the type of bias-injection, it has three variants A-FAB$_{\mathcal{P}}$, A-FAB$_p$ and A-FAB$_p^*$.

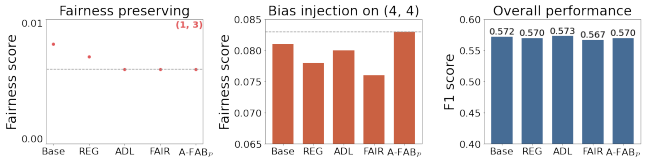
4.3 Implementation details

The training algorithms developed in this paper do not require specific types of space-partitionings (i.e., the information is not used by any decisions in the algorithm). Without loss of generality, our experiment uses the common $m \times n$-type of partitionings that have m rows and n columns and equal-size cells, because of their intuitiveness and simplicity for notation. Specifically, in the following, we use "(m, n)" to denote a $m \times n$ partitioning. We did not use existing partitionings (e.g., voting districts grouped from census blocks) as they are known to contain bias (e.g., debated at both the federal and state courts [9, 20]).

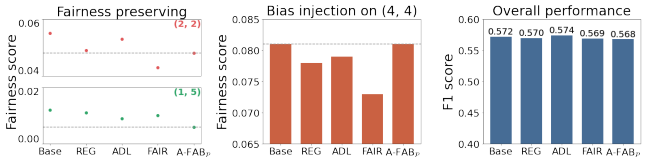
Our proposed methods do not assume specific network architectures. Most results presented in this paper use an 8-layer deep neural network (DNN) as a base model. We also test an LSTM model using a series of remote sensing images for classification. These models take inputs of multi-spectral data at each location and output the land cover label. In our experiment, we first train an initial DNN or LSTM base model for 300 epochs (converged) without considering the fairness, using Adam ($\alpha = 0.001$) as the optimizer. From this base model, we further implement different candidate approaches to improve fairness or inject bias. We train the model for 200 epochs for each partitioning for fairness preservation or bias injection.

4.4 Results

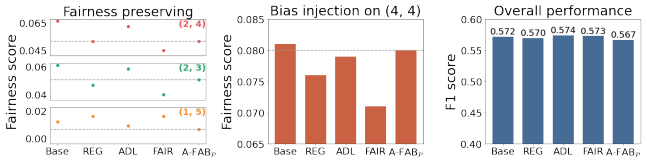
We evaluate the proposed method using randomly selected partitionings for fairness-preservation and bias-injection. Here we report the performance of each method by injecting bias into the partitioning (4,4) while preserving fairness over different sets of partitionings (Fig. 5). The sets cover different numbers of partitionings, i.e., from 1 to 4, as shown in Fig. 5 (a) to (d). We have also tested the performance over



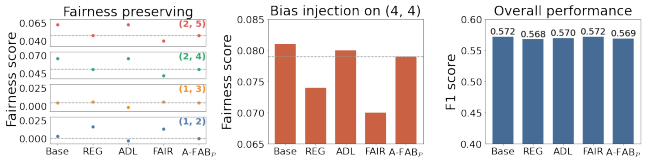
(a) Fairness preserving (1,3), bias injection (4,4)



(b) Fairness preserving (2,2) and (1,5), bias injection (4,4)

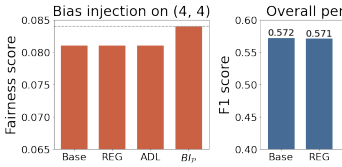


(c) Fairness preserving (2,4), (2,3), and (1,5), bias injection (4,4)

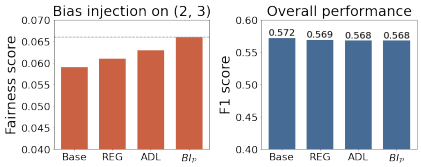


(d) Fairness preserving (2,5), (2,4) (1,3), and (1,2), bias injection (4,4)

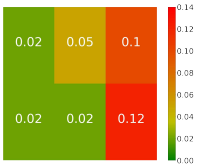
Figure 5: The fairness and overall performance with a different number of fairness-preserving partitionings (from 1 to 4). For each test, we show three results for all the methods: left - obtained fairness scores (m_{fair} in Def. 2) for each of fairness-preserving partitionings, middle - the obtained fairness scores for the bias-injecting partitioning, and right - the overall performance. The higher fairness score indicates worse fairness performance.



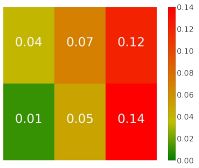
(a) Bias injection (4,4)



(b) Bias injection (2,3)



(c) Base on (2,3)



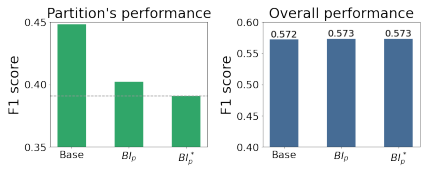
(d) BI$_\mathcal{P}$ on (2,3)

Figure 6: (a)(b) The fairness and overall predictive performance on the target partition (4,4) or (2,3) after applying bias injection. (c)(d) The obtained F-1 scores over different partitions in (2,3) using (c) Base and (d) BI$_\mathcal{P}$.

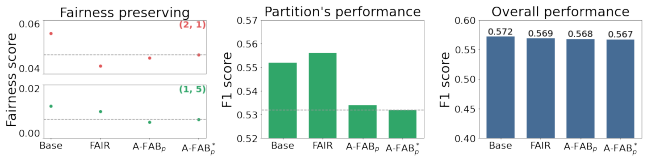
other random partitionings and observed similar performance (see our code repository: https://drive.google.com/drive/folders/1KawFcoJV_xZtifsZmYBLS__uo1P-Vg9w?usp=sharing).

4.4.1 Overall performance. In the crop mapping dataset, we can observe several major trends according to the results. First, our proposed methods FAIR, BI and A-FAB are able to maintain similar global F-1 scores as the other methods. This confirms the capacity of the training strategies in controlling the results in the fairness-bias sphere (i.e., improving or degrading the fairness) without compromising the overall classification performance, revealing the importance of explicit and thorough consideration of location-based fairness in important applications.

4.4.2 Fairness-preserving performance. Second, the proposed methods FAIR and A-FAB in general produce much better fairness scores (lower the better; Def. 4) for partitionings under fairness-protection compared to the base model, REG and ADL. This confirms the effectiveness of our method in maintaining the fairness by using the learning-rate-based strategy (i.e., improved sample representativeness). Especially, the fairness scores obtained by A-FAB are similar to the FAIR method, which confirms that A-FAB can simultaneously preserve the fairness for certain partitionings while injecting bias for a target partitioning, thanks to the use of directional agreements. The ADL method focuses on reducing the distributional gap across partitions. As a result, it treats different partitions equally in the classification process by eliminating partition-specific information, but it is not as effective for the enforcement of fairness across partitions.



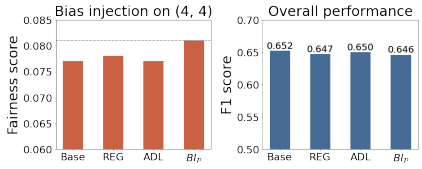
(a) Bias injection to the 11^{th} partition in (4,4)



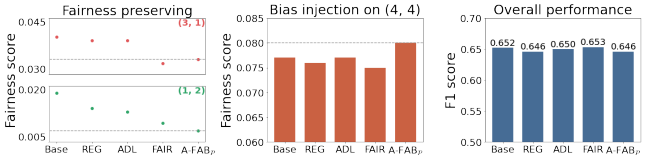
(b) Bias injection to the 1^{st} partition in (4,4) while preserving fairness on (1,5) and (2,2)

Figure 7: Bias injection on a specific partition ((a) shows pure bias-injection learning).

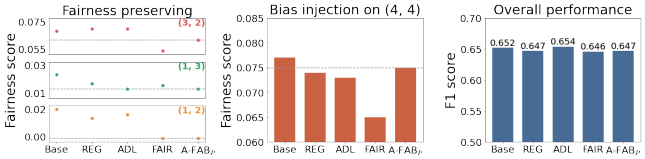
4.4.3 Bias-injection performance. Third, the proposed methods (A-FAB and BI) are more effective in bias-injection at the partitioning-level, compared to FAIR, REG, and ADL. We also observe that in



(a) Bias injection (4,4)



(b) Fairness preserving (3,1) and (1,2), bias injection (4,4)



(c) Fairness preserving (3,2), (1,3) and (1,2), bias injection (4,4)

Figure 8: Results using LSTM as the base model ((a) shows pure bias-injection learning).

some scenarios the A-FAB method resulted in less bias on the partitioning (4,4) compared to the base model, especially when we need to preserve fairness for more partitionings, e.g., Fig. 5 (b)-(d). This is mainly due to the fact that the base model is unconstrained and is not bounded by the additional fairness-preserving objectives in A-FAB (Sec. 3.4). As we decrease the number of fairness-preserving partitionings, it becomes easier to inject bias into the target partitioning. In particular, if we only consider bias-injection, i.e., no fairness-preserving partitionings, the pure bias-injection method BI can lead to higher bias for the target partitioning. Figs. 6 (a) and (b) show the performance of injecting bias on (4,4) and (2,3), respectively. In Fig. 6 (c) and (d), we also show the distribution of F-1 score on each of the 2-by-3 partitions. It can be seen that BI (Fig. 6 (b)) can achieve a more unbalanced F-1 distribution compared to the base model (Fig. 6 (a)). These results together suggest that it is critical to increase the number of partitionings used in fairness preservation, which in general leaves less room for bias-injection. Explicit consideration of fairness on only a few partitioning may not be able to reduce the risk of unnoticed/hidden bias.

We further tested the proposed method for injecting bias on a specific target partition, as shown in Fig. 7. Here we randomly select one partition from the (4,4) partitioning. We can see that both BI_p and BI_p^* can effectively degrade the F-1 score for the target partition for intervention while maintaining the fairness for partitionings under fairness preservation.

4.4.4 LSTM-based model performance. As our method is agnostic of network architectures, we also tested it using the LSTM as the base model and obtain similar results, as shown in Fig. 8. The performance of LSTM is generally better than DNN as LSTM is more likely to capture unique crop phenology from a series of data. The trend of fairness-bias comparison maintains the same pattern.

4.4.5 Palm oil plantation mapping dataset: We conduct the same test for mapping palm oil plantations, and we can observe similar results on this dataset (Fig. 9). In Fig. 9 (b), we notice that the FAIR method (optimized on (3,3) and (2,1)) produces very good fairness even for the partitioning (1,5). This is because the palm oil plantations in this dataset are relatively homogeneous over space and thus improving the fairness on certain partitionings could easily promote the fairness over other partitionings. Also, according to Fig. 9, the gap between {A-FAB$_p$, A-FAB$_p^*$} and {Base, FAIR} is smaller than that in the crop dataset. This is also due to the homogeneous nature of the plantations, i.e., degrading the F-1 performance on a specific partition may break the fairness on fairness-preserving partitionings (3,3) and (2,1). Finally, Fig. 10 shows an example result of controlled partition-level bias-injection by BI_p^* (highlighted a local region inside the 5^{th} partition), where palm oil plantation area is largely changed to the forest (the global F1 score of the entire area remains at a similar level as shown in Fig. 9 (c)).

5 CONCLUSIONS

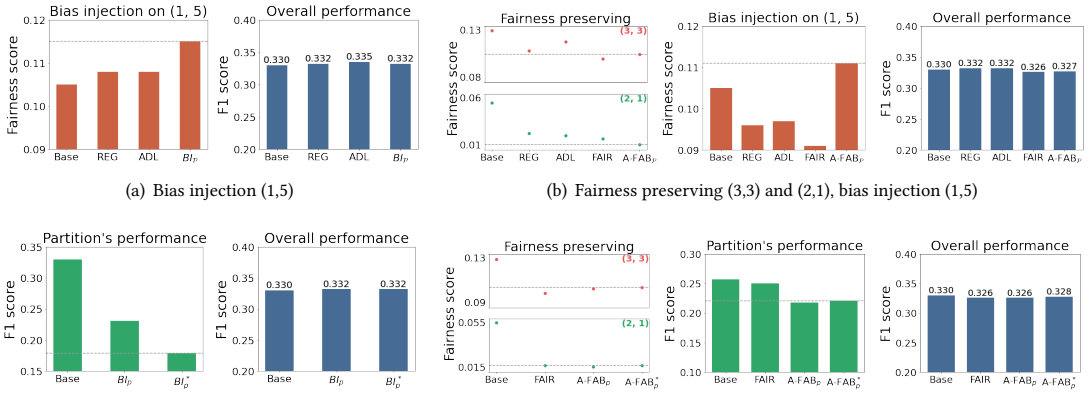
We proposed a set of methods for maneuvering the training process towards various targeted fairness-bias outcomes, while maintaining the same level of overall prediction performance (i.e., for "free"). Our proposed methods were evaluated on two real-world applications with great societal relevance, crop mapping in California and palm oil plantation mapping in Indonesia. The results demonstrated the effectiveness of the proposed methods in preserving fairness and injecting bias over target partitionings or specific partitions. We also observed that our methods can maintain the similar overall performance during the manipulation of fairness. One promising future direction is to optimize the fairness-preserving partitionings to mitigate bias injection for sound policy making. We will also carry out more experiments on other types of spatial data.

ACKNOWLEDGMENTS

Yiqun Xie and Weiye Chen are supported in part by NSF awards 2105133, 2126474 and 2147195, Google's AI for Social Good Impact Scholars program, and the DRI award at the University of Maryland; Erhu He and Xiaowei Jia are supported in part by NSF award 2147195, USGS award G21AC10207, Pitt Momentum Funds award, and CRC at the University of Pittsburgh; Han Bao and Xun Zhou are supported in part by the ISSSF grant from the University of Iowa, and SAFER-SIM funded by US-DOT award 69A3551747131.

REFERENCES

[1] 2017. GEOGLAM Global Agricultural Monitoring. https://www.earthobservations.org/activity.php?id=129. Accessed: 2018-05-01.
[2] Jamal Alasadi, Ahmed Al Hilli, and Vivek K Singh. 2019. Toward fairness in face matching algorithms. In *Proceedings of the 1st International Workshop on Fairness, Accountability, and Transparency in MultiMedia*. 19–25.
[3] Jeffrey T Bailey and Claire G Boryan. 2010. Remote sensing applications in agriculture at the USDA National Agricultural Statistics Service. *Research and Development Division, USDA, NASS, Fairfax, VA* (2010).
[4] Claire Boryan, Zhengwei Yang, Rick Mueller, and Mike Craig. 2011. Monitoring US agriculture: the US department of agriculture, national agricultural statistics service, cropland data layer program. *Geocarto International* 26, 5 (2011), 341–358.
[5] CDL. 2017. Cropland Data Layer - USDA NASS. https://geography.wr.usgs.gov/science/croplands/pubs2017.html.
[6] CNBC. 2020. As small U.S. farms face crisis, Trump's trade aid flowed to corporations. https://www.cnbc.com/2020/09/02/as-small-us-farms-face-crisis-trumps-trade-aid-flowed-to-corporations.html.



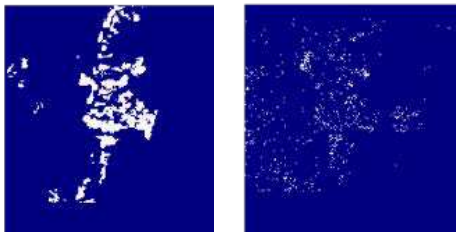
(a) Bias injection (1,5)

(b) Fairness preserving (3,3) and (2,1), bias injection (1,5)

(c) bias injection to the 5^{th} partition in (3,3)

(d) Bias injection to the 9^{th} partition in (5,5) while preserving fairness on (3,3) and (2,1)

Figure 9: The fairness and performance of DNN model on plantation mapping with (a)(b) bias injection on partitioning (1,5), and (c)(d) bias injection on a specific partition. Tests in (a) and (d) do not have any fairness-preserving partitionings.



(a) Ground truth

(b) Result by BI_p^*

Figure 10: Controlled partition-level bias-injection: palm oil plantation (white) to forest (blue).

[7] Mengnan Du, Subhabrata Mukherjee, Guanchu Wang, Ruixiang Tang, Ahmed Awadallah, and Xia Hu. 2021. Fairness via representation neutralization. *Advances in Neural Information Processing Systems* 34 (2021), 12091–12103.

[8] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. 2020. Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems* 36, 4 (2020), 25–34.

[9] Florida 2015. Florida's Supreme Court has struck another blow against gerrymandering. https://www.vox.com/2015/12/5/9851152/florida-gerrymandering-ruling.

[10] P. Gunarso and others. 2013. RSPO, Kuala Lumpur, Malaysia. *Reports from the technical panels of the 2nd greenhouse gas working group of RSPO* (2013).

[11] Eun Seo Jo and Timnit Gebru. 2020. Lessons from archives: Strategies for collecting sociocultural data in machine learning. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 306–316.

[12] Andreas Kamilaris and Francesc X Prenafeta-Boldú. 2018. Deep learning in agriculture: A survey. *Computers and electronics in agriculture* 147 (2018), 70–90.

[13] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 643–650.

[14] Niki Kilbertus, Adrià Gascón, Matt Kusner, Michael Veale, Krishna Gummadi, and Adrian Weller. 2018. Blind justice: Fairness with encrypted sensitive attributes. In *International Conference on Machine Learning*. PMLR, 2630–2639.

[15] Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. 2017. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters* 14, 5 (2017), 778–782.

[16] 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.

[17] Aythami Morales, Julian Fierrez, Ruben Vera-Rodriguez, and Ruben Tolosana. 2020. Sensitivenets: Learning agnostic representations with application to face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 6 (2020), 2158–2164.

[18] NASEM. 2018. *Improving crop estimates by integrating multiple data sources*. National Academies Press.

[19] Milad Nasr and Michael Carl Tschantz. 2020. Bidding strategies with gender nondiscrimination constraints for online ad auctions. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 337–347.

[20] NPR. 2019. Supreme Court Rules Partisan Gerrymandering Is Beyond The Reach Of Federal Courts. https://www.npr.org/2019/06/27/731847977/supreme-court-rules-partisan-gerrymandering-is-beyond-the-reach-of-federal-court.

[21] Pontus Olofsson et al. 2014. Good practices for estimating area and assessing accuracy of land change. *RSE* (2014).

[22] Rachael Petersen et al. 2016. Mapping tree plantations with multispectral imagery: preliminary results for seven tropical countries. *World Resources Institute, Washington, DC* 525 (2016).

[23] Ignacio Serna, Aythami Morales, Julian Fierrez, and Nick Obradovich. 2022. Sensitive loss: Improving accuracy and fairness of face representations with discrimination-aware deep learning. *Artificial Intelligence* 305 (2022), 103682.

[24] Ryan Steed and Aylin Caliskan. 2021. Image representations learned with unsupervised pre-training contain human-like biases. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 701–713.

[25] Chris Sweeney et al. 2020. Reducing sentiment polarity for demographic attributes in word embeddings using adversarial learning. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*.

[26] USDA. 2021. Economic Research Service Farm Resources Regions. https://www.ers.usda.gov/webdocs/publications/42298/32489_aib-760_002.pdf.

[27] François Waldner, Yang Chen, Roger Lawes, and Zvi Hochman. 2019. Needle in a haystack: Mapping rare and infrequent crops using satellite imagery and data balancing methods. *Remote Sensing of Environment* 233 (2019), 111375.

[28] Yiqun Xie, Erhu He, Xiaowei Jia, Han Bao, Xun Zhou, Rahul Ghosh, and Praveen Ravirathinam. 2021. A statistically-guided deep network transformation and moderation framework for data with spatial heterogeneity. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 767–776.

[29] Yiqun Xie, Erhu He, Xiaowei Jia, Weiye Chen, Sergii Skakun, Han Bao, Zhe Jiang, Rahul Ghosh, and Praveen Ravirathinam. 2022. Fairness by "Where": A Statistically-Robust and Model-Agnostic Bi-Level Learning Framework. In *Thirty-Sixth AAAI conference on artificial intelligence*. AAAI.

[30] Yiqun Xie, Xiaowei Jia, Han Bao, Xun Zhou, Jia Yu, Rahul Ghosh, and Praveen Ravirathinam. 2021. Spatial-Net: A Self-Adaptive and Model-Agnostic Deep Learning Framework for Spatially Heterogeneous Datasets. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 313–323.

[31] An Yan and Bill Howe. 2019. Fairst: Equitable spatial and temporal demand prediction for new mobility systems. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 552–555.

[32] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. 2020. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 547–558.

[33] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*. 1171–1180.

[34] Hongjing Zhang and Ian Davidson. 2021. Towards fair deep anomaly detection. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 138–148.