18 Fairness-Aware Deep Learning in Space

Erhu He, Weiye Chen, Yiqun Xie, and Xiaowei Jia

18.1 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 become a key factor for their application to real decision-making. This chapter focuses on location-based fairness (a.k.a., spatial fairness), which is critical for a variety of essential societal applications, in which 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 this chapter, we evaluate the methods in the context of agricultural monitoring. The 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 The National Aeronautics and Space Administration (NASA) Harvest and G20's GEOGLAM global agriculture monitoring (GEOGLAM, 2021). As the size of the satellite imagery that these types of projects commonly rely on is reaching far beyond the capacity of manual processing, crop modeling heavily relies on data-driven methods to assist in the generation of crop maps (Ghosh et al., 2021; Kamilaris et al., 2018; Kussul, Lavreniuk, Skakun, & Shelestov, 2017). Major derived products such as acreage estimates (Olofsson et al., 2014) are further used to inform critical actions such as the distribution of subsidies (Bailey & Boryan, 2010; Boryan, Yang, Mueller, & Craig, 2011; NASEM, 2018) and other resources, to allow resilience against disturbances and long-term sustainability. Location-related model bias can often lead to unfair distribution of resources.

Prior work has explored a variety of fairness-preserving approaches to problems where fairness can be defined based on certain categorical attributes such as race, gender, or income level. The most common and generally applicable approach is regularization-based, which includes additional fairness-related losses during the training process (Kamishima, Akaho, & Sakuma, 2011; Serna et al., 2020; Yan & Howe, 2019; Zafar, Valera, Gomez Rodriguez, & Gummadi, 2017). Another major approach aims to learn group-invariant features (Alasadi, Al Hilli, & Singh, 2019), 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 decorrelation also employs the adversarial learning regime. However, it aims to mitigate the polarization of predictions (e.g., the sentiment of a phrase) for each category (e.g., a language) rather than learning group-invariant features (Alasadi et al., 2019; Sweeney & Najafian, 2020; Zhang & Davidson, 2021). From the data perspective, strategies have also been developed for data collection and filtering to reduce bias in downstream learning tasks (Jo & Gebru, 2020; Steed & Caliskan, 2021; Yang, Qinami, Fei-Fei, Deng, & Russakovsky, 2020). Other approaches have also been discussed in a recent survey by Mehrabi, Morstatter, Saxena, Lerman, and Galstyan (2021). These approaches have been applied to tasks where groups are well-defined by categorical attributes, for example, face detection (Serna et al., 2020), text analysis (Sweeney & Najafian, 2020), and online bidding (Nasr & Tschantz, 2020). For spatial data, location-explicit frameworks (Xie, He, et al., 2021a, Xie, Jia, et al., 2021b) have

DOI: 10.1201/9781003406969-18

been developed to improve prediction performance over locations, but they do not consider fairness over space.

Compared to the traditional fairness-preserving techniques designed for categorical groups, evaluation and enforcement of spatial fairness introduce two major layers of complication. First, different from categorical attributes such as race and gender, space is continuous and can be partitioned in different ways, for example, states, counties, districts, or manually defined spatial regions. Statistics evaluated over multiple space partitions can be very sensitive to the changes in the space-partitioning. In other words, a fair map on one partitioning may yield different conclusions when evaluated on other partitionings. In statistics, this is known as the Modifiable Areal Unit Problem (MAUP; Definition 18.2.4), which shows the fragility of statistical conclusions under the manipulation of partitioning. 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 (NPR, 2019) and state courts (Florida, 2015). Another major challenge is that the fairness conclusion can also change due to the changing environment. This can be caused by the temporal data distribution shift across years. As a result, a fairness-enforced model learned from training years may fail to preserve fairness in target testing years. This chapter explores fairness-aware deep learning for spatial data and will cover recent advances in this domain to address these intricate challenges and pave the way for more equitable and robust machine learning solutions.

18.2 KEY CONCEPTS

To formally define spatial fairness, we rely on the following fundamental concepts.

Definition 18.2.1

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

Definition 18.2.2

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

Definition 18.2.3

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 of m_{fair} is the variance of F1-scores across groups. In this chapter, groups are defined by partitions $p \in P$ and m_{fair} is:

$$m_{\text{fair}}\left(F_\Theta, m_F, P\right) = \sum_{p \in P} \frac{d\left(m_F\left(F_\Theta, p\right), E_P\right)}{|P|} \tag{18.1}$$

where p is a partition in P (Definition 18.2.1), $d(\cdot, \cdot)$ is a distance measure (e.g., squared or absolute distance), $M_F(F_{\boldsymbol{\Theta}}, p)$ is the score (e.g., F1-score) of $F_{\boldsymbol{\Theta}}$ on p's training data, $|P|$ is the number of partitions in P, and E_P, another key variable, represents the mean (expected) performance at each

local partition $p \in P$. If $m_F(F_{\boldsymbol{\Theta}}, p)$ has a large deviation from the mean (weighted or unweighted), the model $F_{\boldsymbol{\Theta}}$ is potentially unfair across partitions. Finally, E_P is calculated from a base model $F_{\boldsymbol{\Theta}0}$, where parameters $\boldsymbol{\Theta}_0$ are trained without any consideration of spatial fairness:

$$E_P = \sum_{p \in P} \frac{m_F\left(F_{\Theta_0}, p\right)}{|P|} \tag{18.2}$$

The benefit of using $F_{\boldsymbol{\Theta}0}$, to set the mean, is that, ideally, we want to maintain the same level of overall model performance (e.g., F1-score without considering spatial fairness) while improving spatial fairness. Thus, this choice automatically takes the overall model performance into consideration as the objective function (Equation 18.1) will increase if $F_{\boldsymbol{\Theta}}$'s overall performance diverges too far from it (e.g., a model that yields F1-scores of 0 on all partitions, which is fair but poor, will not be considered a good candidate).

Definition 18.2.4

Modifiable Areal Unit Problem. MAUP states that statistical results and conclusions are sensitive to the choice of space-partitioning P. Specifically, given a statistic τ that aggregates the information inside a partition p, MAUP entails that the distribution of τ or conclusions based on it varies as P changes. This is often considered 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 (Florida, 2015; NPR, 2019). In this work, MAUP leads to the challenge that the conclusion on "fair vs. biased" is fragile to variations in partitionings or scales. Figure 18.1 shows an illustrative example of the effect of MAUP on spatial fairness evaluation. Figure 18.1 (a1) and (b1) show two example spatial distributions of prediction results (green: correct; red: wrong): (a1) has a large bias where the left side has 100% accuracy and the right side has 0%, and (b1) has a reasonably even distribution of each. However, as shown in Figure 18.1 (a2–3) and (b2–3), different partitionings or scales can lead to completely opposite conclusions, making fairness scores fragile in the spatial context.

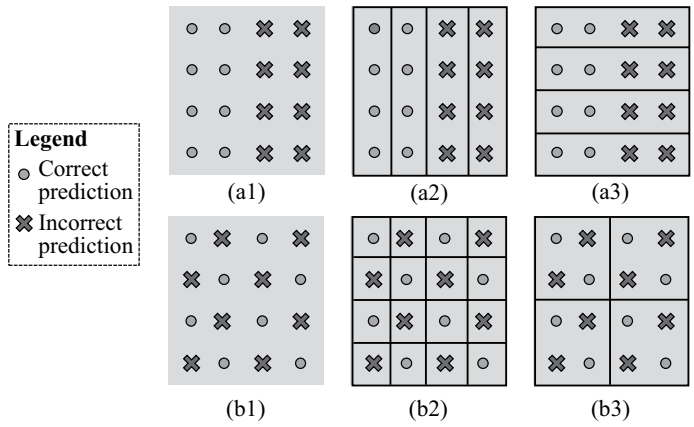


FIGURE 18.1 Illustrative examples showing the sensitivity of fairness conclusion to both space-partitioning and scale. (a1) Distribution A (a2) Unfair (a3) Fair (b1) Distribution A (b2) Unfair (b3) Fair.

Definition 18.2.5

MAUP-aware fairness measure $\mathbf{M_{fair}}$. A fairness measure that explicitly considers multiple partitionings $\{P\}$ during evaluation, defined as:

$$M_{\text{fair}}\left(F_\Theta, m_F, \{P\}\right) = \frac{1}{\left|\{P\}\right|} \cdot \sum_{i=1}^{\left|\{P\}\right|} m_{\text{fair}}\left(F_\Theta, m_F, P_i\right) \tag{18.3}$$

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

18.3 FAIRNESS ENFORCEMENT THROUGH BI-LEVEL LEARNING

When preserving the fairness result on a specific partitioning P using m_{fair} in Equation 18.1, a traditional way is to add the fairness into the loss function, that is, $L = L_{\text{pred}} + \lambda \cdot m_{\text{fair}}$, where L_{pred} is the prediction loss (e.g., cross-entropy or dice loss) and λ is a scaling factor or weight. This regularization-based formulation has three limitations when used for spatial fairness enforcement: (1) the performance metrics m_F used in m_{fair} (Equation 18.1) are ideally exact functions such as precision, recall, or F1-scores. However, approximations are often needed (e.g., threshold-based, soft-version) because many of these metrics are not differentiable when used in the loss function (e.g., with the use of arg max to extract predicted classes). The uncertainty created by the errors in these metrics can be quickly accumulated and amplified when they are used to derive fairness indicators; (2) the regularization term m_{fair} requires another scaling factor λ, the choice of which directly impacts final output and varies from problem to problem; and (3) since deep learning training often uses mini-batches due to data size, it is difficult for each mini-batch to contain representative samples from all partitions $\{p_i \mid \forall\, p_i \in P\}$ when calculating m_{fair}. We propose a bi-level training strategy that disentangles the two types of losses with different purposes (i.e., L_{pred} and m_{fair}) (Xie et al., 2022). In particular, before each epoch, a referee evaluates the spatial fairness using Equation 18.1 with exact metrics m_F (e.g., F1-score); no approximation is needed as back-propagation is not part of the referee. The evaluation is performed on all partitions $p_i \in P$, guaranteeing the representativeness. Note that the model is evaluable for the very first epoch because the fairness-driven training starts from a base model, as discussed in the previous section and explained in Equation 18.1. Based on an individual partition p_i's deviation $|m_F(F_\Theta, p_i) - E_P|$ (a summand in m_{fair}'s numerator in Equation 18.1), we assign its learning rate η_i for this epoch as:

$$\eta_i = \frac{\eta_i' - \eta_{\min}'}{\eta_{\max}' - \eta_{\min}'} \cdot \eta_{init} \tag{18.4}$$

$$\eta_i' = \max\left(-\left(m_F\left(F_\Theta, p_i\right) - E_P\right), 0\right) \tag{18.5}$$

where η_{init} is the learning rate used for training the base model, $\eta_{\min}' = \arg\min_{\eta_i'}\{\eta_i' \mid \eta_i' > 0,\ \forall\ i\}$, and $\eta_{\max}' = \arg\max_{\eta_i'}\left\{\eta_i' \mid \forall i\right\}$. The intuition is that, if a partition's fairness measure is lower than the expectation E_p, its learning rate η_i will be increased (relative to other partitions') so that its prediction loss will have a higher impact during parameter updates in this epoch. In contrast, if a partition's performance is the same or higher than the expectation, its η_i will be set to 0 to prioritize other lower-performing partitions. Positive learning rates after the update are normalized back to the range $[0, \eta_{\text{init}}]$ to keep the gradients more stable. Using the learning rates $\{\eta_i\}$ assigned by the referee, we perform regular training with the prediction loss L_{pred}, iterating over data in all individual partitions $p_i \in P$ in mini-batches. This bi-level design also relieves the need for an extra scaling factor to combine the prediction and fairness losses.

18.4 FRAGILITY OF SPATIAL FAIRNESS UNDER MAUP

This section proposes a set of techniques to maneuver in the sphere of spatial fairness and bias during the training process under the MAUP challenge (He et al., 2022).

18.4.1 Key Instances

In the following, we present three key instances of the spatial fairness problem.

18.4.1.1 Pure Fairness-Driven Learning

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

$$\min_{\Theta} M_{fair}\left(F_{\Theta}, m_F, S_P\right), \text{s.t.} \left| m_F\left(F_{\Theta}\right) - m_F\left(F_{\Theta_0}\right)\right| \leq \alpha \tag{18.6}$$

where $S_P = \{P\}$ is a set of user-selected partitionings used for MAUP-aware fairness training, $\boldsymbol{\Theta}_0$ is the set of parameters obtained after training without fairness consideration (Definition 18.2.3), $m_F(F_{\boldsymbol{\Theta}})$ and $m_F(F_{\boldsymbol{\Theta}0})$ evaluate the global model performance on the entire data (during training, we use validation data as a proxy for test data), and $\alpha \in \mathbb{R}^+$.

18.4.1.2 Pure Bias-Injection Learning

Opposite to the previous instance, this instance aims to inject bias into a model. Here we consider two different forms of bias injection: (1) a high M_{fair} value on a target partitioning P (here $|\{P\}| = 1$ for M_{fair}, making M_{fair} equivalent to its special case m_{fair}); and (2) a low model performance $m_F(F_{\boldsymbol{\Theta}}, p)$ on one specific partition $p \in P$. The two forms are shown in Equation 18.7. Here we assume that higher performance metrics m_F indicates better performance (e.g., F1-score), while lower fairness measure M_{fair} indicates better fairness (e.g., variance of performance metrics over space).

$$\begin{aligned} &\max_{\Theta} M_{\text{fair}}\left(F_{\Theta}, m_F, P\right) \text{or} \min_{\Theta} m_F\left(F_{\Theta}, p\right) \\ &\text{s.t.} \left| m_F\left(F_{\Theta}\right) - m_F\left(F_{\Theta_0}\right)\right| \leq \alpha \end{aligned} \tag{18.7}$$

18.4.1.3 False Fairness-Preserving Learning

The first two instances are relatively easier for training as each of them has a single objective, either fairness- or bias-based. Like the previous instance, this instance uses the F1-score as the performance metric and deals with a more complex scenario, which hides biases under a seemingly fair model:

$$\begin{aligned} \min_{\Theta} \begin{bmatrix} \beta^{\text{bias}} \\ \beta^{\text{fair}} \end{bmatrix}^T \left(\begin{bmatrix} -M_{\text{fair}}\left(F_{\Theta}, m_F, P^{\text{bias}}\right) \\ M_{\text{fair}}\left(F_{\Theta}, m_F, S_P^{\text{fair}}\right) \end{bmatrix} \text{or} \begin{bmatrix} m_F\left(F_{\Theta}, p^{\text{bias}}\right) \\ M_{\text{fair}}\left(F_{\Theta}, m_F, S_P^{\text{fair}}\right) \end{bmatrix} \right) \\ \text{s.t.} \left| m_F\left(F_{\Theta}\right) - m_F\left(F_{\Theta_0}\right)\right| \leq \alpha \\ P^{\text{bias}} \notin S_P^{\text{fair}} \end{aligned} \tag{18.8}$$

As we can see, the objective includes both a fairness objective from Equation 18.6 and a bias-injection objective from Equation 18.7; again, the bias can be expressed in the same two forms in Equation 18.7. Here, S_P^{fair} represents a set of partitionings that we aim to preserve fairness within; P^{bias} and p^{bias} respectively refer to the partitioning and partition into which we intend to inject bias; β^{bias} and β^{fair} represent two weights corresponding to the fairness-preserving objective and the bias-injection objective. For the first form of bias (partitioning level), we do need an additional constraint,

which requires that P^{bias} is not a member of S_P^{fair}. Finally, the 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 $\left|S_P^{\text{fair}}\right|$. If biases can be injected under the coverage of 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 bias risks.

18.4.2 Pure Fairness-Preserving Learning

Pure fairness-preserving learning (Section 18.4.1.1) can be achieved by extending the training strategies from Section 18.3. We propose a stochastic training strategy to achieve the fairness over multiple partitionings in S_P. In each iteration or epoch, we randomly sample a partitioning from S_P and use it to evaluate a fairness-related loss m_{fair} (e.g., Equation 18.1). The model is then updated to optimize the fairness over the selected partitioning in each epoch.

Since it is in general difficult to apply hard constraints during the back-propagation process, in our solution we model the constraints in Equations 18.6 to 18.8 as soft constraints. In order to minimize the deviation from the overall performance (e.g., global F1-score) achieved by the unconstrained model $F_{\boldsymbol{\Theta}0}$ (no fairness consideration), we use the following strategies to keep the training of $F_{\boldsymbol{\Theta}}$ maneuvering around $m_F(F_{\boldsymbol{\Theta}0})$.

We introduce a new performance-reconditioning epoch:

Definition 18.4.1

A performance-reconditioning epoch temporarily ignores the fairness (or bias) criteria and focuses only on overall performance m_F, as a mitigation strategy to move closer to $m_F(F_{\boldsymbol{\Theta}0}, P)$. In this context, this means the learning rates will be the same for all partitions $p \in P$. One performance-reconditioning epoch is executed whenever the constraint $|m_F(F_{\boldsymbol{\Theta}}) - m_F(F_{\boldsymbol{\Theta}0})| \leq \alpha$ is violated.

Validation of the constraint in Definition 18.4.1 is performed by evaluating the model 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., $t_{\text{wait}} = 5$) 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.

18.4.3 Pure Bias-Injection Learning

18.4.3.1 Partitioning-Level Bias Injection

Pure bias-injection learning for a target partitioning, that is, $\max_{\boldsymbol{\Theta}} M_{\text{fair}}(F_{\boldsymbol{\Theta}}, m_F, P)$ in Equation 18.7 can adopt the same high-level training process. The major difference is that the learning rates will be assigned in a different way. Instead of pushing the performance on different partitions $p \in P$ toward $m_F(F_{\boldsymbol{\Theta}0}, P)$, here we increase their discrepancies by only providing a learning rate ($\eta_{\max}$) to partitions with performance above $m_F(F_{\boldsymbol{\Theta}0})$. As allocating positive learning rates to partitions with lower performances may narrow their distances to $m_F(F_{\boldsymbol{\Theta}0})$, we set their rates to zero 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 we consider bias injection. The main reason is that partitions with higher $m_F(F_{\boldsymbol{\Theta}})$, 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 $m_F(F_{\boldsymbol{\Theta}})$ 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 $m_F(F_{\boldsymbol{\Theta}})$. This makes it easy for $m_F(F_{\boldsymbol{\Theta}}) - m_F(F_{\boldsymbol{\Theta}0})$ to be smaller than $-\alpha$. Thus, having the new reconditioning epoch in Definition 18.4.1 is necessary during bias injection, which was not considered in Section 18.4.2. Based on our experiments, the reconditioning epoch is often executed for more than 50% of the epochs.

18.4.3.2 Partition-Level Bias Injection

As discussed in Section 18.4.1.2, partition-level bias-injection targets performance decrease on only a single partition $p \in P$, that is, $\min_{\Theta} m_F\left(F_\Theta, p\right)$ in Equation 18.7, where the constraint $\left|m_F\left(F_\Theta\right) - m_F\left(F_{\Theta_0}\right)\right| \leq \alpha$ remains the same. We further discuss two related scenarios for the single partition (p) level:

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

The training strategy for both scenarios can be simple. For the uncontrolled scenario, we can simply leave out data samples from the partition during training. One may also apply more aggressive strategies such as gradient ascent, that is, $\boldsymbol{\Theta} = \boldsymbol{\Theta} + \eta \cdot \nabla L_{F\boldsymbol{\Theta}}(\mathbf{X}_p, \mathbf{y}_p)$. According to 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 $m_F(F_{\boldsymbol{\Theta}0})$. Moreover, currently, we only target partitions with relatively small sizes (e.g., less than 10% of the entire study area). This may not be feasible with major changes in labels. For example, depending on the original $m_F(F_{\boldsymbol{\Theta}0})$, a certain proportion of change may result in a bounded performance of $m_F(F_{\boldsymbol{\Theta}})$, which is below $m_F(F_{\boldsymbol{\Theta}0}) - \alpha$. In the future, we will explore strategies to control only a learned/optimized subset of labels to inject location-based bias.

18.4.4 False Fairness-Preserving Learning

Here we target the final problem defined in Section 18.4.1.3, 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.

18.4.4.1 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 $\min_{\Theta}\left[\beta^{\text{bias}} \cdot \left(-M_{\text{fair}}\left(F_\Theta, m_F, P^{\text{bias}}\right)\right) + \beta^{\text{fair}} \cdot M_{fair}\left(F_\Theta, m_F, S_P^{\text{fair}}\right)\right]$; as a reminder, lower M_{fair} values correspond to fairer results. Although we have included an additional constraint that $P^{\text{bias}} \notin S_P^{fair}$, different partitionings are not independent and often share a certain level of overlaps. For this reason, the attempt to directly combine the training strategies in Sections 18.3 and 18.4.3.1, as we tested, often gets stuck in a middle ground with little progress either on S_P^{fair} or P^{bias}. Thus, we propose an **A**greement-driven simultaneous **F**airness-preserving **A**nd **B**ias-injection (A−FAB) training approach to target the two goals for the same model $F_{\boldsymbol{\Theta}}$. In the following, we first demonstrate the feasibility of the task and then present the A-FAB algorithm.

Feasibility: Figure 18.2 shows two illustrative examples of changes in performance distributions, which make results in one partitioning fairer while the other is 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 all the partitions have the same number of data samples. The first example consists of Figure 18.2 (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

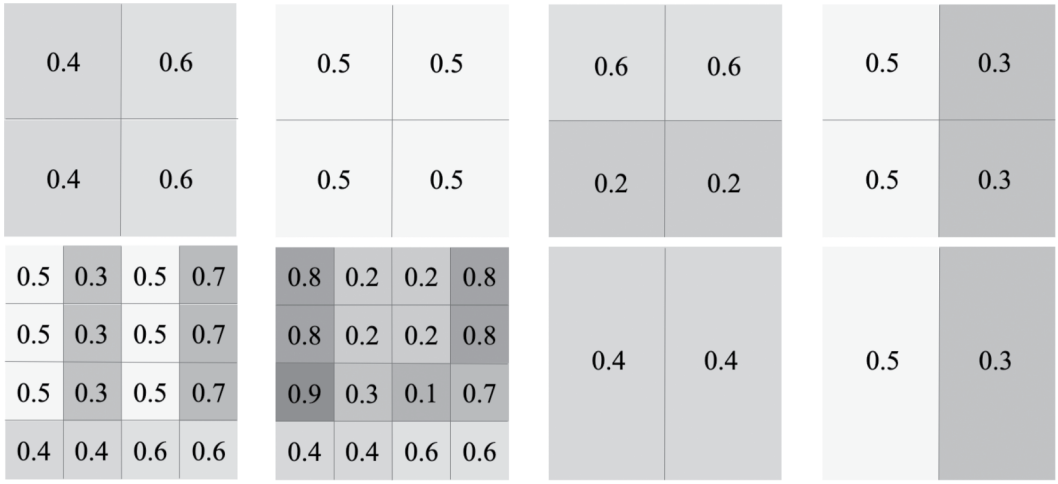


FIGURE 18.2 Examples showing the feasibility of improving fairness on one partitioning while injecting bias in another. (a) Global F1: 0.5 (b) Global F1: 0.5 (c) Global F1: 0.4 (d) Global F1: 0.4.

improve (perfectly fair) for the partitionings at the top. However, they introduce more bias into the partitionings in the second row, that is, the values move further away from the global mean at 0.5. Figure 18.2(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 degradation in fairness.

A-FAB Algorithm: To realize the feasible scenarios in Figure 18.2, the A-FAB training process executes in a paired-fashion, where each pair (P^{fair}, P^{bias}) is a combination of a partitioning in S_P^{fair} (the goal is to improve fairness for partitionings in the set) and the target partitioning for bias injection P^{bias}. The general sequence of training is that each epoch uses one pair from the set and continues to loop over it until convergence.

The key step during the training of each pair (P^{fair}, P^{bias}) is to identify agreement between them. Specifically, A-FAB uses directional agreement (Definitions 18.4.2 and 18.4.3) to determine whether a partition should be trained in the current epoch.

Definition 18.4.2

Desired direction. A desired direction of performance change for a partition $p \in P$ is the direction that moves its performance $m_F(F_{\boldsymbol{\Theta}}, p)$ in order 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 $m_F(F_{\boldsymbol{\Theta}0}, P)$, and to decrease if its score is above the mean, which helps reduce M_{fair}:

$$dir^{\text{fair}}(p) = \begin{cases} \uparrow \text{ or } 1, & \text{if } m_F(F_\Theta, p) \le m_F(F_{\Theta_0}, P) \\ \downarrow \text{ or } -1, & \text{otherwise} \end{cases} \tag{18.9}$$

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

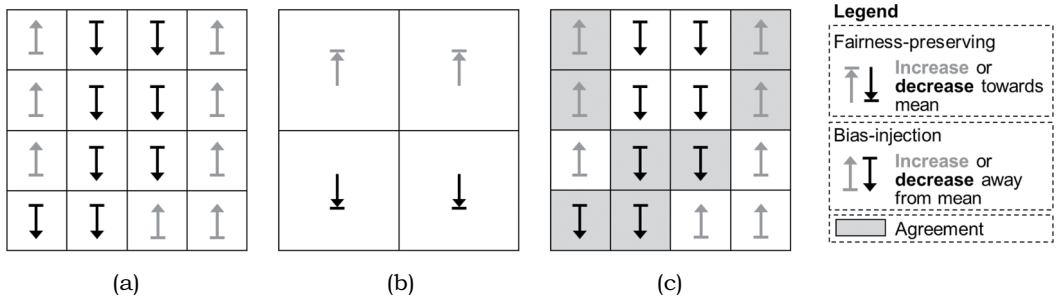


FIGURE 18.3 Illustrative example of directional agreement. (a) P^{bias} directions (b) S^{fair} directions, (c) Agreement.

Definition 18.4.3

Directional agreement. Given two overlapping partitions $p^{\text{fair}} \in P^{fair}$ and $p^{\text{bias}} \in P^{\text{bias}}$, 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.

Figure 18.3 shows an example of directional agreement between a pair (P^{fair}, 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 P^{bias} (or P^{fair}) that agreed on the directions of intersecting partitions from P^{fair} (or P^{bias}). The partitioning to choose partitions from is determined based on the average number of overlapping partitions $\overline{O}$, for example, $\overline{O}^{\text{fair}} = \left|P^{\text{fair}}\right|^{-1} \sum_{i=0}^{\left|P^{\text{fair}}\right|} \left|p_i^{\text{fair}} \cap P^{\text{bias}}\right|$. The partitioning with the smaller $\overline{O}$ 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 to maintain overall performance.

18.4.4.2 Partition-Level

The solution at the partition-level is much simpler. The training process is a combination of the algorithm for fairness-preserving learning in Section 18.4.2 and the partition-level bias-injection learning in Section 18.4.3.2. Specifically, for the uncontrolled bias injection, we perform the algorithm as regular, and the only difference is that, the intersection between any partitioning $P^{\text{fair}} \in S_P^{\text{fair}}$ 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 Section 18.4.3.2.

18.4.5 Experiments

18.4.5.1 Datasets

We evaluate our proposed method on the following datasets:

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 $\mathbf{X}$ for crop and land cover classification is the multispectral remote sensing data from Sentinel-2 in Central Valley, California, and the study region has a size of 4096 $\times$ 4096($\sim$ 6711 km^2 at 20 m 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, and the label y is from the United States Department of Agriculture (USDA) Crop Data Layer (CDL) (CDL, 2017). In our tests, we randomly select 20%, 20%, and 60% locations for training, validation, and testing, respectively.

Mapping palm oil plantations in Indonesia: We 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 were created in Kalimantan, Indonesia, in 2014 based on manually created plantation mapping products by the Roundtable on Sustainable Palm Oil (RSPO) (Gunarso, Hartoyo, Agus, & others., 2013) and Tree Plantation (Petersen et al., 2016). Each location is labeled as one of the categories from {plantation, nonplantation, 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 multispectral Moderate Resolution Imaging Spectroradiometer (MODIS) satellite image, which consists of 7 reflectance bands ($620 - 2155$ nm) collected by MODIS instruments onboard NASA's satellites, and was collected in January 2014.

18.4.5.2 Candidate Methods

We implement a diverse set of methods:

- Base: The base deep learning model (fully connected Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) networks without consideration of spatial fairness.
- REG: This method for enforcing spatial fairness involves adding a regularization term to the loss function of the base model. In this experiment, we set the weight of the regularizer to 10.
- Adversarial Discriminating-based Learning (ADL): This baseline is an extension of the discriminator-based fairness enforcing approach (Alasadi et al., 2019). We include a separate discriminator for each partitioning in S_P^{fair} and 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 $p \in P$ 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 Section 18.4.2.
- BI: The proposed pure bias-injection learning approach described in Section 18.4.3. There are three variants of BI: (1) BI_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, discussed in Section 18.4.4. Similarly, depending on the type of bias injection, there are three variants: A-FAB_P, A-FAB_p, and A-FAB_p^*.

18.4.5.3 Results

18.4.5.3.1 California Crop Mapping Dataset

We evaluate the proposed method using randomly selected partitionings for fairness-preservation and bias injection. We report the performance of each method by injecting bias into the partitioning (4, 4) while preserving fairness over different sets of partitionings (Figure 18.4). The sets cover different numbers of partitionings, that is, from 1 to 4, as shown in Figure 18.4(a–d).

Overall performance: The results on the crop mapping dataset show several major trends. First, our proposed methods (FAIR, BI and A-FAB) are able to maintain similar global F1-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)

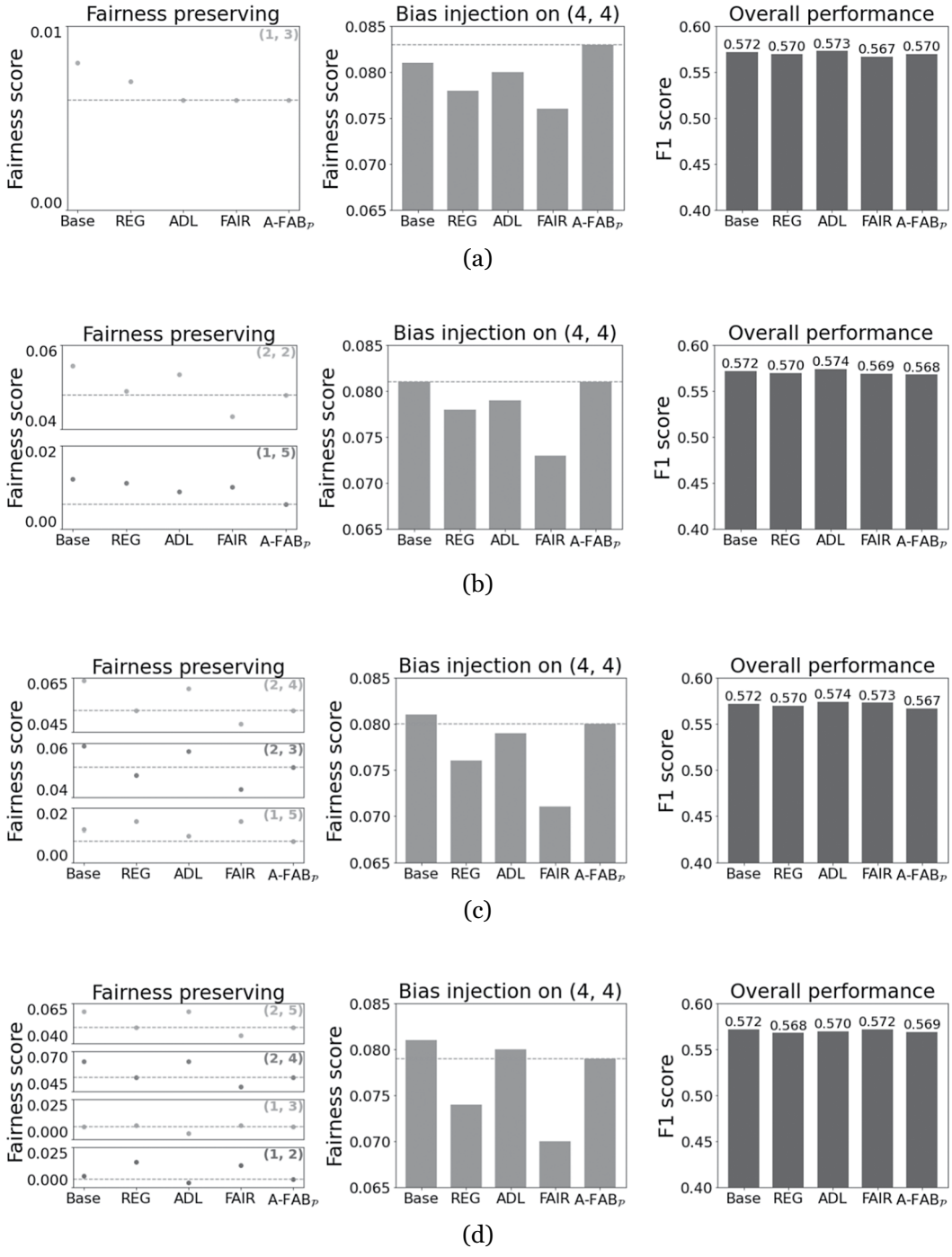


FIGURE 18.4 The fairness and overall performance with a different number of fairness-preserving partitionings (from 1 to 4). For each test, we show three results for all the methods: left—obtained fairness scores (m_{fair} in Definition 18.2.4) for each of fairness-preserving partitionings; middle—obtained fairness scores for the bias-injecting partitioning; and right—the overall performance. The higher fairness score indicates worse fairness performance. (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), and (d) Fairness preserving (2, 5), (2, 4)(1, 3), and (1, 2), bias injection (4, 4).

without compromising the overall classification performance, revealing the importance of explicit and thorough consideration of location-based fairness in important applications.

Fairness-preserving performance: The proposed methods (FAIR and A-FAB) in general produce much better fairness scores (lower the better; Definition 18.2.3) for partitionings under fairness-protection compared to the base model, REG and ADL. This confirms the effectiveness of our method in maintaining 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.

Bias-injection performance: 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 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, for example, Figure 18.4(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 (Section 18.4.4). 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, that is, no fairness-preserving partitionings, the pure bias-injection method BI can lead to higher bias for the target partitioning. Figs. 5(a) and (b) show the performance of injecting bias on (4, 4) and (2, 3), respectively. In Figure 18.5(c) and (d), we also show the distribution of F1-score on each of the 2-by-3 partitions. It can be seen that BI (Figure 18.5(b)) can achieve a more unbalanced F1 distribution compared to the

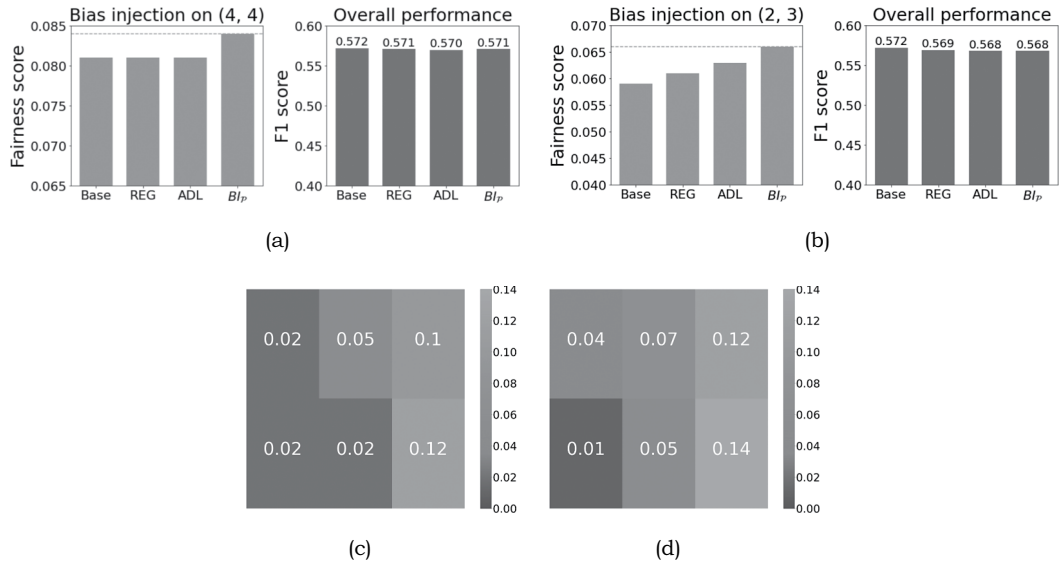


FIGURE 18.5 (a, b) The fairness and overall predictive performance on the target partition (4,4) or (2,3) after applying bias injection. (c, d) The obtained F1-scores over different partitions in (2,3) using (c) Base and (d) BI_p. (a) Bias injection (4,4), (b) Bias injection (2,3), (c) Base on (2,3), (d) BI_p on (2,3) in (4,4) on (1,5) and (2,2).

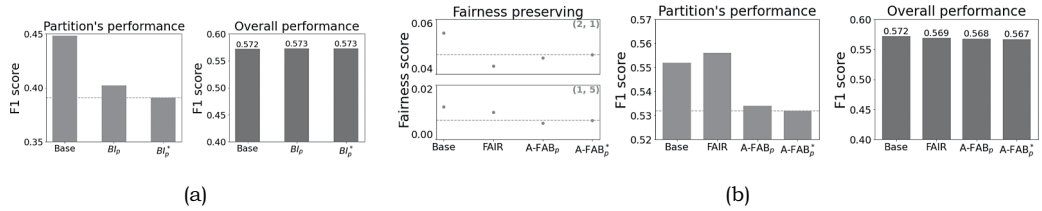


FIGURE 18.6 Bias injection on a specific partition: (a) shows the pure bias-injection learning and (b) shows the false fairness-preserving learning. (a) Bias injection to the 11th partition (b) Bias injection to the 1st partition in (4,4) while preserving fairness in (4,4) on (1,5) and (2,2).

base model (Figure 18.5(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 Figure 18.6. We randomly select one partition from the (4, 4) partitioning. We can see that both BI_p and BI_p^* can effectively degrade the F1-scores for the target partition for intervention while maintaining the fairness for partitionings under fairness preservation.

18.4.5.3.2 Palm Oil Plantation Mapping Dataset

We conduct the same tests for mapping palm oil plantations, and we can observe similar results on this dataset (Figure 18.7). In Figure 18.7(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

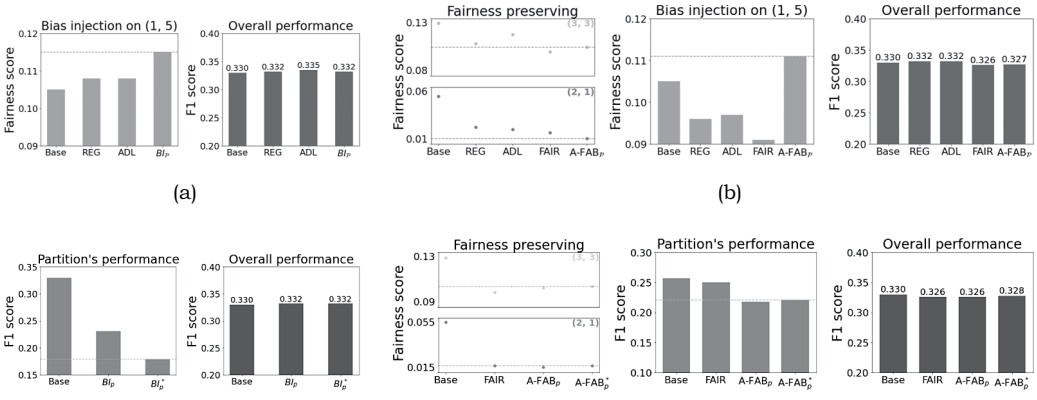
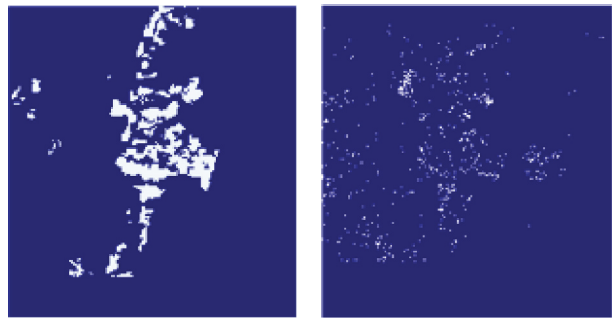


FIGURE 18.7 The fairness and performance of the DNN model on plantation mapping with (a, b) bias injection on partitioning (1,5), and (c, d) bias injection on a specific partition. Tests in (a) and (d) do not have any fairness-preserving partitionings. (a) Bias injection (1,5), (b) Fairness preserving (3,3) and (2,1), bias injection (1,5), (c) bias injection to the 5*th* partition in (d) Bias injection to the 9*th* partition in (5,5) while reserving fairness (3,3) on (3,3) and (2,1).



(a) (b)

FIGURE 18.8 Controlled partition-level bias injection: palm oil plantation (white) to forest (blue). (a) Ground truth and (b) Result by BI_{p*}.

to Figure 18.7, the gap between $\left\{\text{A-FAB}_p, \text{A-FAB}_p^*\right\}$ and {Base, FAIR} is smaller than that in the crop dataset. This is also due to the homogeneous nature of the plantations, that is, degrading the F1 performance on a specific partition may break the fairness on fairness-preserving partitionings (3, 3) and (2, 1). Finally, Figure 18.8 shows an example result of controlled partition-level bias injection by BI_p^* (highlighted a local region inside the 5th partition), where the 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 Figure 18.7(c)).

18.5 TIME-AWARE SPATIAL FAIRNESS

Due to temporal variations in environmental conditions across different locations, a fairness-enforced model trained on data from previous years may fail to maintain fairness when applied to testing data for subsequent years. This section introduces a physics-guided neural network model, which leverages the physical knowledge from existing physics-based models to guide the extraction of representative physical information and discover the temporal data shift across years (He et al., 2023).

18.5.1 Problem Formulation and Preliminary

Problem: The objective is to predict the county-level yield for corn in target years. For each county i, we are provided with input features within each year t, as $\mathbf{X}_{i,t} = \left\{\mathbf{x}_{i,t}^1, \mathbf{x}_{i,t}^2, \ldots, \mathbf{x}_{i,t}^D\right\}$, which are available at daily scales, that is, $D = 365$ in a non-leap year. The daily features $\mathbf{x}_{i,t}^d$ include weather drivers (e.g., precipitation, solar radiation), and soil and crop properties. The feature values are obtained as the average of the variable values from a set of randomly sampled farm locations in each county; see details in the Experiment Section (Section 18.5.4). Additionally, we have access to the crop yield labels $\mathbf{Y} = \{y_{i,t}\}$ from agricultural surveys in the training years $\mathcal{R}$. In the target testing years $\mathcal{T}$, we only have the input features but do not have the crop yield labels in the training process. In this problem, we use the predictive root mean square error (RMSE) as the performance metric, while fairness remains consistent with Equation 18.1.

In addition to the real crop yield dataset, we also run the physics-based Ecosys model (Zhou et al., 2021) to simulate crop yield. We use $\mathcal{S}$ to represent the set of locations and years (i, t) for which we have the simulated crop yield. Another benefit of the physics-based model is that it can also simulate

some intermediate physical variables in the crop growing process, such as variables involved in carbon and nitrogen cycling. It is noteworthy that physics-based models are often biased as they are necessarily approximations of reality due to incomplete knowledge or excessive complexity in modeling underlying processes. Hence, the simulated data can only be used for weak supervision.

Attention-based crop yield predictive model: The predictive model $\mathcal{F}_{\Theta}\left(\mathbf{x}_{i,t}\right)$ used in this work is based on an LSTM-Attention network. In this model, $\boldsymbol{\Theta}$ represents all the parameters in the network. Specifically, we first use an LSTM network to extract hidden representations at every time step (i.e., each date in a year), as $\mathbf{h}_{i,t}^{d=1:D} = \mathrm{LSTM}\left(\mathbf{x}_{i,t}^{d=1:D}\right)$. Then we create attention weights for each time step from its corresponding hidden representation via a linear transformation and a softmax function, as follows:

$$\alpha_{i,t}^{d} = \frac{\exp\left(\mathbf{w}_{\alpha} \cdot \mathbf{h}_{i,t}^{d} + b_{\alpha}\right)}{\sum_{d'} \exp\left(\mathbf{w}_{\alpha} \cdot \mathbf{h}_{i,t}^{d'} + b_{\alpha}\right)}, \tag{18.10}$$

where $\mathbf{w}\alpha \in \mathbb{R}^{D_h}$ and $\mathbf{b}\alpha \in \mathbb{R}^{1}$ are attention model parameters; hereinafter we use $\{\mathbf{w}_{*}, \mathbf{W}_{*}, b_{*}, \mathbf{b}_{*}\}$ to represent model parameters.

The embedding for each county i in year t can be obtained by the weighted mean over all the time steps using the attention weights, as:

$$\mathbf{e}_{i,t} = \sum_{d} \alpha_{i,t}^{d} \mathbf{h}_{i,t}^{d} \tag{18.11}$$

Finally, the model outputs the predicted yield value of the county i in year t as:

$$F_{\Theta}\left(\mathbf{x}_{i,t}\right) = \mathbf{w}_{y}\mathbf{e}_{i,t} + b_{y}. \tag{18.12}$$

The model can be trained by minimizing a mean squared error-based loss function, as follows:

$$\min_{\Theta} \mathcal{L}_{\mathrm{sup}} = \frac{\sum_{t \in \mathcal{R}} \sum_{i} \left(F_{\Theta}\left(\mathbf{x}_{i,t}\right) - y_{i,t}\right)^{2}}{N\left|\mathcal{R}\right|} \tag{18.13}$$

where N is the total number of counties.

Sample reweighting strategy: Sample reweighting strategy has been explored to reduce the gap of input space between the source and target domains (Bickel, Brückner, & Scheffer, 2007; Freedman & Berk, 2008). In our problem, a classifier $\mathcal{G}$ is trained to distinguish between source/training and target/testing years. The classifier $\mathcal{G}$ is implemented as a four-layer fully connected network. Its output is in the range of [0, 1], and is closer to 1 if it predicts the data to be more likely from the target years and otherwise is closer to 0. Then, the weight of each sample (e.g., county i in each year) is estimated as:

$$w_{i,t} = \frac{\mathcal{G}\left(\mathbf{e}_{i,t}\right)}{1 - \mathcal{G}\left(\mathbf{e}_{i,t}\right)} \tag{18.14}$$

After gathering the estimated sample weights, we normalize them to the range of $[\gamma, 1]$, where γ is a small value, for example, 0.1 in our test, which is used to ensure that all the samples are involved in the training process. We represent the normalized weights as $\overline{w}$.

The obtained weights can then be used in the training loss function to alleviate the temporal data shift in the training process, as follows:

$$\min_{\Theta} \mathcal{L}_{\text{rew}} = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T} \overline{w}_{i,t}\left(\mathcal{F}_{\Theta}\left(\mathbf{x}_{i,t}\right) - y_{i,t}\right)^2}{N|\mathcal{R}|} \tag{18.15}$$

18.5.2 Physics-Guided Sample Reweighting

We first build the PG-AN model to embed key variables involved in the carbon cycle and improve the prediction of crop yield (Figure 18.9). During the crop growing process, carbon is cycled through the atmosphere, crops, and soil. Carbon makes a major contribution to soil fertility and soil's capacity to retain water (Zhou et al., 2021). Carbon is absorbed by crops in the form of carbon dioxide, which contributes to the growth of crops. While the crops grow up, their produced roots and leaves also affect the soil carbon storage.

Although most variables in the carbon cycle are not observable, they can be simulated by existing physics-based models based on known physical theories. In this work, we use the physics-based Ecosys model (Zhou et al., 2021) to simulate three key variables in the carbon cycle, ecosystem autotrophic respiration (Ra), ecosystem heterotrophic respiration (Rh), and net ecosystem exchange (NEE). The entire carbon cycle can be captured by a mass conservation relation, as −NEE = GPP − Ra − Rh, where GPP represents the gross primary production, and can be estimated from remote sensing. The estimated GPP values are available over large regions and used as input to the predictive model.

Given the hidden representation $\mathbf{h}_{i,t}^{d}$ extracted by the LSTM-Attention model on each date d, we predict the physical variables Ra, Rh, and NEE using another transformation $\hat{\mathbf{q}}_{i,t}^{d} = f\left(\mathbf{h}_{i,t}^{d}\right)$, where $\hat{\mathbf{q}}$ represents the predicted values of [Ra,Rh,NEE] on the date d, and $f(\cdot)$ can be implemented as a fully connected network. By applying the model on the simulated data, we can compare the predicted $\hat{\mathbf{q}}$ and the simulated values $\mathbf{q}$ in each year, as follows:

$$\text{Diff}-_{\text{sim}_{i,t}} = \sum_{d} \| \hat{q}_{i,t}^{d} - q_{i,t}^{d} \|^2 \tag{18.16}$$

Given the GPP values, we also consider a penalty for violating the carbon mass conservation, as follows:

$$\text{MC}_{i,t} = \sum_{d}\left(\text{GPP}_{i,t}^{d} - \text{Ra}_{i,t}^{d} - \text{Rh}_{i,t}^{d} + \text{NEE}_{i,t}^{d}\right)^2 \tag{18.17}$$

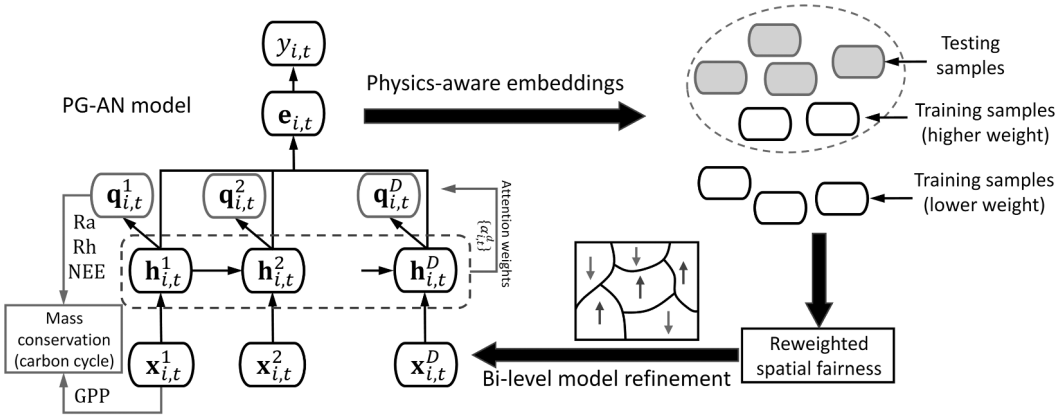


FIGURE 18.9 The overall flow of the proposed method. An LSTM-Attention network is used as a base model.

We then combine Diff-sim and mass conservation (MC) to define a physical loss. Here Diff-sim can be measured only on the simulated data. The MC can be measured on both simulated data and real data (both $\mathcal{R}$ and $\mathcal{T}$) using the predicted Ra, Rh, and NEE. The physical loss can be expressed as:

$$\begin{aligned}\mathcal{L}_{phy} = \beta_1 \frac{\sum_{(i,t)\in\mathcal{S}} \text{Diff} - \text{sim}_{i,t}}{|\mathcal{S}|} \\ + \beta_2 \left(\frac{\sum_{t\in\mathcal{R}\cup\mathcal{T}} \sum_i \text{MC}_{i,t}^2}{|\mathcal{R}\cup\mathcal{T}|N} + \frac{\sum_{(i,t)\in\mathcal{S}} \text{MC}_{i,t}^2}{|\mathcal{S}|} \right)\end{aligned} \tag{18.18}$$

where β_1 and β_2 are model hyper-parameters.

Finally, we optimize the model by combining the supervised loss (Equation 18.13) and the physical loss (Equation 18.18), as follows:

$$\mathcal{L}_{\text{PG-AN}} = \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{phy}} \tag{18.19}$$

We obtain the model $\mathcal{F}$ by minimizing the loss $\mathcal{L}_{\text{PG-AN}}$. We will then use the obtained embeddings e from this PG-AN model to estimate the weights $\{\overline{w}_{i,t}\}$ following Equation 18.14 and the normalization.

18.5.3 Fairness-Driven Model Refinement

After collecting the normalized weights $\overline{w}_{i,t}$, we refine the PG-AN model $\mathcal{F}$ to alleviate the temporal domain shift while preserving the spatial fairness. Note that the direct fine-tuning using the preliminary reweighted loss function (Equation 18.15) only reduces the temporal gap but may impair the spatial fairness. Hence, we propose a bi-level fairness-driven refining strategy for the PG-AN model that considers both the temporal data shift and the spatial fairness.

First, we modify the original fairness objective (Equation 18.1) by considering the similarity to the target dataset $\mathcal{T}$ based on the obtained sample weights $\overline{w}_{i,t}$. Each partition p contains training samples from multiple locations and multiple years. We will increase the weight for each sample (i, t) if the corresponding weight $\overline{w}_{i,t}$ is higher. This will be reflected in the performance measure $m_F(F_{\boldsymbol{\Theta}}, p)$ and the overall mean performance E_P in the fairness definition (Equation 18.1). In this work, we use the predictive RMSE as the performance metric, and the weighted performance on each partition p and the weighted overall performance can be computed as:

$$\begin{aligned}\tilde{m}_F\left(F_\Theta, p\right) &= \sqrt{\frac{\sum_{i\in p, t\in\mathcal{R}} \overline{w}_{i,t}\left(\mathcal{F}_\Theta\left(\mathbf{x}_{i,t}\right) - y_{i,t}\right)^2}{\sum_{i\in p, t\in\mathcal{R}} \overline{w}_{i,t}}} \\ \tilde{E}_P &= \sqrt{\frac{\sum_{i\in P, t\in\mathcal{R}} \overline{w}_{i,t}\left(\mathcal{F}_{\Theta_0}\left(\mathbf{x}_{i,t}\right) - y_{i,t}\right)^2}{\sum_{i\in P, t\in\mathcal{R}} \overline{w}_{i,t}}}\end{aligned} \tag{18.20}$$

Then we use the weighted performance measure to re-define the spatial fairness, as follows:

$$\tilde{m}_{\text{fair}} = \sum_{p\in P} \frac{d\left(\tilde{m}_F\left(F_\Theta, p\right), \tilde{E}_P\right)}{|P|} \tag{18.21}$$

Finally, the physics-guided neural network will be refined via a bi-level optimization process based on the reweighted fairness objective.

18.5.4 Experiments

18.5.4.1 Dataset

We use the corn yield data in Illinois and Iowa from the years 2000–2020 provided by the USDA National Agricultural Statistics Service (NASS).[1] In particular, there are in total 199 counties in our study region (100 counties in Illinois and 99 counties in Iowa). The corn yield data (in gCm^{-2}) are available for each county each year. The input features have 19 dimensions, including NLDAS-2 climate data (Xia et al., 2012), 0–30 cm gSSURGO soil properties,[2] crop type information, the 250 m Soil Adjusted Near-Infrared Reflectance of vegetation (SANIRv) based daily GPP product (Jiang, Guan, Wu, Peng, & Wang, 2021), and calendar year. Moreover, we use the physics-based Ecosys model (Zhou et al., 2021) to simulate Ra, Rh, NEE, and crop yield for 10,335 samples for the years 2001–2018.

In our experiments, we consider two major use cases for yield prediction, data reanalysis, and future prediction, and, hence, two testing scenarios are applied, using the years 2005–2006 and the last two years 2019–2020 as target testing years, respectively. In each testing scenario, the remaining years are used for model training. We also consider two different spatial partitionings. The first partitioning P_{199} treats each county as a spatial partition, and there are totally 199 partitions. The second partitioning P_{30} merges neighboring 6–10 counties as a partition, and contains in total 30 partitions. The number of counties in each partition varies across different partitions as we need to ensure each partition is continuous over space.

18.5.4.2 Experimental Design

We aim to answer several questions in our experiments:

1. **Can the proposed method outperform existing methods given the temporal data shift?** The proposed method is compared against multiple baselines, including the standard LSTM-Attention networks (LSTM-Attn), the adversarial domain adaptation methods (DA) (Ganin et al., 2016), the ADL (Alasadi et al., 2019), regularization-based fairness enforcement method (REG) (Kamishima et al., 2011; Yan & Howe, 2019), REG with the reweighting strategy (REG^{rew}), and self-training-based fairness enforcement method (Self-training) (An, Che, Ding, & Huang, 2022). All these methods use the base LSTM-Attn model but adopt different strategies for preserving fairness or addressing the temporal data shift. Among these methods, ADL and REG consider the fairness objective, DA considers the temporal data shift, and REG^{rew} and Self- training consider both. We also compare two methods that leverage simulated data for enhancing the LSTM-Attn model. As inspired by the prior work (Jia et al., 2021; Read et al., 2019), the first method SIM-ptr pre-trains the LSTM-Attn model using simulated yield data and then fine-tunes it using real data. The second method SIM-inp is trained using simulated data to predict Ra, Rh, and NEE, and then use them as additional input features. We also implement the SIM-inp method with the bi-level refinement ($SIM\text{-}inp^{ref}$). Finally, we evaluate two versions of the proposed method PG-AN (without using the bi-level refinement) and $PG\text{-}AN^{ref}$ (using the bi- level refinement). For each method, we measure the predictive RMSE and the spatial fairness (Equation 18.1 using the mean absolute distance) under two different partitionings, P_{199} and P_{30}.
2. **How will the performance change by adding sample weights and different levels of physical information?** We compare the performance of LSTM-Attn, LSTM-Attn + sample weights ($LSTM\text{-}Attn^{rew}$), LSTM-Attn + sample weights + pre-training using simulated yield ($LSTM\text{-}Attn^{rew+ptr}$), LSTM-Attn + sample weights + pre-training using simulated yield, Ra, Rh, NEE, and the mass conservation on these simulated variables and GPP

(LSTM-Attn$^{rew+phy}$), and LSTM-Attn + sample weights + training using simulated yield, Ra, Rh, NEE, the mass conservation on both simulated data and predicted values in real data, and real yield data (the proposed PG-AN model). We will also report the performance and fairness for each model either with or without using the bi-level fairness refinement.

3. **Can the bi-level fairness-driven refinement outperform other fairness enforcement methods?** We will incorporate the same level of physical information and sample weights for the REG and ADL methods to create two baselines PG-AN^{REG} and PG-AN^{ADL}. We then compare their performance with the proposed PG-AN^{ref} method.

18.5.4.3 Results

Performance comparison: Table 18.1 reports the performance of the proposed method and other baselines using different testing years and different spatial partitionings. It can be seen that the proposed methods (PG-AN and PG-AN^{ref}) outperform other methods by a decent margin in terms of both predictive RMSE and fairness measures. We also have several observations: (1) Compared to the base model LSTM-Attn, existing fairness enforcement methods (ADL, REG) only slightly improve the fairness in some testing cases and can even lead to degraded fairness when tested in the years 2005–2006. This is because they do not consider temporal data shifts across years. (2) The DA method generally produces worse performance compared to LSTM-Attn because it cannot extract informative embeddings for enforcing invariance in the adversarial learning process. (3) The methods using the simulated data (SIM-ptr, SIM-inp, and SIM-inp^{ref}) perform better than the base LSTM-Attn model and most of other baselines, which confirms the effectiveness of incorporating the simulated data. Moreover, SIM-inp performs better than SIM-ptr because it captures the intermediate physical variables in the carbon cycle. (4) The comparisons between SIM-inp and SIM-inp^{ref} and between PG-AN and PG-AN^{ref} show the effectiveness of the bi-level refinement in enhancing the fairness.

Figure 18.10 also shows the distributions of RMSE for each partition in P_{199} (i.e., each county) for the testing years 2005–2006 by the base LSTM-Attn model, the Self-training model, and the proposed PG-AN model. It can be clearly seen that the proposed method can effectively reduce the

TABLE 18.1
The Fairness and Overall RMSE with Two Different Partitionings for Two Testing Scenarios

Method	Testing Scenario 2019–2020				Testing Scenario 2005–2006			
	Partition	ing P_{30}	Partitioni	ng P_{199}	Partition	ing P_{30}	Partitioni	ng P_{199}
	RMSE	Fairness	RMSE	Fairness	RMSE	Fairness	RMSE	Fairness
LSTM-Attn	37.4284	11.0158	37.4284	16.8612	32.2486	8.9010	32.2486	13.6076
DA	38.0840	11.0802	38.0840	16.8274	32.2888	9.1424	32.0610	13.9750
ADL	38.6144	10.9396	38.2536	16.7950	32.2870	9.0022	32.1376	13.6252
REG	37.6738	10.9102	38.5752	16.7746	31.6602	8.9202	31.3974	13.5626
$_{REG}rew$	36.2342	10.4966	36.5012	16.2694	29.5366	8.6024	30.1106	13.0416
Self-training	35.6784	10.3912	35.9520	16.1510	31.0714	8.6522	31.0758	12.9724
SIM-ptr	36.0920	10.5758	36.0920	16.1400	30.8404	8.6258	30.8404	12.7468
SIM-inp	34.3598	9.8968	34.3598	15.9064	30.6056	7.8356	30.6056	12.6990
SIM-inp^{ref}	33.9332	9.5888	33.9892	15.4732	30.0814	7.3696	31.0480	12.1536
PG-AN	30.3688	7.8064	**30.3688**	13.6370	**24.7858**	6.6092	**24.7858**	10.2498
PG-AN^{ref}	**29.9558**	**7.2682**	30.9058	**12.5252**	25.7476	**5.7254**	25.3546	**9.8554**

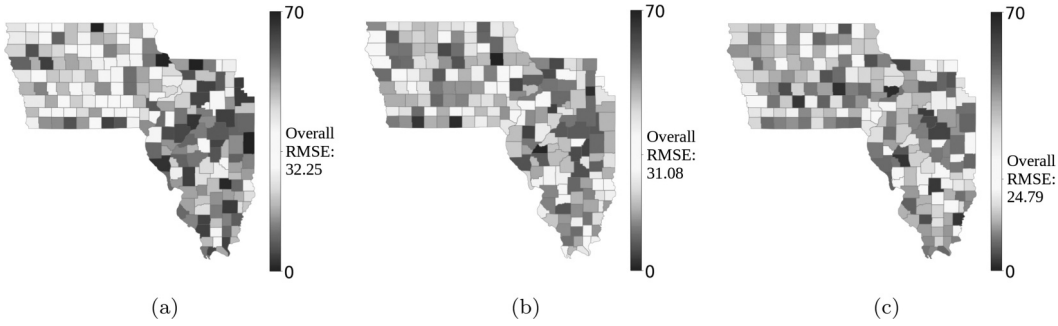


FIGURE 18.10 The distributions of predictive RMSE in 199 counties by three models for the testing years 2005–2006 and partitioning P_{199}. (a): The LSTM-Attn model. (b): The Self-training model. (c): The proposed PG-AN model.

RMSE for those counties that are poorly modeled by the LSTM-Attn method and the Self-training method. Also, the overall RMSE is significantly improved.

Ablation study: Figure 18.11 shows that the model performance and spatial fairness improve as we incorporate sample weights and more physical information. The PG-AN model performs better than LSTM-Attn$^{rew+phy}$ due to the gap between simulated and real data. Also, the bi-level refinement can always improve the spatial fairness for each model while maintaining a similar level of overall performance.

Effectiveness of bi-level training: Table 18.2 shows that the PG-AN model with the bi-level refinement achieves the best fairness without compromising the predictive RMSE performance. This is because the bi-level refinement mitigates the direct competition between predictive performance and spatial fairness, and avoids the selection of hyper-parameters.

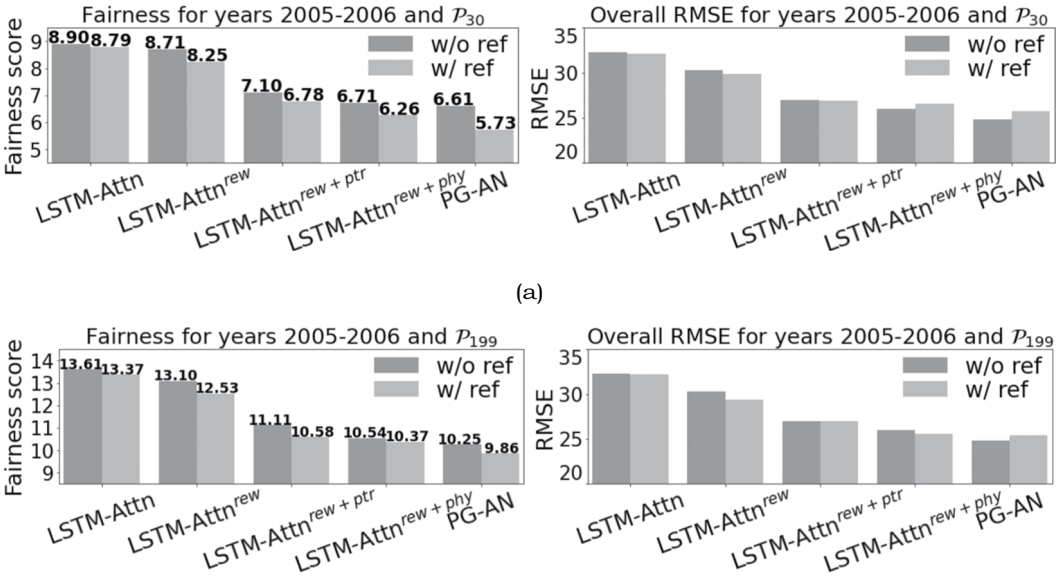


FIGURE 18.11 The performance change for the testing years 2005–2006. A higher fairness score indicates larger mean absolute distance values and worse fairness performance. (a) Fairness and predictive RMSE on P_{30}. and (b) Fairness and predictive RMSE on P_{199}.

TABLE 18.2
Comparison between the Bi-Level Refinement and Other Fairness Enforcement Methods for Refining the PG-AN Model

Method	Testing Scenario 2019–2020				Testing Scenario 2005–2006			
	Partitioni	ng P_{30}	Partitioni	ng P_{199}	Partitioni	ng P_{30}	Partitioni	ng P_{199}
	RMSE	Fairness	RMSE	Fairness	RMSE	Fairness	RMSE	Fairness
PG-AN	30.3688	7.8064	**30.3688**	13.6370	24.7858	6.6092	**24.7858**	10.2498
PG-AN^{ADL}	30.5730	7.7638	30.8136	13.5244	25.9296	6.3658	25.4034	10.2104
PG-AN^{REG}	**29.2328**	7.7154	31.5000	13.5916	**24.5180**	6.3400	25.4384	10.2210
PG-AN^{ref}	29.9558	**7.2682**	30.9058	**12.5252**	25.7476	**5.7254**	25.3546	**9.8554**

18.6 CONCLUSION

This chapter highlights the importance of fairness in the context of machine learning when dealing with spatial data. We conduct a review of recent developments in fairness-aware deep learning for spatial data, which aims to address the challenges posed by continuous and dynamic spatial partitionings, as well as the need to preserve fairness over time. Experiments demonstrate the effectiveness of these methods in enhancing spatial fairness while also keeping the quality of predictive outcomes. These advancements hold great promise in fostering equitable and robust machine learning solutions in many real-world applications of great societal relevance.

ACKNOWLEDGMENT

This work was supported by the NSF awards 2147195, 2239175, 2105133 and 2126474, the USGS awards G21AC10564 and G22AC00266, the NASA award 80NSSC22K1164, the Momentum award at the University of Pittsburgh, the DRI award at the University of Maryland, and the University of Pittsburgh Center for Research Computing.

NOTES

1 https://quickstats.nass.usda.gov/
2 https://gdg.sc.egov.usda.gov/

REFERENCES

Alasadi, J., Al Hilli, A., & Singh, V. K. (2019). Toward fairness in face matching algorithms. In *Proceedings of the 1st international workshop on fairness, accountability, and transparency in multimedia* (pp. 19–25).

An, B., Che, Z., Ding, M., & Huang, F. (2022). Transferring fairness under distribution shifts via fair consistency regularization. *arXiv preprint arXiv:2206.12796*.

Bailey, J. T., & Boryan, C. G. (2010). *Remote sensing applications in agriculture at the USDA national agricultural statistics service* (Tech. Rep.). Research and Development Division, USDA, NASS, Fairfax, VA.

Bickel, S., Brückner, M., & Scheffer, T. (2007). Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on machine learning* (pp. 81–88).

Boryan, C., Yang, Z., Mueller, R., & Craig, M. (2011). Monitoring us agriculture: the US department of agriculture, national agricultural statistics service, cropland data layer program. *Geocarto International*, *26* (5), 341–358.

CDL. (2017). *Cropland data layer - USDA nass*. https://www.nass.usda.gov/ResearchandScience/Cropland/SARS1a.php (Accessed: 03/20/2022).

Florida's supreme court has struck another blow against gerrymandering. (2015). Vox News. https://www.vox.com/2015/12/5/9851152/florida-gerrymandering-ruling

Freedman, D. A., & Berk, R. A. (2008). Weighting regressions by propensity scores. *Evaluation review*, *32* (4), 392–409.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., … Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, *17* (59), 1–35.

Ghosh, R., Ravirathinam, P., Jia, X., Khandelwal, A., Mulla, D., & Kumar, V. (2021). Cal- crop21: A georeferenced multi-spectral dataset of satellite imagery and crop labels. In *2021 IEEE international conference on big data (big data)* (pp. 1625–1632).

Group on earth observations global agricultural monitoring initiative. (2021). https://earthobservations.org/geoglam.php

Gunarso, P., Hartoyo, M. E., Agus, F., & others. (2013). RSPO, Kuala Lumpur, Malaysia. *Reports from the technical panels of the 2nd greenhouse gas working group of RSPO*.

He, E., Xie, Y., Jia, X., Chen, W., Bao, H., Zhou, X., …Ravirathinam, P. (2022). Sailing in the location-based fairness-bias sphere. In *Proceedings of the 30th international conference on advances in geographic information systems* (pp. 1–10).

He, E., Xie, Y., Liu, L., Chen, W., Jin, Z., & Jia, X. (2023). Physics guided neural networks for time-aware fairness: An application in crop yield prediction. In *AAAI conference on artificial intelligence*.

Jia, X., Willard, J., Karpatne, A., Read, J. S., Zwart, J. A., Steinbach, M., & Kumar, V. (2021). Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS Transactions on Data Science*, *2* (3), 1–26.

Jiang, C., Guan, K., Wu, G., Peng, B., & Wang, S. (2021). A daily, 250 m and real-time gross primary productivity product (2000–present) covering the contiguous United States. *Earth System Science Data*, *13* (2), 281–298.

Jo, E. S., & Gebru, T. (2020). Lessons from archives: Strategies for collecting sociocultural data in machine learning. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (pp. 306–316).

Kamilaris, A., et al. (2018). Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, *147*, 70–90.

Kamishima, T., Akaho, S., & Sakuma, J. (2011). Fairness-aware learning through regularization approach. In *2011 IEEE 11th international conference on data mining workshops* (pp. 643–650).

Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, *14* (5), 778–782.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, *54* (6), 1–35.

NASEM. (2018). *Improving crop estimates by integrating multiple data sources*. National Academies Press.

Nasr, M., & Tschantz, M. C. (2020). Bidding strategies with gender nondiscrimination constraints for online ad auctions. In *Proceedings of the 2020 conference on fairness, account- ability, and transparency* (pp. 337–347).

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. NPR News (Accessed: 03/20/2022).

Olofsson, P., Foody, G. M., Herold, M., Stehman, S. V., Woodcock, C. E., & Wulder, M. A. (2014). Good practices for estimating area and assessing accuracy of land change. *Remote Sensing of Environment*, *148*, 42–57.

Petersen, R., Goldman, E., Harris, N., Sargent, S., Aksenov, D., Manisha, A., et al. (2016). *Mapping tree plantations with multispectral imagery: preliminary results for seven tropical countries*. World Resources Institute.

Read, J. S., Jia, X., Willard, J., Appling, A. P., Zwart, J. A., Oliver, S. K., et al. (2019). Process-guided deep learning predictions of lake water temperature. *Water Resources Research*, *55* (11), 9173–9190.

Serna, I., Morales, A., Fierrez, J., Cebrian, M., Obradovich, N., & Rahwan, I. (2020). Sensitiveloss: Improving accuracy and fairness of face representations with discrimination-aware deep learning. *arXiv preprint arXiv:2004.11246*.

Steed, R., & Caliskan, A. (2021). Image representations learned with unsupervised pre-training contain human-like biases. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency* (pp. 701–713).

Sweeney, C., & Najafian, M. (2020). Reducing sentiment polarity for demographic attributes in word embeddings using adversarial learning. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (pp. 359–368).

Xia, Y., Mitchell, K., Ek, M., Cosgrove, B., Sheffield, J., Luo, L., et al. (2012). Continental-scale water and energy flux analysis and validation for North American land data assimilation system project phase 2 (NLDAS-2): 2. validation of model-simulated streamflow. *Journal of Geophysical Research: Atmospheres*, *117* (D03110). doi:10.1029/2011JD016051

Xie, Y., He, E., Jia, X., Bao, H., Zhou, X., Ghosh, R., & Ravirathinam, P. (2021a). A statistically-guided deep network transformation and moderation framework for data with spatial heterogeneity. In *2021 IEEE international conference on data mining (ICDM)* (pp. 767–776).

Xie, Y., Jia, X., Bao, H., Zhou, X., Yu, J., Ghosh, R., & Ravirathinam, P. (2021b). 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* (pp. 313–323).

Xie, Y., He, E., Jia, X., Chen, W., Skakun, S., Bao, H., …Ravirathinam, P. (2022). Fairness by "where": A statistically-robust and model-agnostic bi-level learning framework. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 36, pp. 12208–12216).

Yan, A., & Howe, B. (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* (pp. 552–555).

Yang, K., Qinami, K., Fei-Fei, L., Deng, J., & Russakovsky, O. (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* (pp. 547–558).

Zafar, M. B., Valera, I., Gomez Rodriguez, M., & Gummadi, K. P. (2017). Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web* (pp. 1171–1180).

Zhang, H., & Davidson, I. (2021). Towards fair deep anomaly detection. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency* (pp. 138–148).

Zhou, W., Guan, K., Peng, B., Tang, J., Jin, Z., Jiang, C., … Mezbahuddin, S. (2021). Quantifying carbon budget, crop yields and their responses to environmental variability using the ecosys model for us midwestern agroecosystems. *Agricultural and Forest Meteorology*, *307*, 108521.