Learning with Location-based Fairness: A Statistically-Robust Framework and Acceleration

Erhu He*, Yiqun Xie*, Weiye Chen*, Sergii Skakun, Han Bao, Rahul Ghosh, Praveen Ravirathinam, and Xiaowei Jia

Abstract—Fairness related to locations (i.e., "where") is critical for the use of machine learning in a variety of societal domains involving spatial datasets (e.g., agriculture, disaster response, urban planning). Spatial biases incurred by learning, if left unattended, may cause or exacerbate unfair distribution of resources, social division, spatial disparity, etc. The goal of this work is to develop statistically-robust formulations and model-agnostic learning strategies to understand and promote spatial fairness. The problem is challenging as locations are often from continuous spaces with no well-defined categories (e.g., gender), and statistical conclusions from spatial data are fragile to changes in spatial partitionings and scales. Existing studies in fairness-driven learning have generated valuable insights related to non-spatial factors including race, gender, education level, etc., but research to mitigate location-related biases still remains in its infancy, leaving the main challenges unaddressed. To bridge the gap, we first propose a robust space-as-distribution (SPAD) representation of spatial fairness to reduce statistical sensitivity related to partitionings and scales in continuous space. Furthermore, we propose a new SPAD-based stochastic strategy to efficiently optimize over an extensive distribution of fairness criteria, and a bi-level training framework to enforce fairness via adaptive adjustment of priorities among locations. Finally, we extend this framework with a similarity-based training strategy to improve the computational efficiency. Experiments conducted on two real-world problems, crop monitoring in the US and palm oil plantation mapping in Indonesia, show that SPAD can effectively reduce sensitivity in fairness evaluation and the stochastic bi-level training framework can greatly improve the fairness. Controlled experiments also show that similarity-based acceleration can greatly reduce the training time while keeping the prediction performance and fairness results at the same level.

Keywords—Spatial fairness, bi-level training, clustering, crop mapping.

1 Introduction

The goal of spatial fairness, or fairness by "where", is to reduce biases that have significant linkage to the locations or geographical areas of data samples. Such biases, if left unattended, may cause or exacerbate unfair distribution of resources, social division, spatial disparity, and weaknesses in resilience or sustainability [1].

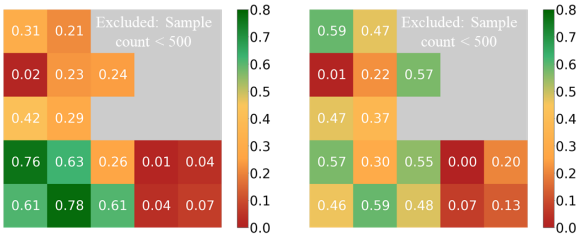
As an important example of societal impact, food production is witnessing tremendous supply stresses as a result of rapidly increasing population, climate change, etc. The urgency of the problem has led to major national and international efforts to monitor crops at large scales (e.g., G20's GEOGLAM global agriculture monitoring initiative), and these systems and alike heavily rely on both satellite Earth-observation imagery and learning methods [2]–[4]. More importantly, resulting products such as crop maps and acreage estimates [5] are used to inform critical actions (e.g., distribution of subsidies [6]–[8]), mitigate risks (e.g., natural disturbance incurred food shortage) and support local farmers, which are necessary for sustainability and stability. However, current products used to support these important decisions are largely subject to concerns on their fairness across locations as: (1) fairness has not been considered in the training process of the vast majority of these products; and (2) spatial data often follow heterogeneous patterns over locations [9], which can easily lead to prediction quality disparity without explicit intervention.

Fig. 1 shows the spatial distributions of the F1-scores achieved by a standard neural network model trained independently for two separate times (all settings are the same except random initial weights) for tomato classification using satellite imagery over an example region in Central Valley, California. The study area has a size of 80km by 80km, and is partitioned into 5 by 5 local regions. Another example in freshwater science is shown in Fig. 2. The water temperature is a master factor for water quality and is critical for many decision making processes in water management, e.g., water reservoir operations. We show the results of two different types of models, the data-driven long-short term memory (LSTM) model and the physics-based SNTemp model [10], in a subset of the Delaware River Basin, which supplies drinking water to a large population in the eastern coast of the US. As we can see from both examples, prediction accuracy in one region can be easily compromised to pursue better results at other places, which

- *E. He and X. Jia are with University of Pittsburgh. E-mail: {erh108, xiaowei}@pitt.edu*
- *Y. Xie, W. Chen and S. Skakun are with University of Maryland. E-mail: {xie, weiyec, skakun}@umd.edu*
- *H. Bao is with University of Iowa. E-mail: han-bao@uiowa.edu*
- *R. Ghosh and P. Ravirathinam are with University of Minnesota. E-mail: {ghosh128,pravirat}@umn.edu*

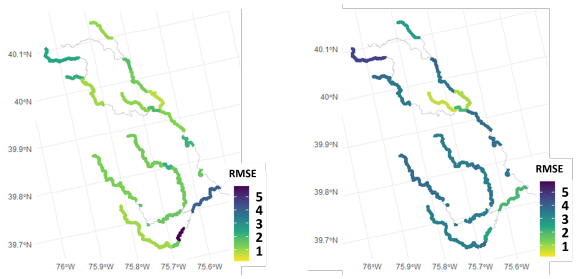
* *Equal contribution.*

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3371460



(a) Global F1: 0.573 (b) Global F1: 0.572

Fig. 1: Spatial bias examples. (a) and (b) show F1-scores of tomato classification by the same model (trained twice).



(a) LSTM (b) Process-based model

Fig. 2: Error distribution (in terms of root mean squared error (RMSE)) by (a) an LSTM model and (b) a process-based SNTemp [10] model for water temperature prediction over different river segments in a subset of the Delaware River Basin (the Christina River Watershed).

can be especially hurtful for disadvantaged groups, e.g., small holders representing the main production force behind minor crops [1], [11], [12]. This can also lead to unfair damage estimations (e.g., yield decrease) due to floods, drought, and hurricanes, which are often used to calculate farm insurance. Broadly, spatial fairness has important implications in decision-making across many domains, including disaster management (e.g., floods, wildfires), large-scale carbon monitoring which affects carbon tax, transportation (e.g., traffic and accident prediction, delivery estimation, demand forecast), and many more.

The formulation and enforcement of spatial fairness introduce several major challenges. First, unlike traditional categorical-attribute-based fairness (e.g., race or gender-based), spatial domain is a continuous space, which means the "categories" are not well-defined or given-for-free. Second, statistics (e.g., fairness scores based on variance) calculated from spatial datasets are fragile or sensitive to both the partition of space and scales, which is also known as the modifiable areal unit problem (MAUP; detailed in Def. 2). In other words, conclusions on "fair" or "unfair" can be easily altered by simple changes in spatial partitions or scales. The lack of consideration on MAUP has led to major societal concerns such as the recent debate on partisan gerrymandering at the US Supreme Court [13].

Despite the importance of spatial fairness for the use of deep learning in societal applications, research on this topic is still in its infancy and has barely been studied explicitly in the context of deep learning. Traditional line of research on fairness and equity in space mainly focuses on direct analysis over existing maps or their derivatives (e.g., COVID-19 statistics, access to resources) [14]–[16]. However, existing formulations and methods have yet to address the new challenges brought by spatial fairness, where conclusions can be easily flipped or manipulated due to statistical sensitivity introduced by MAUP. One possible way to address this challenge is by considering a larger number of spatial partitionings or scales during model training [17], [18]. However, as the number of spatial partitions or scales increases, it becomes increasingly challenging to ensure fairness across all scenarios. In addition to enhancing spatial fairness, the training time also escalates proportionally as the consideration of various space-partitionings and scales increases. This poses a significant challenge in terms of efficient optimization. Addressing this challenge is also crucial for the practical application of spatial fairness.

We aim to tackle the challenges by exploring new formulations and model-agnostic learning frameworks that are spatially-explicit and statistically-robust, i.e., the fairness is expected to be preserved across a set of different partitionings of the continuous space. Specifically, our contributions are:

- We propose a SPace-As-Distribution (SPAD) representation to formulate and evaluate spatial fairness in the context of continuous space, which mitigates the statistical sensitivity problems introduced by MAUP.
- We propose a SPAD-based stochastic strategy to efficiently optimize over an extensive distribution of candidate criteria for spatial fairness, which are needed to harness MAUP.
- We propose a bi-level player-referee training framework to enhance spatial fairness enforcement via adaptive adjustments of training priorities among locations.
- We extend the framework with a similarity-based training strategy, where SPAD candidates with similar fairness behaviors are dynamically clustered and sampled to reduce the computational cost.

Experiments on real datasets show that the proposed SPAD-based formulation and stochastic training can effectively promote fairness with improved robustness against MAUP-incurred sensitivity. The bi-level training also improves the stability of the model and fairness results compared to traditional regularization-based paradigms. The new similarity-based strategy can also accelerate the training process without undermining the overall performance and spatial fairness.

This paper is a significant extension of our previous conference paper [17]. We propose a similarity-based training strategy to accelerate the training process and improve the spatial fairness by selecting a small number

of representative partitionings for training based on the similarity amongst partitionings. We substantially extended the experiments with a new dataset and different base models (e.g., the LSTM model), and results show that the new extensions have a reduced training time compared to the previous method in the conference paper without undermining the overall performance and fairness.

2 Related Work

Existing fairness related work has explored a variety of techniques. One commonly used and generally applicable strategy is the regularization-based approach, which includes additional fairness-related losses during the training process [19]–[22]. Specifically, the model training incorporates fairness loss as a regularization term in the loss function . Note that the fairness measure used in this function needs to be end-to-end differentiable, amenable for training with back-propagation and updating ML model parameters. The resulting fairness-enforced model depends more on holistic task-relevant information, and conditions less on categorical attributes (e.g., race, gender, and age) at the same time. Another major direction of methods aims to learn group-invariant features [23], 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) [23]–[25]. Specifically, a predictor and an adversarial classifier are learned simultaneously. The goal of the predictor is to learn a high-level representation that is maximally informative for the major prediction task, while the role of the adversarial classifier is to minimize the predictor's ability to predict the sensitive attribute. From the data perspective, strategies have also been developed for data collection and filtering to reduce bias in downstream learning tasks [26]–[28]. More variations have also been discussed in a recent survey [29]. These methods have been applied to tasks where groups are well-defined by categorical attributes (e.g., face detection [22], text analysis [24], online bidding [30]). For spatial data, location-explicit frameworks [31], [32] have been developed to improve prediction performance over locations, but they do not consider fairness.

3 Key Concepts

Definition 1: **Partition p vs. Partitioning $\mathcal{P}$.** In this paper, a partitioning $\mathcal{P}$ splits an input space into m individual partitions p_i (Fig. 3), i.e., $\mathcal{P} = \{p_1, ..., p_i, ..., p_m\}$, where m is a variable.

Definition 2: **Modifiable Areal Unit Problem (MAUP).** MAUP states that statistical results and conclusions are sensitive to the choice of space partitioning $\mathcal{P}$ and scale. A change of scale (e.g., represented by the average area of $\{p_i \,|\, \forall p_i \in \mathcal{P}\}$) always infers a change of $\mathcal{P}$ but not vice versa. MAUP is often considered as a dilemma as statistical results are expected to vary if different aggregations or groupings of locations are used.

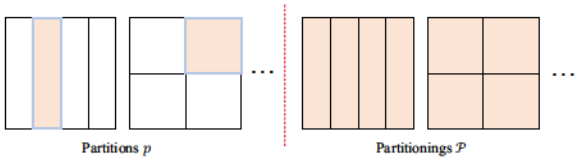


Fig. 3: Partition p vs. Partitioning $\mathcal{P}$.

Definition 3: **Fairness measure M_{fair}.** A statistic used to evaluate the fairness across a learning model's performance across several mutually-exclusive groups of individuals. For example, M_{fair} can be the variance of accuracy across groups. In this paper, groups are defined by partitions $p \in \mathcal{P}$.

Within the scope of this work, we consider partitionings $\mathcal{P}$ that follow a $s_1 \times s_2$ pattern (i.e., s_1 rows by s_2 columns). Fig. 4 shows an illustrative example of the effect of MAUP on spatial fairness evaluation. Fig. 4 (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 Fig. 4 (a2-3) and (b2-3), different partitionings or scales can lead to completely opposite conclusions, making fairness scores

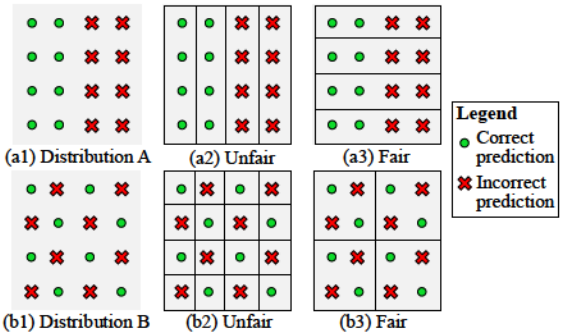


Fig. 4: Illustrative examples showing sensitivity to both space-partitioning and scale.

4 Formulation and Method

In this section, we first propose a novel space-as-distribution (SPAD) formulation to mitigate MAUP-incurred statistical sensitivity for fairness evaluation. Then, we propose a SPAD-based stochastic strategy as well as a bi-level training framework to enforce spatial fairness for an input deep network $\mathcal{F}$ selected by users. Finally, we propose a clustering-based sampling algorithm to accelerate the training process without the degradation of fairness performance.

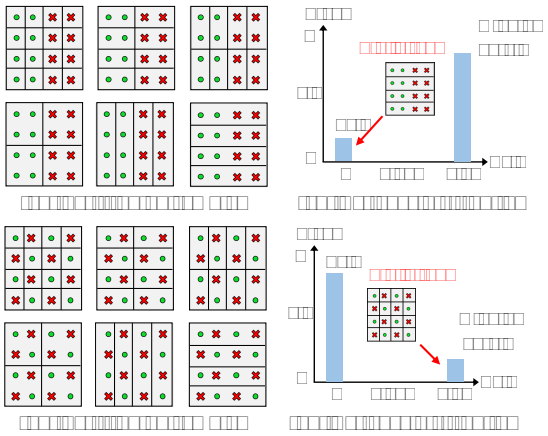


Fig. 5: Distributional representation by SPAD.

4.1 Space as a Distribution of Partitionings

As the grouping of locations is naturally needed for fairness evaluation using common performance metrics (e.g., precision, recall, accuracy), in the scope of this work we focus on scenarios where space-partitionings are used to generate location groups; in other words, each partition is analogous to a gender, race, etc. in related fairness studies. However, due to the MAUP dilemma (Def. 2), conclusions drawn from most – if not all – of common statistical measures are fragile to the variability in space-partitionings and scales. If this issue is ignored, then one may unintentionally or intentionally introduce additional bias (e.g., partisan gerrymandering [13]).

Thus, instead of relying on fragile scores calculated from a fixed partitioning or scale, we propose a SPace-As-Distribution (SPAD) representation to define spatial fairness. The idea is to go beyond a single partitioning or scale by treating space-partitionings at different scales as outcomes of a generative process governed by a statistical distribution. As mentioned in key concepts, in this work we consider partitionings that follow a pattern of rows by columns. So, in this case, an example generative process may follow a joint two-dimensional distribution where , , (e.g., 10). By default, one may assume a uniform distribution where (for equal-size partitioning). This scheme also allows users to flexibly impose a different distribution or prior, which may be dynamically adjusted based on intermediate results.

With the SPAD representation, spatial fairness becomes a distribution of scores, which can more holistically reflect fairness situations across a diverse set of partitionings and scales. As there may exist a large number of distinct partitionings (e.g., exponential to and for -partitioning with unequal-sized partitions), in practice, we may estimate the distribution using random samples of partitionings from the generative process. For example, Fig. 5 (a1) and (b1) show the same set of partitioning samples (different patterns and scales) overlaid on top of distributions A and B in Fig. 4, respectively. The variance of accuracy across partitions for all 6 partitioning samples is aggregated in Fig. 5 (a2) and (b2), where lower variance means fairer results. As we can see, with the distributional extension, the majority of scores reflect our expected results on the fairness evaluation for distributions A and B, and the partitioning samples leading to unexpected results become outliers (highlighted by red arrows).

Once a distribution of scores is obtained from the SPAD representation, summary statistics can be conveniently used for fairness evaluation based on application preferences (e.g., mean). Finally, the formal formulation of spatial-fairness-aware learning is defined as follows:

(1)

where is an input deep network; are the parameters of ; represents variables describing a space-partitioning (e.g., number of rows and columns for -partitionings) that are related to its probability as specified by a statistical distribution (e.g., uniform or user-defined); is a metric used to evaluate the performance of a model (e.g., F1-score); and is a fairness measure (loss) that is defined as:

(2)

where is a partition in (Def. 1), is a distance measure (e.g., squared or absolute distance), is the score (e.g., F1-score) of on 's training data, is the number of partitions in , and is another key variable, which represents the mean (expected) performance at each local partition . If has a large deviation from the mean (weighted or unweighted), it means the model is potentially unfair or biased across partitions. Finally, the mean here is calculated from a based model , where parameters are trained without any consideration of spatial fairness:

(3)

The benefit of using 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 (Eq. (1)) will increase if 's overall performance diverges too far from it (e.g., a model that yields 0 F1-scores on all partitions – which is fair but poor – will not be considered as a good candidate).

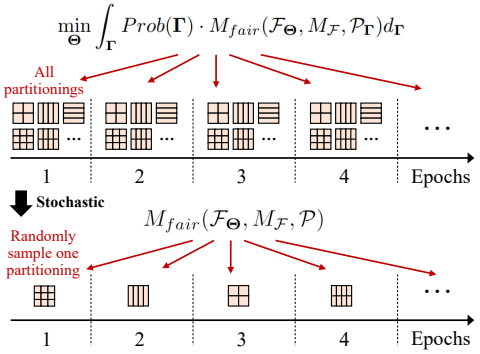


Fig. 6: SPAD-based stochastic training strategy.

4.2 SPAD-based Stochastic Training

A direct way to incorporate the distributional SPAD representation into the training process – either through loss functions or the bi-level method to be discussed in the next section – is to aggregate results from all the partitionings for each iteration or epoch. However, this is computationally expensive and sometimes prohibitive. For example, the number of possible partitionings can be exponential to data size (e.g., the number of sample locations) when general partitioning schemes are considered (e.g., arbitrary, hierarchical, or partitionings with unequal-size cells). Even for equal-size partitionings, there can be easily over hundreds of candidates when large and values (e.g., 10, 40, or more) are used for large-scale applications.

Thus, we propose a stochastic training strategy for SPAD to mitigate the cumbersome aggregation. Considering SPAD as a statistical generative process , in each iteration or epoch, we randomly sample a partitioning from and use it to evaluate fairness-related loss (Def. 3). For example, for equal-size partitionings, each time the generator may randomly sample (,) from a joint discrete distribution (Fig. 6). In this way, the probability of each partitioning (Eq. (1)) is automatically taken into consideration during optimization over epochs. In addition, in scenarios where the difficulty of achieving fairness varies for different partitionings, the SPAD-based stochastic strategy may accelerate the overall convergence. It may first help a subset of partitionings reach good fairness scores faster without the averaging effect, which may in turn help related partitionings to move out local minima traps. In practice, we have three further recommendations for implementation:

Unconstrained initial training: Ideally, we wish to maintain a high overall performance (e.g., F1-scores) while improving fairness across locations. However, it can be pre-mature to try to find a balance between the two objectives when the model still has a very poor overall performance (e.g., untrained). Hence, we keep fairness-related losses or constraints on-hold at the beginning, and optimize parameters by minimizing only prediction errors till stable.

Epoch as a minimum unit: Deep network training often involves mini-batches (i.e., a middle-ground between stochastic and batch gradient descent). As a result, the combined randomness of mini-batches and SPAD-based stochastic strategy may make the training unstable. Thus, using epoch as a minimum unit for changing partitioning samples can help reduce the superposed randomness.

Increasing frequency: Extending the last point, denote as the number of continuous epochs to train before a partitioning sample is changed. At the beginning of training, a biased model without any fairness consideration may need more epochs to make meaningful improvements, which means a larger (e.g., 10) is preferred. In contrast, towards the end of the training, a large can be undesirable as it may cause the model to overfit to a single partitioning at the finish. Thus, we recommend a decreasing (finally) during training.

4.3 Bi-level Fairness Enforcement

Next, we discuss the method to enforce fairness on each space partitioning sampled by the SPAD method. A traditional way to incorporate fairness loss (e.g., Eq. (2)) is to add it as a term in the loss function, e.g., , where is the prediction loss (e.g., cross-entropy or dice loss) and is a scaling factor or weight. This regularization-based formulation has three limitations when used for spatial-fairness enforcement: (1) 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 when calculating . (2) To reflect true fairness over partitions, metrics used in in Eq. (2) are ideally exact functions such as precision, recall or F1-scores. However, since many of the functions are not differentiable as a loss function (e.g., with the use of to extract predicted classes), approximations are often needed (e.g., threshold-based, soft-version), which introduce extra errors. Additionally, as such approximations are used to further derive fairness indicators (e.g.,), the uncertainty created by the errors can be quickly accumulated and amplified; and (3) The regularization term requires another scaling factor , the choice of which directly impacts final output and varies from problem to problem.

To mitigate these concerns, we propose a bi-level training strategy that disentangles the two types of losses with different purposes (i.e., and). Specifically, there are two levels of decision-making in-and-between epochs:

Partitioning-level ( ): Before each epoch, a referee evaluates the spatial fairness using Eq. (2) with exact metrics (e.g., F1-score); no approximation is needed as back-propagation is not part of the referee. The evaluation is performed on all partitions , guaranteeing the representativeness. Note

that the model is evaluatable for the very first epoch because the fairness-driven training starts from a base model, as discussed in the previous section and explanations for Eq. (2). Based on an individual partition p_i's deviation $d(M_{\mathcal{F}}(\mathcal{F}_{\boldsymbol{\Theta}}, p_i), E_{\mathcal{P}})$ (a summand in M_{fair}'s numerator in Eq. (2)), we assign its learning rate η_i for this epoch as:

$$\eta_i = \begin{cases} \frac{\eta'_i - \eta'_{min}}{\eta'_{max} - \eta'_{min}} \cdot \eta_{init}, & \text{if } \eta'_i > 0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$\eta'_i = \max(-(M_{\mathcal{F}}(\mathcal{F}_{\boldsymbol{\Theta}}, p_i) - E_{\mathcal{P}}), 0) \quad (5)$$

where η_{init} is the learning rate used to train the base model, $\eta'_{min} = \arg\min_{\eta'_i}\{\eta'_i \,|\, \eta'_i > 0, \forall i\}$, and $\eta'_{max} = \arg\max_{\eta'_i}\{\eta'_i \,|\, \forall i\}$.

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 (relatively 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 $[0, \eta_{init}]$ to keep the gradients more stable. This bi-level design also relieves the need for an extra scaling factor to combine the prediction and fairness losses.

- **Partition-level (p):** Using learning rates $\{\eta_i\}$ assigned by the referee, we perform regular training with the prediction loss $\mathcal{L}_{pred}$, iterating over data in all individual partitions $p_i \in \mathcal{P}$ in mini-batches.

4.4 Similarity-based Training Acceleration

The SPAD-based stochastic training needs to consider all the individual partitionings, which can still be time-consuming given a large number of partitionings. To address this issue, we propose a clustering-based sampling algorithm to accelerate the training process by selecting a smaller number of partitionings based on the similarity amongst partitionings.

Fig. 7 shows an illustrative example of changes in performance distributions, which make results in one partitioning fairer while the other fairer as well. This example consists of Fig. 7 (b) and (c), where the grids represent two different examples of space-partitionings, which are partitioning 1×4 and partitioning 1×2 respectively. The changes in performance distribution from the top to the bottom (Fig. 7 (a)) make the location-based fairness improve for partitioning 1×4, as shown as Fig. 7 (b). Meanwhile, they also introduce more fairness into partitioning 1×2, i.e., the first column and the second column have similar performance. This indicates the potential for enforcing the spatial fairness over a smaller set of representative partitionings to obtain the fairness over other partitionings. We formulate the sampling of representative partitionings in the training process as a clustering problem. In the following, we will describe similarity estimation, the clustering algorithm, clustering-based sampling, and implementation details.

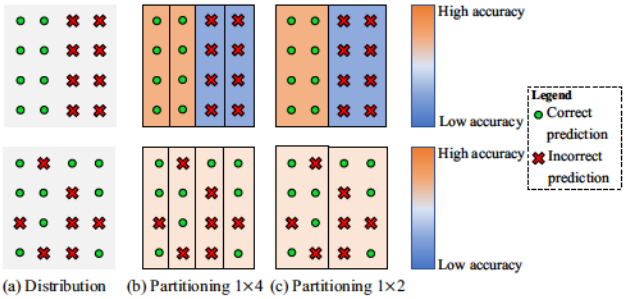


Fig. 7: Illustrative example showing similarity between two partitionings 1×4 and 1×2. Improving fairness on partitioning 1×4 can also guarantee the fairness for partitioning 1×2.

Similarity estimation: We create an N-by-N adjacency matrix Mat_{adj} to record the similarity relation between each pair of partitionings, where N denotes the total number of candidate partitionings. Here the similarity $Mat_{adj}[i, j]$ between each pair of partitionings $\mathcal{P}_i$ and $\mathcal{P}_j$ measures the fairness improvement on partitioning $\mathcal{P}_j$ when we enforce fairness on partitioning $\mathcal{P}_i$. A higher similarity value $Mat_{adj}[i, j]$ indicates a greater tendency for these partitionings to become fair simultaneously. Obtaining each row $Mat_{adj}[i, :]$ in the adjacency matrix requires running the bi-level fairness enforcement on partitioning $\mathcal{P}_i$ and then using the obtained model to evaluate the fairness over all the other partitionings. In particular, the similarity is computed as the proportion of fairness improvement, as follows:

$$Mat_{adj}[i, j] = \frac{M_{fair}(\mathcal{F}_{\boldsymbol{\Theta}_i^0}, \mathcal{P}_j) - M_{fair}(\mathcal{F}_{\boldsymbol{\Theta}_i^1}, \mathcal{P}_j)}{M_{fair}(\mathcal{F}_{\boldsymbol{\Theta}_i^0}, \mathcal{P}_j)} \quad (6)$$

where $\boldsymbol{\Theta}_i^0$ and $\boldsymbol{\Theta}_i^1$ denote the model parameters before and after we enforce the fairness on partitioning $\mathcal{P}_i$. Here we measure the proportion of the fairness improvement rather than the absolute improvement because the fairness values may have different scales for different partitionings. A positive similarity value (i.e., $Mat_{adj}[i, j] > 0$) indicates that the fairness enforcement on partitioning $\mathcal{P}_i$ also positively contributes to the fairness of partitioning $\mathcal{P}_j$, and otherwise indicates a negative influence. We further process the obtained matrix Mat_{adj} in three steps: (i) we assign 0 to all the negative entries; (ii) we replace each pair of entries $Mat_{adj}(i, j)$ and $Mat_{adj}(j, i)$ by their harmonic mean, as we prefer to establish a close connection between a pair of partitionings only if they have mutually positive influences; (iii) we convert the similarity matrix into a distance matrix for the clustering algorithm using $dist(i, j) = \frac{1}{1+Mat_{adj}[i,j]}$ for each entry.

Based on the distance matrix, we perform clustering to group partitioning with similar fairness behaviors (i.e.,

increase vs. decrease after an update). As the distances between partitionings may not be uniform for different clusters (i.e., partitionings' fairness behaviors can be more similar in certain clusters and less in the others), a preferable property of the clustering algorithm is to be able to identify clusters with varying densities. In addition, the clustering algorithm should not require the number of clusters, which is unknown for this problem. Thus, we use hierarchical density-based clustering, HDBSCAN [33]–[35], which is an integration of DBSCAN and OPTICS, which have the desired properties.

The HDBSCAN algorithm finds the clustering structure from the minimum spanning tree created using mutual reachability distances amongst partitionings instead of direct distances. The use of mutual reachability distance facilitates the clustering process by maintaining the distances between points inside clusters while increasing the distances between cluster points and susceptible noise points, which help avoid different clusters being detected as one due to bridges formed by noise points. In our integration, we additionally include a "spatial" prior to enlarge the distance between partitionings if their spatial patterns are highly different. For example, partitionings 1 5 and 1 6 are more spatially similar, whereas partitionings 1 5 and 5 1 are more different. We use spatial-overlap-based mutual information to measure this prior similarity. Specifically, the mutual information between two partitionings and is computed based on the maximum overlap between any partitions in and , as follows:

$$MI(\mathcal{P}_i, \mathcal{P}_j) = \Big(\sum_{p_a \in \mathcal{P}_i} \frac{\max_{p_b \in \mathcal{P}_j} (|p_a \cap p_b|)}{L} + \sum_{p_b \in \mathcal{P}_j} \frac{\max_{p_a \in \mathcal{P}_i} (|p_a \cap p_b|)}{L} \Big) / 2, \tag{7}$$

where represents the overlapping area between two partitions and , and denotes the total area of the study region, i.e., for any partitioning . Then the mutual information is used to rescale the distance as and the rescaled distance is used to create the minimum spanning tree in the HDBSCAN algorithm.

Partitionings selection: Once we obtain the clustering structure, we sample a subset of partitionings out of all the partitionings for enforcing fairness. The intuition is to directly optimize the fairness for a small number of representative partitionings, which stands a higher chance to positively contribute to the fairness of other partitionings. Specifically, we first determine the number of partitionings to be sampled from each cluster , as , where is the total number of clusters, and denotes the number of partitionings in the cluster . Each cluster will have at least one partitioning to be selected.

When sampling partitionings from each cluster , we prioritize partitionings based on how representative they are for the cluster and how frequently they were selected during past updates. In particular, we use the following two metrics for each partitioning, the sum of similarity to other partitionings in the same cluster and the number of times it has been selected in the previous training process. More formally, the probability of choosing a specific partitioning in the cluster can be computed via:

$$\frac{\exp \quad _k}{{}' \quad _k \exp \quad _k} \qquad \frac{\exp}{_k \exp} \tag{8}$$

where denotes the number of times the partitioning has been selected in the previous training process. Then we randomly sample partitionings from the cluster without replacement based on the obtained for .

Training process: Algorithm 1 depicts the whole training procedure using our similarity-based clustering. In our implementation, the whole training process has three stages, the initial stage, the clustering-sampling stage, and the sampling-only stage.

The initial stage covers the first loops of partitionings. The first loops aim to train an initial predictive model following the standard SPAD-based stochastic and bi-level training processes. The goal is to train a reasonably fair and accurate initial model so as to avoid any bias from a completely random model. Here we set in our test. The loop aims to initialize the similarity matrix by tentatively enforcing the fairness of each partitioning and measuring the contribution to the fairness of other partitionings. It is noteworthy that the model only gets temporarily updated in this loop, i.e., the model will be reset to the initial model obtained from the first loops after enforcing the fairness of each partitioning. The idea is to create the initial similarity matrix that is independent of the training order of partitionings.

The next clustering-sampling stage applies the HDBSCAN clustering algorithm and samples a small number of partitionings to be optimized for each loop. Due to the uncertainty of the stability and accuracy of the initial similarity matrix , additional updates are performed to enhance its reliability. Specifically, when a partitioning is selected and trained, the proportion of fairness improvement is calculated again using the Eq (6). Subsequently, is updated as the average value of all obtained proportions of fairness improvements after training (from the current and previous loops). This updating process ensures consistency among the partitionings with varying training frequencies during the clustering-sampling stage.

Compared to the original SPAD-based stochastic training process, this cluster-sampling process reduces the

time cost of iterating over all the partitionings but only optimizes the fairness for all the selected partitionings. The total number of selected partitionings is and it is often much smaller than . However, it needs to update the clustering structure in each loop, which requires an additional computational cost of lg . The total computational cost for each loop is lg , where the denotes the total number of locations for training, and is a constant value depending on the model complexity (e.g., the number of hidden neurons). In contrast, the standard SPAD-based stochastic training with the cost of for each loop. Hence, despite the clustering overhead, each loop in the clustering-sampling stage usually has a lower cost because .

In the last sampling-only stage, the clustering structure is already stable and requires no further adjustment based on our experiments. For the remaining training loops, we directly sample partitionings in every loop and train the model with the bi-level fairness enforcement on the selected partitionings. The time cost for each loop gets further reduced to .

Algorithm 1: Training process

Input: The set of training samples .
The set of candidate partitionings . The number of loops for the initial stage, clustering-sampling stage, and sampling-only stage: , , .

```
// Initial stage:
for to do
    repeat
        Randomly select a partitioning without replacement using the SPAD-based stochastic training strategy;
        Train the model with bi-level fairness enforcement on the selected partitioning;
    until all partitionings are selected;
Save the model ;
for to do
    Select a partitioning ;
    Train the model with the bi-level fairness enforcement to the partitioning ;
    Update the ;
    Reset the model to ;
// Clustering-sampling stage:
for to do
    Run the HDBSCAN algorithm for clustering;
    Sample partitionings from each cluster ;
    repeat
        Randomly select a partitioning from partitionings without replacement using the SPAD-based stochastic training strategy;
        Train the model with bi-level fairness enforcement on the selected partitioning;
        Update the corresponding row in the similarity matrix ;
    until all partitionings are selected;
// Sampling-only stage:
for to do
    Sample partitionings from each cluster ;
    repeat
        Randomly select a partitioning from partitionings without replacement using the SPAD-based stochastic training strategy;
        Train the model with bi-level fairness enforcement on the selected partitioning;
    until all partitionings are selected;
```

5 Dataset and Implementation Details

California crop mapping dataset: 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 for crop and land cover classification is the multi-spectral remote sensing data from Sentinel-2 in Central Valley, California (Sentinel tile T11SKA), and the study region has a size of 4096 4096 (6711 km 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. In particular, the Sentinel-2 data product has 13 spectral bands at three different spatial resolutions of 10, 20 and 60 metres. We leave out the atmospheric bands (Band 1, 9 and 10) of 60 metres resolution and re-sample all the bands to 20 metres. The label is from the USDA Crop Data Layer (CDL) [36]. Specifically, our experiment covers 18 land cover types, including a variety of crop types such as corn, cotton, sorghum, wheat, alfalfa, grapes, citrus, almond, walnut, pistachio, tomato, garlic, and others.

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 [37] and Tree Plantation [38]. 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% and 80% locations for training and testing respectively in our tests. However, in our subsequent tests to compare with the proposed SPAD method integrating clustering algorithm, we experiment with different sizes of training samples, which include 20%, 35%, 50%, and 65% locations for model training. Also, the remaining locations are used for model testing.

Implementation details: As mentioned in scope, we consider a set of partitionings denoted as , which represents multiple partitionings. In experiments, to allow comparisons with non-stochastic-based SPAD methods (computationally expensive), we set the maximum values for and to 5 and generate a set of partitionings , which leads to 24 different equal-size partitionings (the partitioning is excluded). In addition, to validate the speedup performance of integrating the clustering algorithm, we generate five other sets of equal-size partitionings , , , , , which leads to 15, 35, 63, 99 and 143 different equal-size partitionings.

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 model for 300 epochs (converged) without considering the fairness, using Adam ( ) as the optimizer. From this base model, we further implement different candidate approaches to improve fairness. Based on the stochastic training strategy, we sample a new partitioning in each epoch, and repeat this process over all the partitionings for 50 loops. Overall there are 50 expected epochs for each partitioning. Both weighted and unweighted F1-scores are considered as the performance metric in Eqs. (2) and (3). The unweighted F1-score is computed using the arithmetic mean of all the per-class F1-scores, treating all classes equally without considering class imbalance. On the other hand, the weighted F1-score takes into account the contribution of each class by weighting the average based on the number of examples in each class.

6 Experiments

Our experiments aim to answer the following questions:

Q1. Does the SPAD representation improve spatial fairness over different space-partitionings?

Q2. Does the bi-level training strategy improve over other fairness enforcement methods, such as regularization-based approaches and adversarial discriminating-based approaches?

Q3. Is the SPAD-based stochastic training able to maintain or improve fairness with a lower computational load?

Q4. What is the effect of the clustering-based sampling algorithm on the training efficiency, predictive performance, and the fairness performance?

The results to these questions can serve as an initial base for spatial-fairness driven learning. To answer these questions, we consider the following candidate methods:

Base: The base deep learning model (fully-connected DNN and LSTM) without consideration of spatial fairness.

REG: Spatial fairness is enforced using the SPAD representation by adding a regularization term to the loss function; the inclusion of a regularizer is a common strategy in related work [20], [21]. As F1-score is not differentiable, we use standard approximation via the threshold-based approach, which amplifies softmax predictions over a threshold to 1 and suppresses others to 0 using ReLU ReLU , where is a sufficiently large number (in our tests). The scaling factor for the regularizer is set to 5.

REG-Single: Spatial fairness is evaluated and improved using the baseline REG on a single space-partitioning . Specifically, our experiment targets partitioning (4, 1).

ADL-Single: This baseline is an extension of the discriminator-based fairness enforcing approach [23]. For fairness preservation, the model aims to learn group-invariant (or fair) features that make it difficult for a discriminator to identify the partition from which data samples come. Similar to REG-Single, our experiment uses partitioning (4, 1).

BL-Single: Spatial fairness is evaluated and improved using the proposed bi-level training strategy on a single space-partitioning . Same as REG-Single and ADL-Single, our experiment considers partitioning (4, 1).

SPAD-GD: This is the proposed SPAD method without using the stochastic strategy, i.e., it aggregates over gradients from all the candidate partitionings before making parameter updates in each loop.

SPAD-RND: This method combines SPAD with the random sampling strategy, i.e., it randomly samples partial partitionings for enforcing fairness. Specifically, for our experiment on partitionings , we randomly sample five partitionings in each training loop. For experiment on partitionings , we randomly sample ten partitionings in each training loop. These sample sizes (i.e., 5 in and 10 in) are selected to ensure that the model is trained with a similar number of partitionings as the SPAD method integrating the clustering-based partitioning sampling.

SPAD: This is the proposed approach using the SPAD representation with the stochastic and bi-level training strategies.

SPAD-SIM: This is the proposed approach with similarity-based acceleration. Specifically, at the initial stage, we train the model using the standard SPAD-based stochastic and bi-level training strategies at the

first 5 loops and then initialize the similarity matrix in the next loop. At the clustering-sampling stages, we update the clustering structure for another 5 loops. At the last sampling-only stage, we fix the clustering structure for the remaining loops.

6.1 California Crop Mapping Dataset

Comparison to existing fairness-preserving methods: We compare the fairness for partitionings achieved by SPAD, the base DNN model (without considering fairness) and the REG method in Fig. 8. For each partitioning (x-axis), we report mean absolute distance $_{\text{mean}}$, which is the mean of the absolute distances between F1-scores achieved on each partition and the average performance over all partitions ; both weighted and unweighted F1-scores are considered. In Tables 1 and 2, we summarize the overall performance (global F1-scores), the sum of mean absolute distance $_{\text{mean}}$ and the sum of maximum absolute distance $_{\text{max}}$ across all partitionings under two sets of partitionings and using weighted and unweighted F1, respectively. Specifically, we have: $_{\text{mean}}$ $_i$ $_i$ $\frac{\mathcal{F} \quad \Theta \quad \mathcal{P}_i}{i}$ and $_{\text{max}}$ $_i$ $_i$ $_i$. If $_{\text{mean}}$ or $_{\text{max}}$ is larger, it generally indicates larger $_{\text{mean}}$ or $_{\text{max}}$ values over different partitionings, and the curve will be higher in Fig. 8.

Fig. 8 shows that both SPAD and REG achieve a lower mean absolute distance compared to the base model over all space partitionings, which confirms the effectiveness of the SPAD representation in improving the fairness (**Q1**). Comparing SPAD and REG, we can see that SPAD consistently outperforms REG in the experiments (**Q2**), which shows that the bi-level design is more effective in enforcing spatial fairness than regularization terms by improving sample representativeness, allowing the use of exact metrics (i.e., no need to use approximations of F1-scores for differentiability purposes) and eliminating the need for an extra scaling factor for the regularizer which may add extra sensitivity.

From the first columns of Tables 1 and 2, we can see that SPAD is able to maintain a similar overall/global classification performance compared to the base DNN, which does not have any fairness consideration. Meanwhile, the second and third columns in the tables show that our method can significantly reduce the sums of mean and max absolute distance over all partitionings. This confirms that SPAD can effectively promote the fairness without compromising the classification performance.

In Fig. 9, we also show the absolute distances between F1-scores achieved on each partition and the average performance across all partitions for a sampled partitioning with six partitions, where each pair of bars in the figure represents the results of two methods for one partition. It can be seen that SPAD can achieve a more balanced distribution of F1-scores compared to the base model. This highlights how SPAD can positively influence fairness and provides evidence of the effectiveness of the proposed method in improving fairness.

TABLE 1: Classification and fairness results on crop mapping by weighted F1-scores (denotes "timeout", which means model training requires more than 18 hours.)

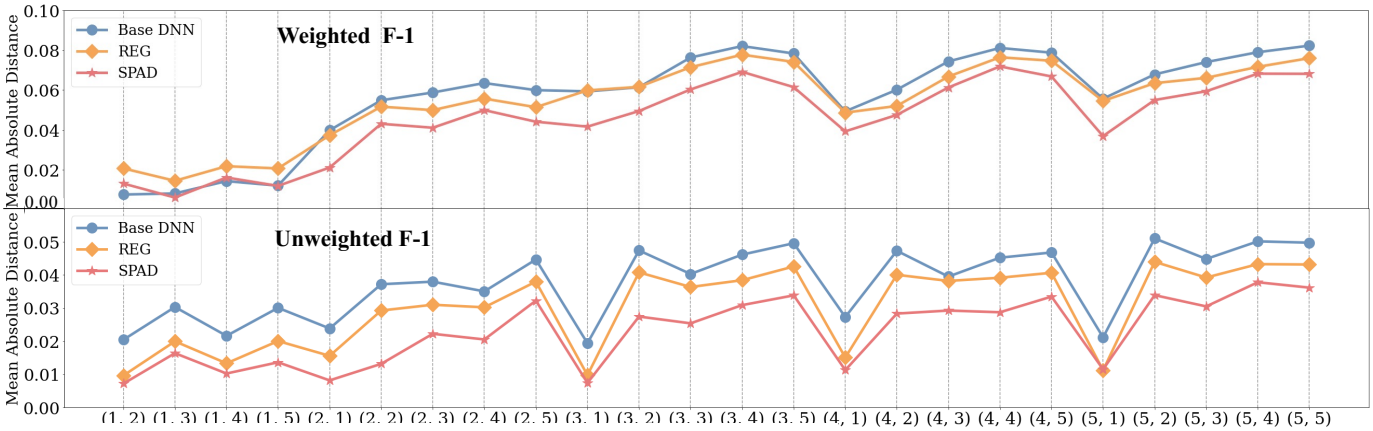
Method	$\mathcal{P}'$ (24 partitionings)			$\mathcal{P}$ ' (99 partitionings)		
	W. F1	mean	max	**W. F1**	mean	max
Base DNN	0.572	1.379	3.799	0.572	7.585	23.680
REG-Single	0.567	1.366	3.718	0.567	7.543	23.359
ADL-Single	0.572	1.358	3.748	0.572	7.459	23.442
BL-Single	0.558	1.355	3.712	0.558	7.400	22.710
REG	0.566	1.319	3.821	0.567	7.274	23.274
SPAD-GD	0.573	1.275	3.571			
SPAD-RND	0.564	1.308	3.541	0.572	7.006	23.123
SPAD	0.573	**1.102**	3.190	0.570	6.700	**21.880**
SPAD-SIM	0.565	1.163	**3.033**	0.576	**6.686**	21.937

TABLE 2: Classification and fairness results on crop mapping by unweighted F1-scores (denotes "timeout", which means model training requires more than 18 hours.)

Method	$\mathcal{P}'$ (24 partitionings)			$\mathcal{P}$ ' (99 partitionings)		
	UW. F1	mean	max	**UW. F1**	mean	max
Base DNN	0.377	0.906	1.808	0.377	4.202	10.068
REG-Single	0.376	0.884	1.774	0.376	4.135	9.928
ADL-Single	0.375	0.814	1.775	0.375	3.855	9.707
BL-Single	0.374	0.733	1.553	0.374	3.695	9.251
REG	0.376	0.728	1.691	0.376	3.544	9.629
SPAD-GD	0.372	0.602	1.384			
SPAD-RND	0.376	0.706	1.498	0.375	3.578	9.190
SPAD	0.374	**0.549**	**1.337**	0.372	**3.047**	8.764
SPAD-SIM	0.372	0.595	1.349	0.368	3.136	**8.689**

Comparison to partitioning-specific methods: We aim to verify that SPAD can achieve better fairness over the majority of the partitionings compared to non-SPAD-based variants that only optimize fairness over a specific spatial partitioning using different fairness enforcement methods. Fig. 10 shows the fairness performance of partition-specific methods REG-Single, ADL-Single, and BL-Single. The overall trend is that SPAD achieves better spatial fairness in most partitionings by modeling space-partitionings as a distribution (**Q1**). Meanwhile, we can see that BL-Single obtains a better fairness result for its target partitioning (i.e., the partitioning (4,1)). However, its fairness improvements are limited for other partitionings. This conforms to the expectation that partitioning-specific methods are able to reach further improvements on a given , but cannot generalize well to the others. Interestingly, we can also observe that BL-Single can produce better fairness on certain partitionings. For example, in this test, BL-Single also achieves good fairness on partitionings (2, 1), (3, 1) and (5, 1) because these partitionings are more structurally similar to the target

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3371460



nd the base model over all the partitionings.

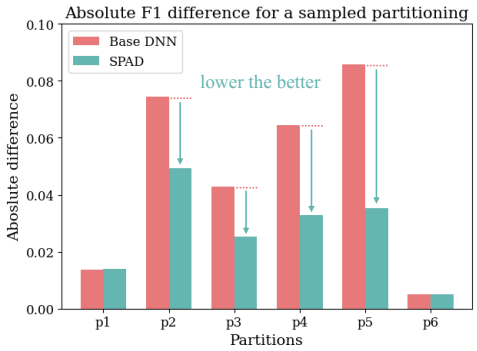


Fig. 9: The absolute difference between the obtained F1-scores over different partitions and the average F1-score for crop mapping.

partitioning (4, 1). This result confirms that the similarity relationships amongst partitionings can be leveraged to further improve the efficiency of the stochastic sampling process, as used in our proposed clustering-based approach. Tables 1 and 2 (row 4) show the weighted and unweighted F1-scores achieved by BL-Single. This method has similar global F1-scores with SPAD since our design takes the overall performance into account (Eqs. (2) and (3)). However, BL-Single produces larger values of $_{mean}$ and $_{max}$ compared to SPAD, which again confirms the benefits of the SPAD representation.

In addition, Fig. 10 as well as Tables 1 and 2 (rows 2-4) show that the model using the bi-level training strategy achieves the best fairness on target partitioning without compromising the global F1-scores (**Q2**). This is because the bi-level fairness enforcement mitigates the direct competition between predictive performance and spatial fairness, and also avoids the selection of hyper-parameters.

In both the experiments with weighted and unweighted F1-scores (Fig. 10), SPAD can often get very close to the fairness scores achieved by partitioning-specific methods on their sole-input . This shows the potential dependency relationships between partitionings. We also explored a variant that uses only finer or finest-scale partitioning such as (5, 5). One issue we observed is that the method faces difficulty in convergence, leading to poorer results on both fine and coarse scales. This is potentially due to the fact that fairness enforcement at finer-scale naturally leads to stricter criteria.

Validation of stochastic training strategies: Next, we validate the effectiveness of the SPAD-based stochastic training strategy (**Q3**). We first compare SPAD to the SPAD-GD approach, which is the standard gradient descent method across all the partitionings. Specifically, this approach loops over all the partitionings in each iteration, aggregates their gradients, and uses the aggregated gradients to update model parameters. Compared to our SPAD-based stochastic approach, the aggregation in SPAD-GD leads to a heavier computational load and requires longer time for model training (i.e., 2.5 hours vs. 9.5 hours for partitionings using NVIDIA Tesla K80 GPU over two runs). Here we maintain the same number of parameter updates for the two methods, and the only difference is that each SPAD update is made by gradients from a sampled partitioning whereas each SPAD-GD update uses average gradients from all partitionings. Fig. 12 shows their performance comparison. We can see that the two methods have about the same performance for the unweighted scenario (the lower part of Fig. 12), which is expected. Interestingly, SPAD outperforms SPAD-GD in the weighted scenario (the upper part of Fig. 12). One reason is that the added randomness from the stochastic sampling in SPAD may allow a better chance for the training to move out of local minima traps without the averaging effects, especially when fairness is harder to achieve at the beginning for some partitionings.

Effectiveness of integrating clustering algorithm: Finally, we verify the effectiveness of SPAD-SIM, which uses the clustering obtained by the HDBSCAN algorithm to sample partitionings during the model training (**Q4**).

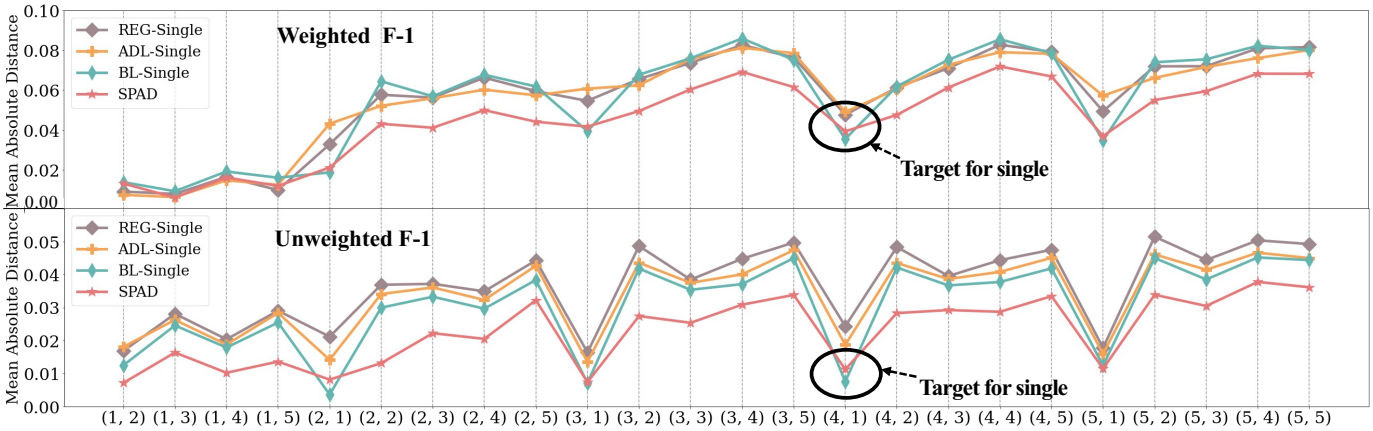


Fig. 10: Fairness comparison on crop mapping amongst SPAD, REG-Single, ADL-Single, and BL-Single over all the partitionings.

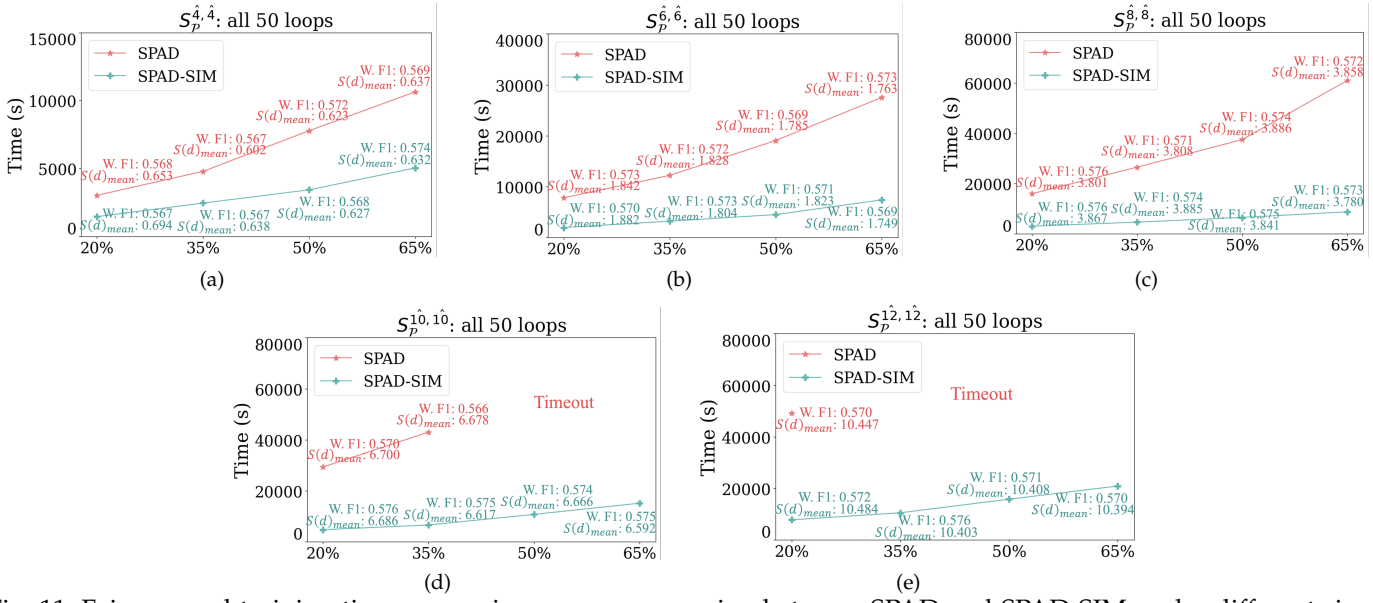


Fig. 11: Fairness and training time comparison on crop mapping between SPAD and SPAD-SIM under different sizes of training samples and different sets of partitionings across all 50 loops.

To ensure a fair comparison, we train 50 training loops for all sampled partitionings in both the SPAD and SPAD-SIM approaches. In our tests, we notice that the models converged within these 50 loops.

We first validate the efficiency of SPAD-SIM under different scenarios. We evaluate the performance and training time consumption of SPAD and SPAD-SIM under different sizes of training samples and different sets of partitionings in Fig. 11. For each size of training samples (x-axis), we report the training time across all 50 loops. Also, the global weighted F1-score and the sum of mean absolute distance $_{\text{mean}}$ are grouped as a pair and presented for each testing scenario. Note that "timeout" means model training requires more than 18 hours when using the NVIDIA Tesla K80 GPU.

Fig. 11 shows that SPAD-SIM can have a shorter time for model training compared to the method SPAD without undermining the overall performance and fairness. With an increasing number of samples or partitioning, SPAD-SIM leads to a higher speedup for model training. This confirms that training the model using a representative subset of partitionings selected by the clustering-based approach is sufficient to promote the spatial fairness over all partitionings.

We further compare the performance of SPAD-SIM to two methods SPAD and SPAD-RND. Fig. 12 shows that both SPAD and SPAD-SIM achieve similar mean absolute distances over all space partitionings, and they consistently outperform SPAD-RND. When considering the overall fairness performance over all the candidate

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3371460

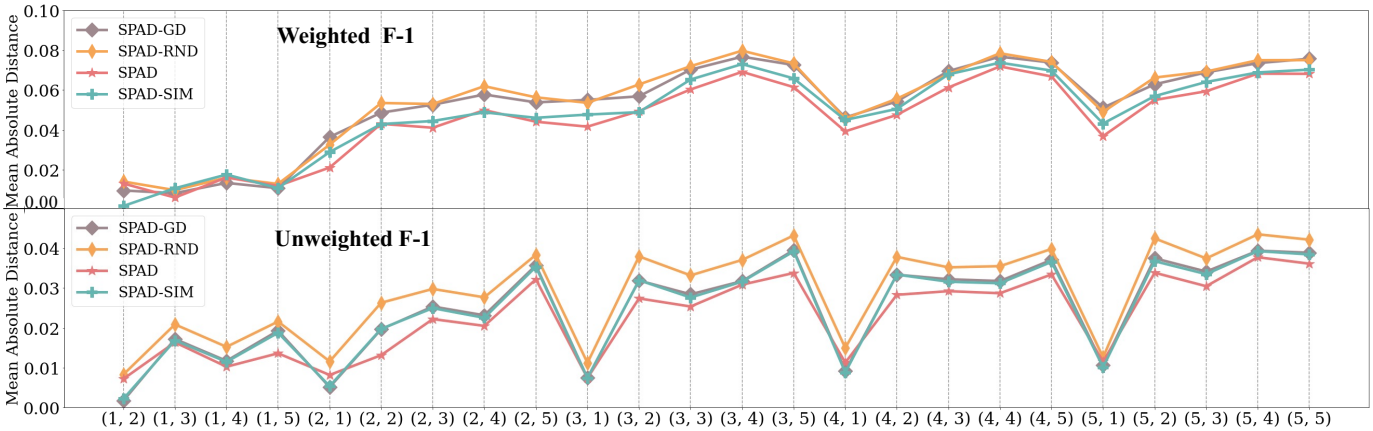


Fig. 12: Fairness comparison on crop mapping amongst SPAD, SPAD-SIM, SPAD-RND, and SPAD-GD over all the partitionings.

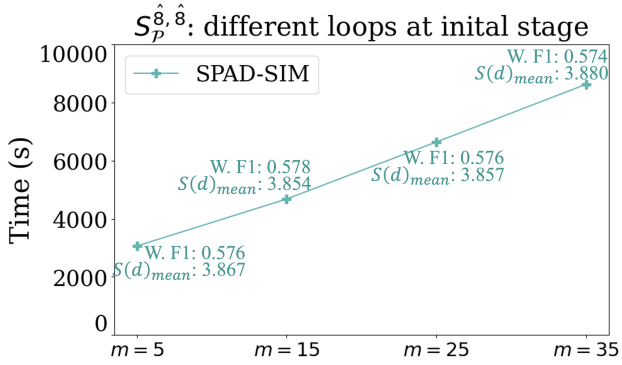


Fig. 13: Fairness and training time comparison on crop mapping by SPAD-SIM with different loops at the initial stage.

TABLE 3: Classification and fairness results of DNN-based model on plantation mapping

Method	$\mathcal{P}'$ (24 partitionings)		
	F1	mean	max
Base DNN	0.648	4.630	8.615
REG-Single	0.650	4.303	7.959
ADL-Single	0.630	4.330	7.984
BL-Single	0.652	4.301	7.931
REG	0.647	4.254	7.927
SPAD-GD	0.647	4.232	7.803
SPAD-RND	0.646	4.251	7.901
SPAD	0.640	**4.166**	**7.783**
SPAD-SIM	0.637	4.170	7.833

partitionings, Table 1 and 2 (rows 7-9) show that SPAD-SIM has a smaller mean absolute distance and maximal absolute distance than SPAD-RND, and can achieve a similar level of spatial fairness and overall performance (in F-1) with SPAD. The superiority of SPAD-SIM over SPAD-RND is because SPAD-SIM can extract more representative partitionings for model training.

In addition, the proposed SPAD-SIM method has the initial stage to iterate all sampled partitionings in order to initialize an unbiased and reasonable similarity matrix for clustering. We also evaluate the fairness performance and training time with varying loops for the initial stage, as shown in Fig. 13. Even though the total number of training loops is fixed as 50, the time required for model training is growing as the number of loops increases at the initial stage, which is expected. Also, it shows that SPAD-SIM under our setting ( ) is sufficient to generate an unbiased similarity matrix and produce a good fairness result.

In the end, Fig. 14 shows a sequence of clusters for partitionings for the method SPAD-SIM at the clustering-sampling stage. It can be clearly seen that the partitionings that have higher similarities are grouped together. For example, partitionings (1, 2) and (1, 4) are grouped into the same cluster. Also, it shows the stability of using the HDBSCAN clustering algorithm. The clusters keep stable with only a few partitionings switching between clusters each time, when starting clustering at the clustering-sampling stage.

To further augment our analysis, we introduce an additional experiment focusing on the training time efficiency of SPAD and SPAD-SIM in achieving a near-optimal performance level for the first time during the training process. Fig. 15 shows the performance and training time consumption of SPAD and SPAD-SIM under different sizes of training samples and different sets of partitionings. This further highlights that SPAD-SIM excels in terms of training time efficiency. Even when considering varying training sample sizes and diverse partitioning scenarios, SPAD-SIM consistently exhibits a faster convergence towards near-optimal performance compared to SPAD.

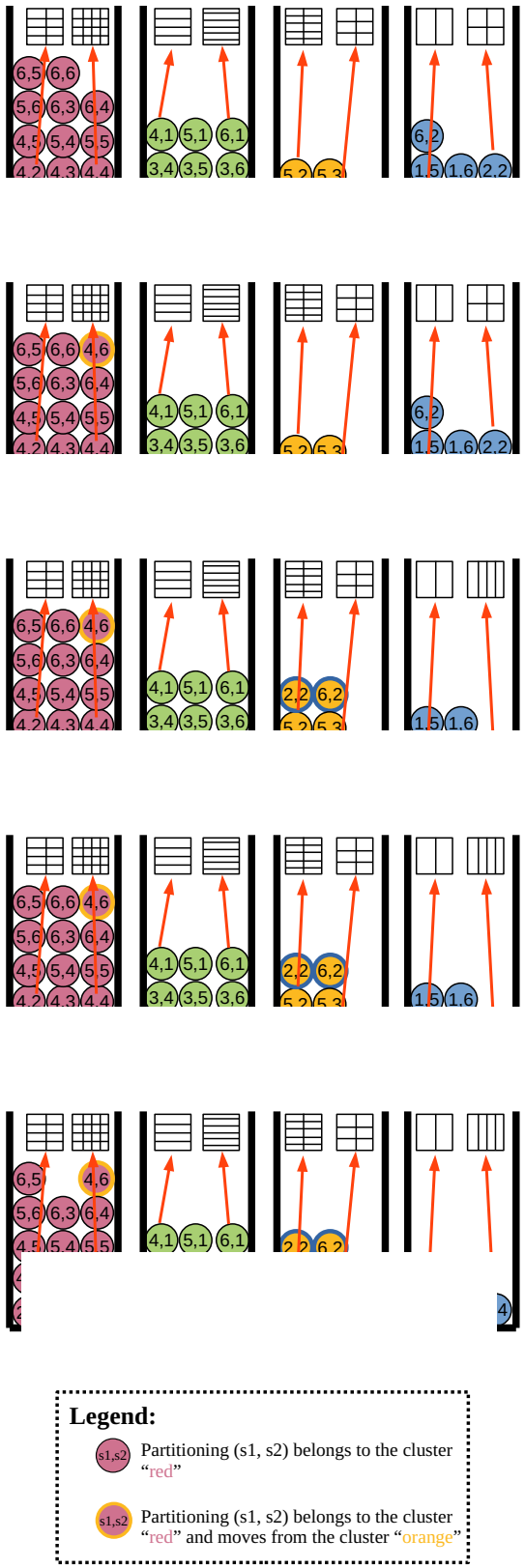


Fig. 14: The sequence of generated clusters on partitionings by SPAD-SIM for each loop at the clustering-sampling stage.

TABLE 4: Classification and fairness results of LSTM-based model on plantation mapping

Method	$\mathcal{P}$ (24 partitionings)		
	F1	mean	max
Base LSTM	0.804	2.587	6.859
REG-Single	0.807	2.398	6.235
ADL-Single	0.818	2.403	6.217
BL-Single	0.812	2.374	6.037
REG	0.824	2.262	6.161
SPAD-GD	0.827	2.143	5.370
SPAD-RND	0.816	2.317	6.031
SPAD	0.819	**2.139**	**5.270**
SPAD-SIM	0.826	2.140	5.404

6.2 Palm Oil Plantation Mapping Dataset

DNN-based model performance: In our evaluation of palm oil plantation mapping, we conduct the same tests on the dataset and observe consistent results. We perform a comparative analysis of model performance and fairness across partitionings obtained by SPAD, SPAD-SIM, and other baseline methods. Note that there is no distinction between weighted and unweighted F1-scores in the context of binary classification.

Table 3 presents the overall performance (global F1-scores), the sum of mean absolute distance $_{\text{mean}}$ and the sum of maximum absolute distance $_{\text{max}}$ for the set of partitionings . We can see that SPAD maintains the same level of overall/global classification performance as the base DNN, which does not have any fairness consideration. Meanwhile, SPAD achieves significant reductions in both the sums of mean and maximum absolute distances across all partitionings compared to other baseline methods. These results confirm the effectiveness of SPAD in promoting fairness without compromising classification performance. Additionally, the fairness gap between BL-SINGLE, solely trained with partitioning (4, 1), and SPAD is smaller than that in the California crop mapping dataset. This is because the palm oil plantations in this dataset are relatively homogeneous over space and thus improving the fairness on certain partitioning could easily promote the fairness over other partitionings.

Then, we verify the effectiveness of SPAD-SIM. Same as the experiments conducted on the California crop mapping dataset, we evaluate the classification performance and training time consumption of SPAD and SPAD-SIM for partitionings under different sizes of training samples. This evaluation is twofold: Fig. 16 presents a comprehensive comparison of fairness and training time across all 50 loops, whereas Fig. 17 focuses on the comparison at the first achievement of good performance during the training process. It can be seen that SPAD-SIM still achieves shorter times for model training compared to the method SPAD while preserving the overall prediction performance and fairness.

Finally, we also evaluate the fairness performance and

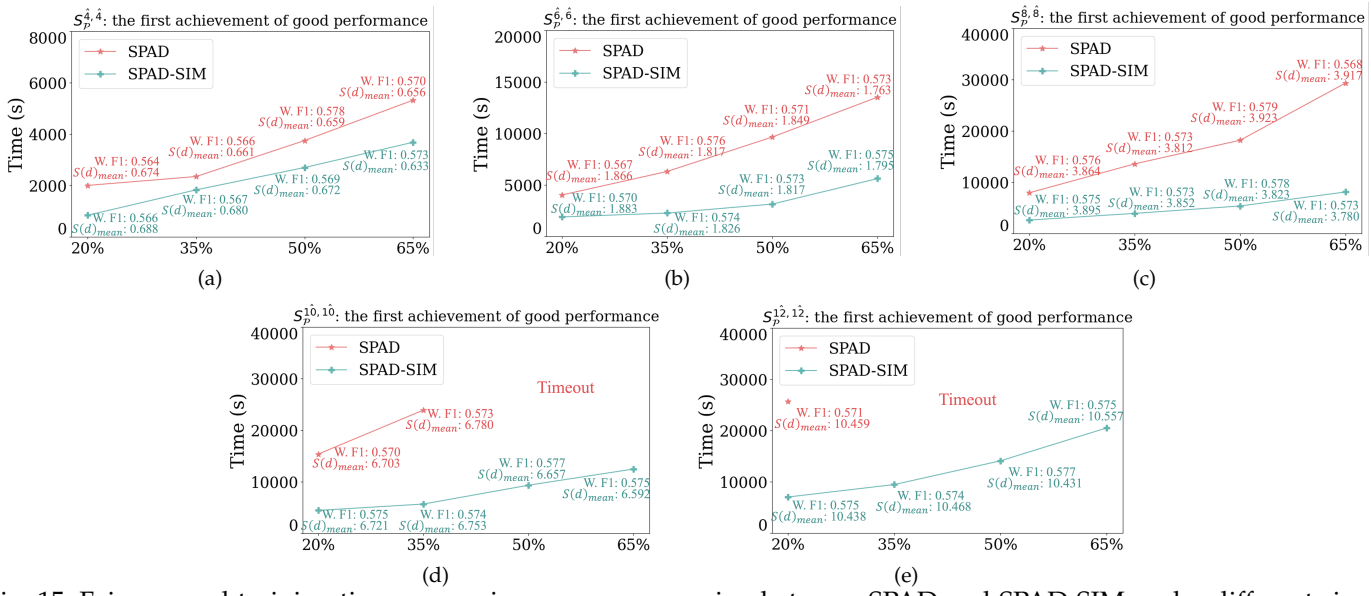


Fig. 15: Fairness and training time comparison on crop mapping between SPAD and SPAD-SIM under different sizes of training samples and different sets of partitionings at the first achievement of good performance.

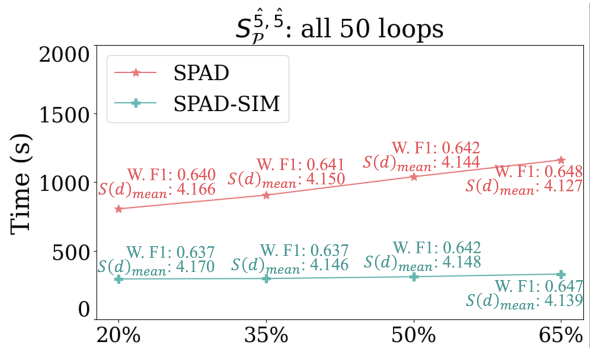


Fig. 16: Fairness and training time comparison on plantation mapping between SPAD and SPAD-SIM under different sizes of training samples across all 50 loops.

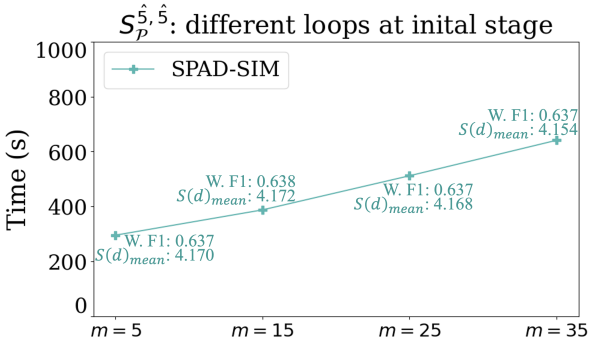


Fig. 18: Fairness and training time comparison on plantation mapping by SPAD-SIM with different loops at the initial stage.

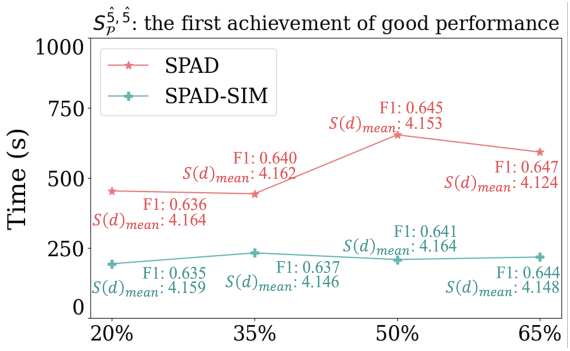


Fig. 17: Fairness and training time comparison on plantation mapping between SPAD and SPAD-SIM under different sizes of training samples at the first achievement of good performance.

training time by varying the number of loops for the initial stage, as presented in Fig. 18. It shows that SPAD-SIM under our setting (m = 5) is capable of generating an unbiased similarity matrix and producing a good fairness result.

LSTM-based model performance: In addition to evaluating our method with the DNN-base model, we conduct experiments using LSTM as the base model to assess the generalizability of our approach across different network architectures. The results are presented in Table 4 and depicted in Fig. 19, 20 and 21. We can see that the F1 performance of LSTM is generally better than DNN as LSTM is more likely to capture palm oil plantations from a series of data. Despite the architectural differences, the comparison of classification performance and fairness exhibits consistent patterns as observed in the previous experiments using DNN as the base model. Additionally,

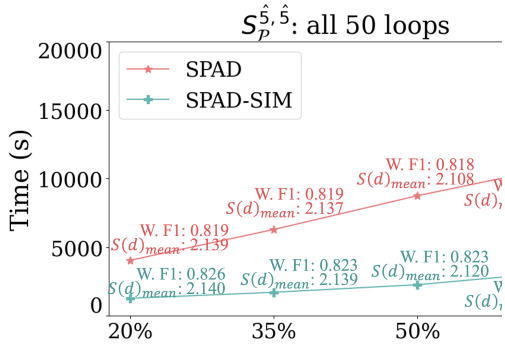


Fig. 19: Fairness and training time comparisc tation mapping between SPAD and SPAD-S different sizes of training samples across all 50 loops.

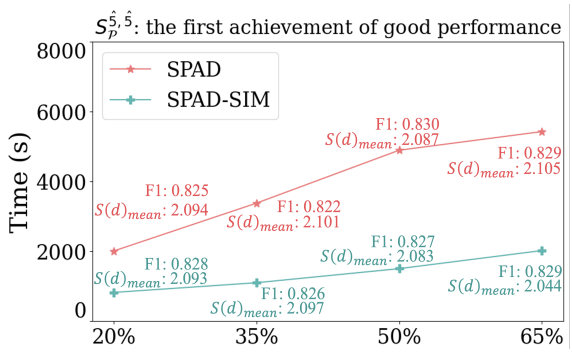


Fig. 20: Fairness and training time comparison on plantation mapping between SPAD and SPAD-SIM under different sizes of training samples at the first achievement of good performance.

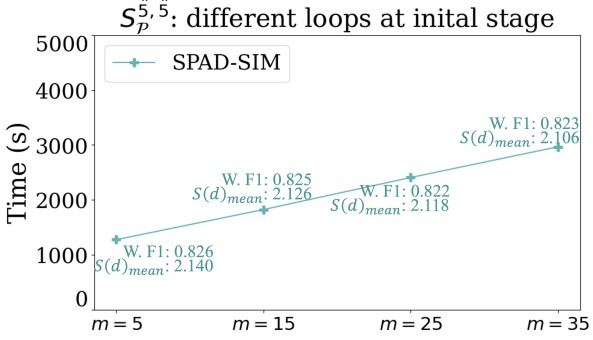


Fig. 21: Fairness and training time comparison on plantation mapping by SPAD-SIM with different loops at the initial stage.

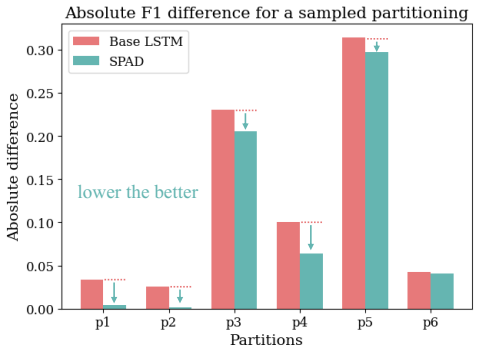


Fig. 22: The absolute difference between the obtained F1-scores over different partitions and the average F1-score for plantation mapping.

integrating the HDBSCAN clustering algorithm still improves the computational efficiency and reduces training time with the LSTM architecture. Finally, Fig. 22 shows an example result of the impact of SPAD on fairness for a sampled partitioning with six partitions, where SPAD demonstrates its ability to rectify disparities in F1-scores across different partitions.

7 Conclusion

Understanding and controlling location-related bias is critical for fair resource distribution in many societal domains, including agriculture, disaster management, etc. We proposed a new formulation of spatial-fairness-aware learning using the SPAD representation, which addresses statistical sensitivities in fairness evaluation caused by MAUP. We also proposed SPAD-based stochastic and bi-level training strategies to enforce spatial fairness in learning. Finally, we integrated a clustering algorithm to improve the computational efficiency of the proposed approach. Experiments on real-world agriculture monitoring data confirmed that the proposed approach is effective in improving spatial fairness while maintaining a similar level of overall performance. Also, it is shown that the integration of the clustering-based sampling algorithm can greatly reduce the time for model training without compromising the overall performance and fairness.

Our future work will expand the types of distributions and partitionings used in SPAD beyond the examples of uniform distribution and grid-based partitionings. We will also extend the method to cover a larger variety of spatial data types.

8 Acknowledgement

This work was supported by the NSF awards 2147195, 2239175, 2105133, and 2126474, 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.

REFERENCES

[1] CNBC, "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, 2020, accessed: 03/20/2022.

[2] A. Kamilaris *et al.*, "Deep learning in agriculture: A survey," *Computers and electronics in agriculture*, vol. 147, pp. 70–90, 2018.

[3] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.

[4] X. Jia, A. Khandelwal, D. J. Mulla, P. G. Pardey, and V. Kumar, "Bringing automated, remote-sensed, machine learning methods to monitoring crop landscapes at scale," *Agricultural Economics*, vol. 50, pp. 41–50, 2019.

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

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

[7] J. T. Bailey and C. G. Boryan, "Remote sensing applications in agriculture at the usda national agricultural statistics service," Research and Development Division, USDA, NASS, Fairfax, VA, Tech. Rep., 2010.

[8] C. Boryan, Z. Yang, R. Mueller, and M. Craig, "Monitoring us agriculture: the us department of agriculture, national agricultural statistics service, cropland data layer program," *Geocarto International*, vol. 26, no. 5, pp. 341–358, 2011.

[9] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-temporal data mining: A survey of problems and methods," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–41, 2018.

[10] S. L. Markstrom, *P2S–coupled Simulation with the Precipitation-Runoff Modeling System (PRMS) and the Stream Temperature Network (SNTemp) Models*. US Department of the Interior, US Geological Survey, 2012.

[11] USDA, "Economic research service farm resources regions," https://www.ers.usda.gov/webdocs/publications/42298/32489_aib-760_002.pdf, 2021, accessed: 03/20/2022.

[12] F. Waldner, Y. Chen, R. Lawes, and Z. Hochman, "Needle in a haystack: Mapping rare and infrequent crops using satellite imagery and data balancing methods," *Remote Sensing of Environment*, vol. 233, p. 111375, 2019.

[13] NPR, "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, 2019, accessed: 03/20/2022.

[14] I. M. Karaye and J. A. Horney, "The impact of social vulnerability on covid-19 in the us: an analysis of spatially varying relationships," *American journal of preventive medicine*, vol. 59, no. 3, pp. 317–325, 2020.

[15] J. Thebault-Spieker, B. Hecht, and L. Terveen, "Geographic biases are'born, not made' exploring contributors' spatiotemporal behavior in openstreetmap," in *Proceedings of the 2018 ACM Conference on Supporting Groupwork*, 2018, pp. 71–82.

[16] J. Thebault-Spieker, L. Terveen, and B. Hecht, "Toward a geographic understanding of the sharing economy: Systemic biases in uberx and taskrabbit," *ACM Transactions on Computer-Human Interaction*, vol. 24, no. 3, pp. 1–40, 2017.

[17] Y. Xie, E. He, X. Jia, W. Chen, S. Skakun, H. Bao, Z. Jiang, R. Ghosh, and P. Ravirathinam, "Fairness by "where": A statistically-robust and model-agnostic bi-level learning framework," in *Thirty-Sixth AAAI conference on artificial intelligence*. AAAI, 2022.

[18] E. He, Y. Xie, X. Jia, W. Chen, H. Bao, X. Zhou, Z. Jiang, R. Ghosh, and P. Ravirathinam, "Sailing in the location-based fairness-bias sphere," in *Proceedings of the 30th International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL)*, 2022, pp. 1–10.

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

[20] A. Yan and B. Howe, "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*, 2019, pp. 552–555.

[21] T. Kamishima, S. Akaho, and J. Sakuma, "Fairness-aware learning through regularization approach," in *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 2011, pp. 643–650.

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

[23] J. Alasadi, A. Al Hilli, and V. K. Singh, "Toward fairness in face matching algorithms," in *Proceedings of the 1st International Workshop on Fairness, Accountability, and Transparency in MultiMedia*, 2019, pp. 19–25.

[24] C. Sweeney and M. Najafian, "Reducing sentiment polarity for demographic attributes in word embeddings using adversarial learning," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 359–368.

[25] H. Zhang and I. Davidson, "Towards fair deep anomaly detection," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 138–148.

[26] E. S. Jo and T. Gebru, "Lessons from archives: Strategies for collecting sociocultural data in machine learning," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 306–316.

[27] K. Yang, K. Qinami, L. Fei-Fei, J. Deng, and O. Russakovsky, "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*, 2020, pp. 547–558.

[28] R. Steed and A. Caliskan, "Image representations learned with unsupervised pre-training contain human-like biases," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 701–713.

[29] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[30] M. Nasr and M. C. Tschantz, "Bidding strategies with gender nondiscrimination constraints for online ad auctions," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 337–347.

[31] Y. Xie, E. He, X. Jia, H. Bao, X. Zhou, R. Ghosh, and P. Ravirathinam, "A statistically-guided deep network transformation and moderation framework for data with spatial heterogeneity," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 767–776.

[32] Y. Xie, X. Jia, H. Bao, X. Zhou, J. Yu, R. Ghosh, and P. Ravirathinam, "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*, 2021, pp. 313–323.

[33] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2013, pp. 160–172.

[34] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 1, pp. 1–51, 2015.

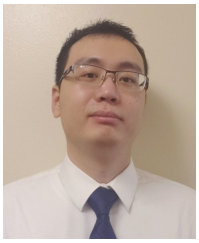
[35] A. C. A. Neto, J. Sander, R. J. Campello, and M. A. Nascimento, "Efficient computation of multiple density-based clustering hierarchies," in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 991–996.

[36] CDL, "Cropland data layer - usda nass," https://www.nass.usda.gov/Research_and_Science/Cropland/SARS1a.php, 2017, accessed: 03/20/2022.

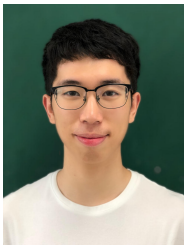
[37] P. Gunarso and others., "Rspo, kuala lumpur, malaysia," *Reports from the technical panels of the 2nd greenhouse gas working group of RSPO*, 2013.

[38] R. Petersen *et al.*, "Mapping tree plantations with multispectral imagery: preliminary results for seven tropical countries," *World Resources Institute, Washington, DC*, vol. 525, 2016.

Erhu He is a Ph.D. student in the Department of Computer Science at School of Computing and Information, University of Pittsburgh. His broad research interests are in machine learning, data mining, and fairness. More specifically, he is currently working on developing machine learning models that extract complex spatio-temporal data patterns. He received the Best Paper Award at IEEE ICDM 2021.

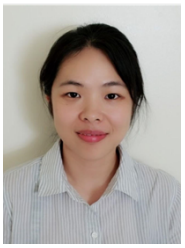


Yiqun Xie received his PhD degree in Computer Science from the University of Minnesota. He is currently an Assistant Professor in Geospatial Information Science at the University of Maryland, College Park. His research focuses on data mining and artificial intelligence methods for spatial data, such as satellite remote sensing data, UAV imagery, trajectories, etc. His recent research results have received the Best Paper Award from IEEE ICDM 2021, the Best Vision Paper Award from SIGSPATIAL 2019, and the Best Paper Award from SSTD 2019. His work was also highlighted by the Great Innovative Ideas program at the Computing Community Consortium at CRA.



Weiye Chen is working towards a PhD in Geospatial Information Science (GIS) at the University of Maryland, College Park. He has a BS degree in GIS at Zhejiang University and a MS degree in Geography at the University of Illinois, Urbana-Champaign. His research focuses on advancing techniques of spatial data mining and artificial intelligence.

Sergii Skakun received his PhD in System Analysis and Theory of Optimal Solutions (Computer Science) at the Space Research Institute of National Academy of Sciences of Ukraine and National Space Agency of Ukraine. He is an Assistant Professor with a joint appointment at the Department of Geographical Sciences and the College of Information Studies (iSchool) at the University of Maryland, College Park. His current research focus is to advance data science methods for heterogeneous remote sensing data fusion and analysis. He is the Associate Editor of Remote Sensing of Environment.



Han Bao is working towards a PhD degree in Informatics at the University of Iowa. Her research focuses on developing novel data mining and AI techniques for spatio-temporal big data, with applications in smart cities, transportation, public health, etc. She received the Best Paper Award from IEEE ICDM 2021.

Rahul Ghosh is a Computer Science Ph.D. student at the University of Minnesota, Twin Cities. His research highlights a general paradigm of entity-aware systems modeling in scientific applications by combining Physics-Guided Machine Learning for machine learning and physics-based modeling approaches. He is the recipient of Best Paper Awards from SDM'22, ICDM'21, DeepSpatial KDD'21

Praveen Ravirathinam is a Computer Science PhD student at the University of Minnesota, Twin Cities. He obtained his Bachelor's in Computer Science from Birla Institute of Technology and Sciences, Pilani. His research focuses on developing spatiotemporal deep learning algorithms, with a special focus on applications in remote sensing, such as crop monitoring, river width tracking, and water body classification.

Xiaowei Jia is an Assistant Professor in the Department of Computer Science at the University of Pittsburgh. His research interests include knowledge-guided data science and spatio-temporal data mining for societally important applications. A major highlight of his research is a general paradigm Physics-Guided Machine Learning for combining machine learning and physics-based modeling approaches. He is the recipient of the University of Minnesota Best Dissertation Award and Best Paper Awards from SDM 22, ICDM 21, SDM 21, ASONAM 16, and BIBE 14.