Point-to-Region Co-learning for Poverty Mapping at High Resolution Using Satellite Imagery

Zhili Li[1], **Yiqun Xie**[1*], **Xiaowei Jia**[2], **Kara Stuart**[3], **Caroline Delaire**[3], **Sergii Skakun**[1]

[1]University of Maryland
[2]University of Pittsburgh
[3]The Aquaya Institute
{lizhili, xie}@umd.edu, xiaowei@pitt.edu, {kara, caroline}@aquaya.org skakun@umd.edu

Abstract

Despite improvements in safe water and sanitation services in low-income countries, a substantial proportion of the population in Africa still does not have access to these essential services. Up-to-date fine-scale maps of low-income settlements are urgently needed by authorities to improve service provision. We aim to develop a cost-effective solution to generate fine-scale maps of these vulnerable populations using multi-source public information. The problem is challenging as ground-truth maps are available at only a limited number of cities, and the patterns are heterogeneous across cities. Recent attempts tackling the spatial heterogeneity issue focus on scenarios where true labels partially exist for each input region, which are unavailable for the present problem. We propose a dynamic point-to-region co-learning framework to learn heterogeneity patterns that cannot be reflected by point-level information and generalize deep learners to new areas with no labels. We also propose an attention-based correction layer to remove spurious signatures, and a region-gate to capture both region-invariant and variant patterns. Experiment results on real-world fine-scale data in three cities of Kenya show that the proposed approach can largely improve model performance on various base network architectures.

Introduction

While access to safe water and sanitation services in low-income countries has increased substantially over the past twenty years, governments struggle to ensure that poor and vulnerable households equally benefit from these services. Today, 70% of the population in Africa does not have access to safe drinking water, and 50% does not have access to an adequate toilet (Organization et al. 2019). The imperative set by the United Nations Sustainable Development Goals (UN 2022) to "leave no one behind" requires dedicated strategies to improve water, sanitation and hygiene (WASH) access amongst the most vulnerable and hard-to-reach populations. Local governments however lack critical information that they need to plan sanitation interventions targeted at the most vulnerable. For example, there is currently no publicly available, high-resolution poverty map for the majority of African cities to support poverty-targeted programs. Moreover, existing maps of urban slums rely on field data collection and can be quickly outdated as slums grow, densify, gentrify, or appear in new locations. As a result, when crises hit, the response is delayed because decision-makers lack information on where services may be needed most. The ability to accurately map impoverished settlements at a low-cost will provide substantial help to WASH implementers as they design, customize, and budget programs. For example, such maps can better inform subsidy distribution, which recent findings have shown to often benefit high-income households, failing to reach the very poor (Andres et al. 2019).

We aim to develop a learning-based solution to generate fine-scale maps of low-income areas (LIAs) in fast-growing urban areas. The inputs leverage publicly available information for cost-effectiveness purposes, including Sentinel-2 multi-spectral satellite imagery at 10m resolution, and auto-tracked internet and cellular connection data. The use of free sources is critical for sustainable real-world deployment.

The problem poses two major challenges. First, ground-truth maps of LIAs at fine-scale are expensive and time-consuming to collect due to the need of household level field surveys. As a result, such maps only exist in a limited number of cities and no label is available for most cities. Second, the functional relationship between features $\mathbf{X}$ and label $\mathbf{y}$ is often nonstationary over space due to intrinsic spatial heterogeneity (Xie et al. 2021; Goodchild and Li 2021), making the classification criteria for LIAs vary across cities.

Earlier AI-based poverty maps (Xie et al. 2016; Jean et al. 2016) are often generated using low-resolution data (e.g., NOAA nighttime light intensity), which are ideal for continental analysis but not informative for detailed local action and implementation (e.g., WASH programmers). Recent developments suggested correlations between the count of objects (e.g., buildings) and poverty level and deep object detectors were used to help estimate poverty scores (AyushBurak Uzkent et al. 2021). However, the analysis is designed for rural areas and are not well-suitable for urban areas with dense and highly dynamic landscapes, which is the focus of this work. In addition, the detection task requires very high-resolution imagery (e.g., 0.3m), which is often not freely available and not sustainable for implementers. The study also points out the critical issue on spatial heterogeneity, which substantially limits the accuracy of the model in new cities (AyushBurak Uzkent et al. 2021). Domain adaptation mitigates domain shifts by homogenizing distributions be-

*Corresponding author.

tween source and target domains (Pan et al. 2010; Ben-David et al. 2007; Wang and Deng 2018; Jia et al. 2019). However, in our problem different regions may actually need to learn different criteria due to distinctions in patterns (compared in experiments). While meta-learning such as MAML (Finn et al. 2017; Rußwurm et al. 2020) can quickly adapt to new criteria in new tasks, it requires new ground-truth data which are often not available for LIA mapping in the already resource-constrained geographic areas. In addition, meta-learning often needs a large number of tasks, whereas our training data only exist in a very limited number of cities. There have also been deep learning frameworks that tackle the spatial heterogeneity problem, which partition data (e.g., SVANN (Gupta et al. 2021), spatial ensemble (Yuan et al. 2018)) or learn to partition data (e.g., STAR (Xie et al. 2021)) into homogeneous groups to improve prediction performance, whereas our goal is to translate criteria learned from limited cities to new ones without training samples. Geographically weighted regression (Brunsdon et al. 1999), a traditional nonparametric approach to handle spatial heterogeneity, requires dense training data over space, and its linear modeling is not suitable for detecting complex patterns in satellite data. Finally, we conducted extensive field evaluations of existing LIA maps through household interviews, and found the maps from local governments are largely out of date and no longer usable for resource allocation (e.g., voucher programs).

We propose a Point-to-Region Co-Learning approach to address the challenges with the following contributions:

- We propose a point-to-region scheme to go beyond patterns at the level of individual data points and capture region-dependent classification criteria that are mainly reflected by the relative relationships between points within a region; in this paper, a region is represented by a city.
- We present several strategies to reduce the overhead needed for co-training at the point- and region-level.
- We introduce an attention-based correction layer and region-gate to remove spurious region-level patterns and improve generalizability.
- We conducted detailed and extensive field surveys in three target cities in Kenya to create a high-quality ground-truth dataset for experiments and facilitate future deployment.

Experiment results on different base deep network architectures show that the proposed point-to-region co-learning framework can greatly improve prediction performance on test cities with no training samples.

Problem Formulation

Inputs:

– Features $\mathbf{X}$ from multiple public sources, including satellite imagery (e.g., Sentinel-2 multi-spectral imagery), internet speed, and cellular connection information.

– Ground-truth labels $\mathbf{y}$ of low-income areas (LIA) in cities where field surveys were conducted.

Output: Predicted $\hat{\mathbf{y}}$ for new cities.
Objective: Classification performance (e.g., F1-scores).

The ground truth labels in this work are collected through field surveys in selected cities with WASH access issues. Given that the label data only exist in limited cities, the goal is to apply the trained model onto new cities where LIA labels are not available but are essential for resource distribution (e.g., WASH programs). We use features that are publicly available for longer-term sustainability in deployment.

Point-to-Region Co-Learning

We present a new framework to learn region-dependent classification criteria to reduce the generalization gap to new regions (e.g., new cities) where no true labels are available. Specifically, each data point will be evaluated in the context of a collection of data points, where the relative relationships provide helpful information to guide region-specific classification. In the following, we introduce the point-to-region representation, a co-learning strategy with reduced computation, a spurious-signature correction layer, and a region-gate. The overall framework is shown in Fig. 1.

Point-to-Region Representation

Definition 1 *A **point** is a data point $\boldsymbol{X}_i \in \boldsymbol{X}$, which contains information of a local region in space-time. The format of $\boldsymbol{X}_i$ depends on the task formulation of the LIA classification problem. For example, $\boldsymbol{X}_i \in \mathbb{R}^{m\times m\times d}$ if semantic segmentation is used, where m is the size of a local patch (in the entire city-wide satellite imagery); and $\boldsymbol{X}_i \in \mathbb{R}^{t\times d}$ if LSTM-based model is used, where t is the number of images per year (e.g., one per month) at each location.*

Definition 2 *A **region** G is the collection of data points that together form a representative distribution of a city $\mathcal{C}_k$. G can either be all data points inside $\mathcal{C}_k$, i.e., $G = \{\boldsymbol{X}_i \,|\, \forall loc_i \in \mathcal{C}_k\}$, where loc_i is the location of $\boldsymbol{X}_i$, or a representative sampling of points.*

We use this point-to-region view because in LIA mapping the criteria or signatures for classifying LIAs and non-LIAs may vary from region to region. The variation makes it difficult to infer the class based solely on information from local data points without considering their relative relationship to the other points in a region. For example, certain cities have relatively sparser tree coverage due to various physical (e.g., dryer climate) or social conditions. In such cities, greenness from large trees in residential areas is often an informative indicator of higher household income, i.e., non-LIAs. However, in naturally "greener" cities – which have a denser and more widespread tree coverage across locations – the tree-based criteria need to be stricter. Fig. 2 shows an example where residential areas with similar tree signatures fall into different classes in two different cities. Similarly, the overall quality of buildings may also vary by city. Houses that can be characterized as part of LIAs in a city with better overall infrastructure may be considered as a typical non-LIA in another city. The same challenge broadly applies to other features of data points $\mathbf{X}_i$, including internet speed, building pattern, etc. This requires different criteria for data points in different regions, calling for a learning scheme that can explicitly capture the relative relationship between a data

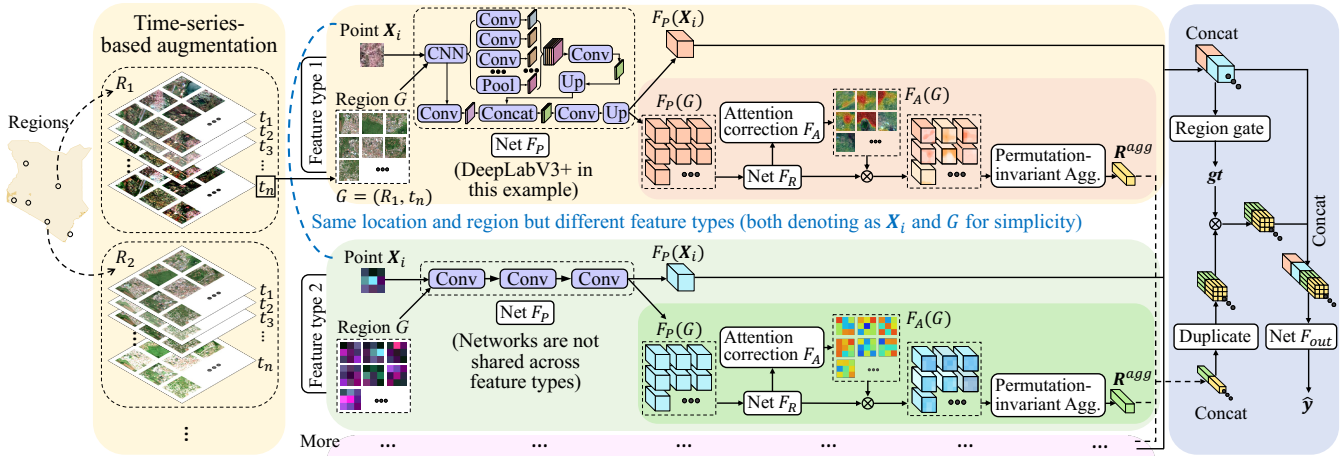
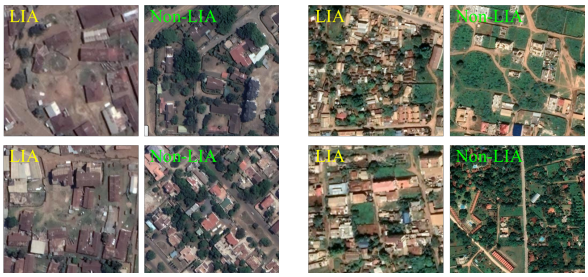


Figure 1: Illustration of the overall framework of point-to-region co-learning.



(a) Kisumu (b) Malindi

Figure 2: Examples of LIAs and non-LIAs in Kenya, showing the importance of region-based criteria adjustments.

point $\mathbf{X}_i$ and the collection of data points from its region $G = \{\mathbf{X}_i \,|\, \forall loc_i \in \mathcal{C}_k\}$, where $\mathcal{C}_k$ is a city in this paper.

We use Definitions 1 and 2 to satisfy two desired characteristics for the point-region representation.

Definition 3 ***Feature consistency.*** *The features at region-level should be extracted using full-information from each of its data points $\boldsymbol{X}_i$, which is necessary for the region-features to be able to form a basis for criteria adjustments at point-level. In other words, it must be possible for a learned region-level features $F_R(G)$ to reproduce learned point-level features $F_P(\boldsymbol{X}_i)$, where:*

$$F_R(G) = \{F_R(\boldsymbol{X}_i) \,|\, \forall \boldsymbol{X}_i \in G\} \tag{1}$$

Using satellite imagery as an example, Def. 3 implies that feature learning at the region-level should execute on inputs that have the same resolution as the individual points. Otherwise, $F_R(G)$ generated by a lower-resolution inputs may not contain useful information for $F_P(\mathbf{X}_i)$ at a higher-resolution. Our region definition (Def. 2) satisfies this requirement by representing a region as a collection of original data points. Additionally, we need another property:

Definition 4 ***Immunity to perturbation.*** *The features learned at the region level should not be subject to perturbations of data points in the region collection G.*

Perturbation immunity is necessary to avoid order-sensitivity in training and applications. Otherwise, two highly similar collections (e.g., one is a copy of the other but with one point removed) may return very distinct results. In addition, the order of data points cannot be standardized in a meaningful way between different regions, which may have different sizes and shapes. In our design, we keep feature learning independent for each data point $\mathbf{X}_i \in G$, and then aggregate point-features to region-features using global mean, combined with region-correction (discussed later).

Co-Training Sequence

To allow region-level features $F_R(G)$ to serve as a basis for adjusting classification criteria for points in different regions, they need to be trained together with point-level features $F_P(\mathbf{X}_i)$. A main challenge in co-training $F_R(G)$ and $F_P(\mathbf{X}_i)$ is that region collection G often contains a large number of data points (e.g., thousands), making it computationally expensive, especially for data points of larger size (e.g., larger local patches of satellite imagery) and smaller batch sizes. Denote B as the batch size used in point-level training, and $|G|$ as the number of points in a region collection. Suppose all points in a batch are from the same region and the deep network size for learning point- and region-features are the same, the computational overhead ratio for adding the region-level training is then $|G|/B$. We use two strategies to reduce the extra overhead introduced by the region-level training as follows.

Region feature reduction. The goal of region feature reduction is to reduce the extra memory consumption and related computation. Specifically, the additional $|G|$ inputs from the region collection not only adds computational overhead, but also requires larger memory to hold intermediate tensor outputs. Since one desired property of region-level feature learning is information consistency (Def. 3), during co-training, we apply the same network and parameters used to construct point-features $F_P(\mathbf{X}_i)$ to generate initial inputs for region-feature learning. As illustrated in Fig. 3 (a) and (b), the region-feature network F_R is thinned by reducing

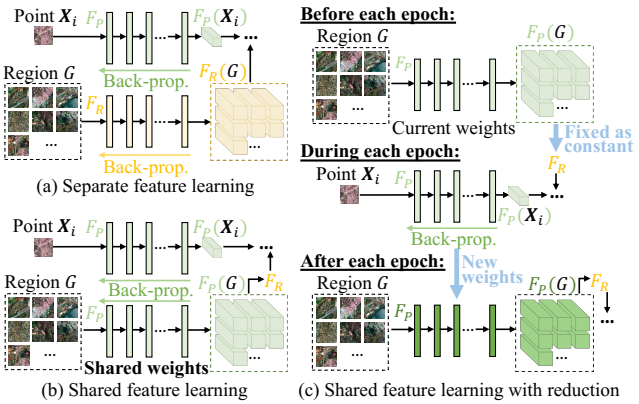


Figure 3: Region feature reduction.

the learning process from the raw $\mathbf{X}_i$ to $F_P(\mathbf{X}_i)$, i.e., from $F_R(G)$ to $F_R(F_P(G))$, where $F_P(G) = \{F_P(\mathbf{X}_i) \,|\, \forall \mathbf{X}_i \in G\}$. The region-features have the capacity to reproduce point-features, which satisfies Def. 3.

Asynchronous update. To further reduce the overhead, we use asynchronous updates in the training sequences for point- and region-features. As shown in Fig. 3 (c), the region-feature learning component is updated at two different frequencies, where (1) its input $F_P(G)$ is made stationary within each epoch and (2) F_R and following layers are updated over iterations as normal. Note that the point-feature learning network F_P is still updated over iterations.

The asynchronous update algorithm is illustrated in Alg. 1. The process includes an initialization phase, where $\mathbf{\Theta}_P$ is trained without the point-to-region representation; several output layers F_{out}^{temp} are temporarily attached for classification during this phase. This helps create more stable region-features before the co-training starts.

Parallel network branches for point-to-region co-training. As different types of input features (e.g., spectral signals from satellite imagery, internet speeds) may require different adjustments of classification criteria over different regions, in our network we create a separate feature-learning branch for each type of feature (Fig. 1).

Attention-based Regional Aggregation Correction

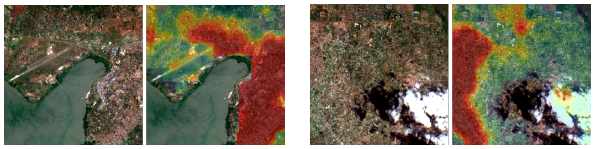
While features learned from the regional level can provide meaningful information to adjust classification at the point level, one extra challenge that may hinder this process is the common presence of noises in real data. As aggregation is used in F_R to combine point-level features, noisy points can easily pollute the results, especially when they occupy a large proportion. Explicit filtering of such noises is critical for region-feature learning in the LIA mapping problem. For example, large volumes of noises (or unrelated information) are often expected in satellite observations. Common phenomenon or objects such as clouds, large water bodies (e.g., sea surface, major lakes), etc., do not carry meaningful information to train the region-features. Fig. 4 shows examples of large lakes and cloud coverage in Kisumu, Kenya from a

Algorithm 1: Asynchronous update

Require: • Features $\mathbf{X}$ and labels $\mathbf{y}$; • Regions $\{G_k \,|\, k = 1, ..., K\}$; • Learning rate α.

{# Initialize $\mathbf{\Theta}_P$}

for i = 1 to $init\_epochs$ **do**
  for batch $(\mathbf{X}_B, \mathbf{y}_B)$ in $(\mathbf{X}, \mathbf{y})$ **do**
    $\hat{\mathbf{y}}_B = F_{out}^{temp}(F_P(\mathbf{X}_B))$
    $\mathbf{\Theta}_P = \mathbf{\Theta}_P - \alpha \cdot \nabla_{\mathbf{\Theta}_P} \mathcal{L}(\mathbf{y}_B, \hat{\mathbf{y}}_B)$
    $\mathbf{\Theta}_{out}^{temp} = \mathbf{\Theta}_{out}^{temp} - \alpha \cdot \nabla_{\mathbf{\Theta}_{out}^{temp}} \mathcal{L}(\mathbf{y}_B, \hat{\mathbf{y}}_B)$
  end for
end for
for i = $init\_epochs$ to max_epochs **do**
  for k = 1 to K **do**
    Region update with current $\mathbf{\Theta}_P$: $F_k^{constant} = F_P(G_k)$
    $data_k = \{(\mathbf{X}_j, \mathbf{y}_j) \,|\, \forall \mathbf{X}_j \in G_k\}$
    {# For simplicity, F_{com} and its weights $\mathbf{\Theta}_{com}$ in ln. 13 & 16 cover all layers in Fig. 1's blue part on the right.}
    for batch $(\mathbf{X}_B, \mathbf{y}_B)$ in $data_k$ **do**
      $\hat{\mathbf{y}}_B = F_{com}(F_P(\mathbf{X}_B), F_R(F_k^{constant}))$
      $\mathbf{\Theta}_P = \mathbf{\Theta}_P - \alpha \cdot \nabla_{\mathbf{\Theta}_P} \mathcal{L}(\mathbf{y}_B, \hat{\mathbf{y}}_B)$
      $\mathbf{\Theta}_R = \mathbf{\Theta}_R - \alpha \cdot \nabla_{\mathbf{\Theta}_R} \mathcal{L}(\mathbf{y}_B, \hat{\mathbf{y}}_B)$
      $\mathbf{\Theta}_{com} = \mathbf{\Theta}_{com} - \alpha \cdot \nabla_{\mathbf{\Theta}_{com}} \mathcal{L}(\mathbf{y}_B, \hat{\mathbf{y}}_B)$
    end for
  end for
end for
return $\mathbf{\Theta}_P, \mathbf{\Theta}_R, \mathbf{\Theta}_{com}$



(a) Large open water (b) Clouds

Figure 4: Attention-based correction for aggregation in F_R. Lower attention values have higher transparency.

Sentinel-2 satellite imagery (only visible bands are used for the visualization). Including phenomenons such as clouds during aggregation in F_R will contaminate the features and make region-feature based predictions unstable.

To allow the network to have the flexibility of removing such undesired data points, we add a region-level attention component F_A, which provides an importance mask $\mathbf{A}_i = F_A(\mathbf{X}_i)$ to each data point $\mathbf{X}_i$ in a region G, where $\mathbf{A}_i \in \mathbb{R}^{m \times m}$ and the values are in range $[0, 1]$. The region-level aggregated vector $\mathbf{R}^{agg} \in \mathbb{R}^K$ (K is the total number of output features from F_R) is then:

$$\mathbf{R}_k^{agg} = \left(\sum_{i=1}^{|G|} \mathbf{e}^T (\mathbf{A}_i \otimes \mathbf{R}_{i,k}) \mathbf{e} \right) / \left(\sum_{i=1}^{|G|} \mathbf{e}^T \mathbf{A}_i \mathbf{e} \right) \quad (2)$$

where $\mathbf{R}_k^{agg} \in \mathbb{R}$ is the k^{th} element of the $\mathbf{R}^{agg}$; $\mathbf{R}_{i,k} \in \mathbb{R}^{m \times m}$ is the k^{th} layer of the final region feature learned for point $\mathbf{X}_i$ (i.e., $\mathbf{R}_{i,k} = F_R(F_P(\mathbf{X}_i))$, which is part of the output of "Net F_R" in Fig. 1); $\mathbf{e}$ is a vector of ones; and $\otimes$ is the Hadamard product. Note that the dimension of $\mathbf{R}^{agg}$ was reduced by the aggregation over all pixels within each point and all points. As different types of features are processed

through parallel network branches, $\mathbf{X}_i$ here represents features of one type (e.g., spectral band values from imagery) for a data point. Accordingly, the attention-based filters are kept separate for different features as well.

Fig. 4 shows examples of attention-based correction values predicted for open water and clouds. Here each point $\mathbf{X}_i$ is a local image patch, and we generated this map by predicting attention scores of a moving-window that has the same size of the patch. In the actual co-training or prediction, the data points we use to form each region do not have mutual overlaps. The attention weights are automatically learned through back-propagation from the final classification loss $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$, and do not require additional labels. However, if labels are known and exist, one can always introduce an additional loss function to train or initialize the filter.

Region-Gate for Variant-Invariant Patterns

The final component – region-gate – aims to reconcile the inconsistency between patterns that vary or do not vary across regions. While the framework is designed to capture potential differences in classification criteria between regions, there are also patterns that are mostly invariant. For example, houses with large pools or large green yards are often non-LIAs regardless of the region, and pure urban forests or lake points are non-LIAs (i.e., non-residential areas).

As both variant and invariant patterns exist in the data, forcing the application of region-features, or region-based adjustments, may introduce additional difficulty in classifying the region-invariant "easy" samples. Thus, to mitigate the inconsistency, we add a region-gate to determine whether a location needs to be coupled with the region features before classification. The region-gate output $\mathbf{gt} \in \mathbb{R}^{m \times m}$ represents the importance of region features over locations:

$$\mathbf{gt} = F_{gate}(F_P(\mathbf{X}_i)) \tag{3}$$

where the input contains learned point-features from F_P instead of the raw $\mathbf{X}_i$ and each value in $\mathbf{gt}$ is in range (0,1). The region-gate may be used with the output layers F_{out} in two different forms (Fig. 1 only shows the first form):

$$\hat{\mathbf{y}}_a = F_{out}\big(F_P(\mathbf{X}_i) +\!\!+ \mathbf{gt} \otimes \mathbf{R}^{mat}\big) \tag{4}$$

$$\hat{\mathbf{y}}_b = \mathbf{gt} \otimes F_{out}\big(F_P(\mathbf{X}_i) +\!\!+ \mathbf{R}^{mat}\big) + (1 - \mathbf{gt}) \otimes F'_{out}\big(F_P(\mathbf{X}_i)\big) \tag{5}$$

where $\mathbf{R}^{mat} \in \mathbb{R}^{m \times m \times K}$ is matrix containing $m \times m$ copies of $\mathbf{R}^{agg}$ (Eq. (2)); $+\!\!+$ is concatenation; and $\otimes$ is Hadamard product. The first form $\hat{\mathbf{y}}_a$ in Eq. (4) concatenates the point and region features, and the gate controls the contribution from region features. $\hat{\mathbf{y}}_b$ in Eq. (5) includes another set of output layers F'_{out} to generate the outputs using only the point-level features. It then combines the results using an weighted average based on $\mathbf{gt}$.

Backbone and Augmentation

Backbones. We evaluated the framework on three backbones: U-Net, PSPNet and DeepLabV3+. More details on the implementation can be found in the Appendix. We use the Dice loss for all three backbones to mitigate potential class imbalance issues in the LIA mapping problem: $\mathcal{L}_{dice} = 1 - 2 \cdot \sum_i \mathbf{y}_i \hat{\mathbf{y}}_i / \sum_i (\mathbf{y}_i + \hat{\mathbf{y}}_i)$, where the second term is the Dice-coefficient.

Time-series-based region augmentation. As the ground-truth labels are only available in very limited number of cities, we add a time-series-based region augmentation strategy to help improve the robustness of the training. While the labels, i.e., the LIA maps are relatively stable within a year, some of our input features are often collected during multiple timestamps of a year, e.g., the multi-spectral remote sensing imagery. As there are often variations in these observation values, observations at each timestamp are used to form a separate training sample to increase the data size. More importantly, treating data from each region and timestamp (e.g., each month) as one separate region (i.e., uniquely identified by a combination of region and time) allows the model to see a more diverse set of regions during training and makes it more stable. Thus, we use this time-series-based augmentation in the training process (the augmentation is also used with baseline methods where applicable).

Experiments

Urban LIA Data

We carried out detailed field studies to collect the ground-truth data of LIAs in three cities of Kenya: Kisumu, Malindi and Nakuru. The cities are selected according to a recent study that modeled the costs to achieve the United Nations' 2030 Sustainable Development Goals on universal access to safely managed sanitation (Delaire et al. 2020). We use Sentinel-2 multi-spectral imagery (13 bands), as well as fixed broadband and mobile (cellular) network speed statistics (e.g., download/upload speeds, latency) as input features. More details are available in the Data Appendix.

Candidate Methods

We implemented all the candidate methods using the three backbones, i.e., U-Net, PSPNet and DeepLabV3+:

- **Base:** The backbone network of choice, which is trained on the data $\mathbf{X}^{train}$, $\mathbf{y}^{train}$ from the two training cities and evaluated on the unseen test city.
- **DA-d:** Backbone with domain adaption, where the training data include the features $\mathbf{X}^{test}$ from the test cities (Yan et al. 2017; Li et al. 2020). And a loss function is added to penalize large divergence and learn domain-invariant features: $\mathcal{L} = \mathcal{L}_{pred} + \lambda \mathcal{L}_{domain}$, where $\mathcal{L}_{domain} = \sum_i^{N_B} [\mu(F(\mathbf{X}_i^{train})) - \mu(F(\mathbf{X}_i^{test}))]$, F is the features learned by the backbone, N_B is the batch size, μ is the mean function, and λ is the weight (0.5).
- **DA-GAN:** Backbone with domain adaptation by adversarial learning (Tzeng et al. 2017; Huang et al. 2018), where the backbone is extended with a generative adversarial network (GAN).
- **CT-f:** A common context-based formulation, where additional auxiliary features are used to describe general contextual information or external factors (e.g., static environmental variables in streamflow prediction) about each

	Test Region	Base	DA-d	DA-GAN	CT-f	CT-KNWL	SE	MOD	PTR-g_1	PTR-g_2
U-Net	Kisumu	0.360	0.153	0.360	0.327	0.304	0.178	0.453	**0.504****	**0.506***
	Malindi	0.233	0.299	0.299	**0.344****	0.338	0.276	0.277	0.269	**0.388***
	Nakuru	0.533	0.227	**0.598***	0.536	**0.586****	0.547	0.573	0.583	0.549
	Mean	0.375	0.226	0.419	0.402	0.409	0.334	0.434	**0.452****	**0.481***
	Mean$_w$	0.411	0.192	0.440	0.403	0.408	0.318	0.480	**0.512***	**0.512****
PSPNet	Kisumu	0.211	0.154	0.295	0.332	0.154	0.154	0.349	**0.418***	**0.385****
	Malindi	0.279	0.310	0.299	0.338	0.346	0.121	0.342	**0.409***	**0.351****
	Nakuru	**0.562****	0.224	**0.576***	0.555	0.542	0.090	0.561	0.531	0.543
	Mean	0.351	0.229	0.390	0.408	0.347	0.121	0.417	**0.453***	**0.426****
	Mean$_w$	0.342	0.192	0.396	0.412	0.309	0.128	0.424	**0.458***	**0.438****
DeepLabV3+	Kisumu	0.343	0.154	0.154	0.329	0.376	0.372	**0.451****	**0.461***	0.391
	Malindi	0.208	0.313	0.341	**0.377***	0.284	0.167	0.281	**0.364****	0.360
	Nakuru	0.513	0.236	0.450	0.537	**0.597***	0.403	0.385	**0.563****	0.563
	Mean	0.355	0.234	0.315	0.414	0.419	0.314	0.372	**0.463***	**0.438****
	Mean$_w$	0.392	0.197	0.276	0.407	0.447	0.366	0.413	**0.489***	**0.450****

Note: * for best results; ** for runner-ups.

Table 1: F1 scores for LIA mapping (when a city is a test region, it is left out from training).

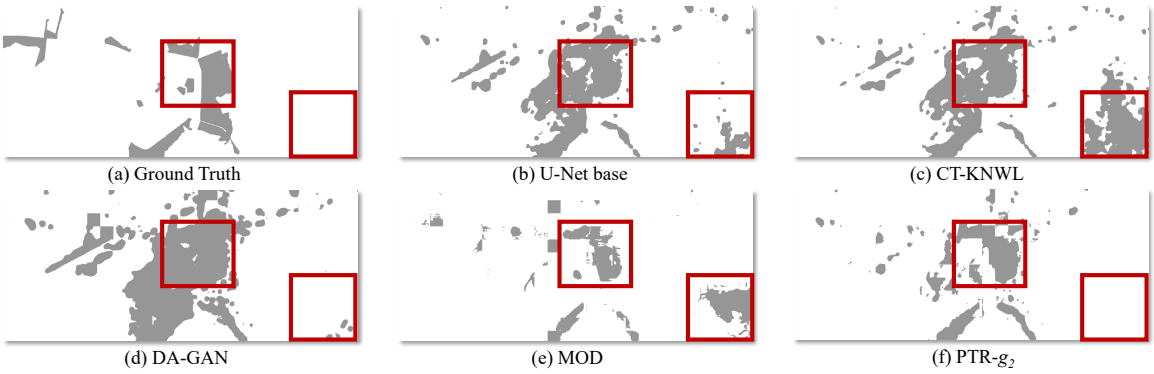
data point (Zhang, Zheng, and Qi 2017; Ye et al. 2019; Jia et al. 2021). Here we include two types of context modeling. This CT-f uses mean feature values over all data points in each entire city as the auxiliary context vector.

- **CT-KNWL:** This version generates context features based on domain knowledge. We consider three remote sensing indices commonly used in practice (Chen et al. 2006): Normalized Difference Vegetation Index (NDVI), Normalized Difference Water Index (NDWI) and Normalized Difference Built-Up Index (NDBI). Similarly, the mean index values of each city are used as the context.
- **SE:** The spatial ensemble approach for handling spatial heterogeneity, where deep learning models $\{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_K\}$ are first trained for each data subset (i.e., each city in this problem), and then a linear regressor is used to derive optimal weights $\boldsymbol{\beta}$ to ensemble individual model predictions (Yuan et al. 2018), which are applied to test data: $\hat{\mathbf{y}}^{test} = \sum_i^K \boldsymbol{\beta}_i \cdot \mathcal{M}_i(\mathbf{X}^{test})$.
- **MOD:** The spatial moderator is a spatial generalization component in STAR (Xie et al. 2021). Compared to SE, MOD uses adaptive weights $\boldsymbol{\beta}$ for individual models instead of using fixed weights. Specifically, $\boldsymbol{\beta}$ is dynamically derived using a deep network – the moderator – for each input data point $\mathbf{X}_i$. The feature-adaptive ensemble is learned using data points from the training cities and applied to those in the test cities.
- **PTR-g_1** and **PTR-g_2:** The proposed point-to-region framework using the two forms of gate (Eqs. (4) and (5)).

Results

Table 1 shows the performances (F1-scores) from the candidate methods using each of the three different backbones. Since the target application scenario of the proposed approach is to apply models trained from cities $\{C\}$ to other cities $\{C'\}$ where no training label is available, we used the same setting during the experiments. Specifically, in each experiment, we use data from two cities for training and the city left for testing. For example, when we use the Kisumu as the test city, data from Nakuru and Malindi are used for training. Following this, each row in Table 1 shows the results for a corresponding test city. In addition, the tables also include two types of mean values over the three cities: (1) "Mean" represents a direct average over the F1-scores from the three test cities for each candidate method; and (2) "Mean$_w$" represents the weighted average of the F1-scores, where the weight for each city is assigned based on its number of data points. Fig. 5 also shows example visualizations of LIA mapping in Kisumu with U-Net. The hyper-parameters are determined by maximizing the performance on the validation dataset, which is a 20% subset separated from the training data. The models are trained using the Adam optimizer with an initial learning rate of 10^{-4}.

Comparison to baseline methods. From Table 1 we can see that the proposed point-to-region approaches outperformed the other candidate methods in most of the scenarios. Specifically, PTR-g_1 has the best or runner-up performances (i.e., denoted by * and **) in 12 out of the 15 scenarios, and PTR-g_2 in 10 out of the 15 scenarios, showing the improvements achieved by the additional region-level modeling. Among the other candidate methods, MOD has better results potentially due to its ability to dynamically adapt to new regions. For this problem, domain adaptation methods (especially DA-d) do not show stable improvements. The reason can be that different cities may need different criteria for classification, so minimizing feature differences between source and target domains may constrain the availability of useful features to classify LIAs in the test city. Comparing the results from different rows (test cities) in Table 1, the candidate methods tend to have smaller F1-scores in Malindi, Kenya. The reason may be that Malindi has a smaller size and more cloud coverage. Some of its feature values (e.g., internet speeds) also tend to be slightly outside the range of the other cities, making it more difficult to approximate its pattern using data from the other two cities. In contrast, when Malindi is used as one of the training cities, the models tend to have a better performance. The scores are



(a) Ground Truth (b) U-Net base (c) CT-KNWL

(d) DA-GAN (e) MOD (f) PTR-g_2

Figure 5: Visualization of LIA mappings in Kisumu, Kenya.

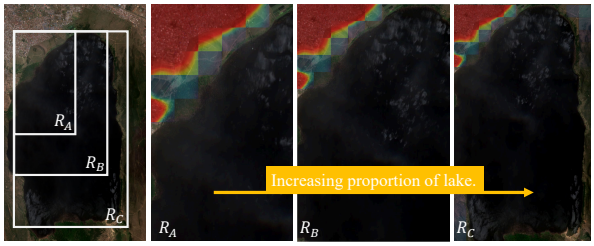


Figure 6: Attention overlaid on top of Sentinel-2 imagery.

not super high (e.g., > 0.9) for all models due to the large variability of LIAs and limited data, but the P2R methods demonstrate improvements.

Results with different backbones. Comparing across the backbones in Table 1, we can see the performances of the candidates methods do not present a strong pattern for this problem. For example, according to the results of the base models themselves (the first column in Table 1), PSPNet has the best performance on Malindi and Nakuru, whereas U-Net has the best result in Kisumu. For the proposed PTR methods, U-Net shows the highest scores as the backbone on average. Overall, there is no dominant pattern, which suggests it is worth trying out these base backbones in applications.

Sensitivity Analysis: Correction and Features

Since the correction aims to avoid the inclusion of non-useful information during the aggregation phase of region-feature learning, we used the city of Nakuru to design this experiment as it is adjacent to the large open water from the Lake Nakuru. Specifically, Fig. 6 shows three regions represented by three bounding boxes, where the proportion of open-water gradually increases. Since in real applications it may not be straightforward to manually select a "perfect" boundary to use as the region in the framework, we expect the correction layer to help us filter out the effect of non-useful information when included. Based on the attention results from Fig. 6 (the backbone in this example is DeepLabV3+), we can see that pixels over the water body receive near-zero attention values while the important urban areas have high weights. This allows the model to be robust against the size of the input region. Table 2 shows the corresponding F1-scores for results with and without the correction. The attention sum represents the total attention values from the region and the area is calculated by the number of pixels. To best control the effects of the proportion of the water body in this experiment, the top-left corner includes only part of the urban areas of Nakuru. Thus, the F1 scores here are for the LIAs inside that specific region, which are different from those in Table 1. As we can see, for the version without the correction layer, the performance continues to drop as the region includes more water body. In contrast, the correction makes the performance much more stable.

Region	Corr.	No corr.	Diff.	Att. sum	Area
R_A	0.864	0.659	0.205	43140	245760
R_B	0.864	0.614	0.249	43174	516096
R_C	0.849	0.465	0.385	50836	856064

Table 2: Attention-based correction (Nakuru; DeepLabV3+)

On the feature side, we found both visible (plus near-infrared) and other bands are important for LIA mapping. Non-visible bands show more impact on the F1-scores. Similarly, fixed broadband speed shows more importance than mobile speed. More details are included in the Appendix.

Conclusions and Future Work

We proposed a point-to-region co-learning framework for mapping LIAs using features from multiple sources. Within the framework, we proposed a point-to-region representation with a co-training strategy, a region correction layer and a region-gate to allow adaptive adjustments of classification criteria when the model is applied to a test city without any labels. We also created a new LIA mapping dataset through field work in three cities of Kenya. By comparing the proposed approach with various baselines on different backbone architectures, the new method demonstrated promising performance improvements. In future work, we will identify additional features to further improve the results and incorporate consideration of spatial fairness (Xie et al. 2022) into modeling and training. We will also extend the framework to other application scenarios beyond poverty mapping.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2105133, 2126474 and 2147195; NASA under Grant No. 80NSSC22K1164 and 80NSSC21K0314; USGS under Grant No. G21AC10207; US-DOT under Grant No. 69A3551747131 (through SAFER-SIM); Google's AI for Social Good Impact Scholars program; the DRI award at the University of Maryland; Pitt Momentum Funds award and CRC at the University of Pittsburgh; and the ISSSF grant from the University of Iowa.

References

Andres, L.; et al. 2019. Doing More With Less: Smarter Subsidies for Water Supply and Sanitation. World Bank. https://openknowledge.worldbank.org/handle/10986/32277. Accessed: 2022-8-14.

AyushBurak Uzkent, K.; Uzkent, B.; Burke, M.; Lobell, D.; and Ermon, S. 2021. Generating interpretable poverty maps using object detection in satellite images. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 4410–4416.

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2007. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19: 137.

Brunsdon, C.; et al. 1999. Some notes on parametric significance tests for geographically weighted regression. *Journal of regional science*, 39(3): 497–524.

Chen, X.-L.; Zhao, H.-M.; Li, P.-X.; and Yin, Z.-Y. 2006. Remote sensing image-based analysis of the relationship between urban heat island and land use/cover changes. *Remote sensing of environment*, 104(2): 133–146.

Delaire, C.; Peletz, R.; Haji, S.; Kones, J.; Samuel, E.; Easthope-Frazer, A.; Charreyron, E.; Wang, T.; Feng, A.; Mustafiz, R.; et al. 2020. How much will safe sanitation for all cost? Evidence from five cities. *Environmental Science & Technology*, 55(1): 767–777.

Finn, C.; et al. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Goodchild, M. F.; and Li, W. 2021. Replication across space and time must be weak in the social and environmental sciences. *Proceedings of the National Academy of Sciences*, 118(35): e2015759118.

Gupta, J.; et al. 2021. Spatial Variability Aware Deep Neural Networks (SVANN): A General Approach. *ACM Trans. on Intelligent Systems and Technology (TIST)*.

Huang, S.-W.; Lin, C.-T.; Chen, S.-P.; Wu, Y.-Y.; Hsu, P.-H.; and Lai, S.-H. 2018. Auggan: Cross domain adaptation with gan-based data augmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 718–731.

Jean, N.; Burke, M.; Xie, M.; Davis, W. M.; Lobell, D. B.; and Ermon, S. 2016. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301): 790–794.

Jia, X.; Xie, Y.; Li, S.; Chen, S.; Zwart, J.; Sadler, J.; Appling, A.; Oliver, S.; and Read, J. 2021. Physics-Guided Machine Learning from Simulation Data: An Application in Modeling Lake and River Systems. In *2021 IEEE International Conference on Data Mining (ICDM)*, 270–279. IEEE.

Jia, X.; et al. 2019. Classifying heterogeneous sequential data by cyclic domain adaptation: An application in land cover detection. In *SDM*. SIAM.

Li, J.; Chen, E.; Ding, Z.; Zhu, L.; Lu, K.; and Shen, H. T. 2020. Maximum density divergence for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 43(11): 3918–3930.

Organization, W. H.; et al. 2019. *Progress on household drinking water, sanitation and hygiene 2000-2017: special focus on inequalities*. World Health Organization.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2010. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2): 199–210.

Rußwurm, M.; et al. 2020. Meta-learning for few-shot land cover classification. In *CVPR Workshops*, 200–201.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7167–7176.

UN. 2022. United Nations' Sustainable Development Goals. https://sdgs.un.org/goals. Accessed: 2022-11-30.

Wang, M.; and Deng, W. 2018. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153.

Xie, M.; Jean, N.; Burke, M.; Lobell, D.; and Ermon, S. 2016. Transfer learning from deep features for remote sensing and poverty mapping. In *Thirtieth AAAI Conference on Artificial Intelligence*.

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

Xie, Y.; He, E.; Jia, X.; Chen, W.; Skakun, S.; Bao, H.; Jiang, Z.; Ghosh, R.; and Ravirathinam, P. 2022. Fairness by "Where": A Statistically-Robust and Model-Agnostic Bi-Level Learning Framework. In *Thirty-Sixth AAAI conference on artificial intelligence*.

Yan, H.; Ding, Y.; Li, P.; Wang, Q.; Xu, Y.; and Zuo, W. 2017. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2272–2281.

Ye, J.; Sun, L.; Du, B.; Fu, Y.; Tong, X.; and Xiong, H. 2019. Co-prediction of multiple transportation demands based on deep spatio-temporal neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 305–313.

Yuan, Z.; et al. 2018. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *ACM SIGKDD*, 984–992.

Zhang, J.; Zheng, Y.; and Qi, D. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-first AAAI conference on artificial intelligence*.