Efficient Poverty Mapping using Deep Reinforcement Learning

Kumar Ayush*
Department of Computer Science
Stanford University
kayush@cs.stanford.edu

Burak Uzkent*
Department of Computer Science
Stanford University
buzkent@cs.stanford.edu

Marshall Burke
Department of Earth Science
Stanford University
mburke@stanford.edu

David Lobell
Department of Earth Science
Stanford University
dlobell@stanford.edu

Stefano Ermon
Department of Computer Science
Stanford University
ermon@cs.stanford.edu

Abstract

The combination of high-resolution satellite imagery and machine learning have proven useful in many sustainability-related tasks, including poverty prediction, infrastructure measurement, and forest monitoring. However, the accuracy afforded by high-resolution imagery comes at a cost, as such imagery is extremely expensive to purchase at scale. This creates a substantial hurdle to the efficient scaling and widespread adoption of high-resolution-based approaches. To reduce acquisition costs while maintaining accuracy, we propose a reinforcement learning approach in which free low-resolution imagery is used to dynamically identify where to acquire costly high-resolution images, prior to performing a deep learning task on the high-resolution images. We apply this approach to the task of poverty prediction in Uganda, building on an earlier approach that used object detection to count objects and use these counts to predict poverty. Our approach exceeds previous performance benchmarks on this task while using 80% fewer high-resolution images. Our approach could have application in many sustainability domains that require high-resolution imagery.

1 Introduction

When combined with machine learning, satellite imagery has proven broadly useful for a range of computer vision tasks including object detection [9], object tracking [23, 24], cloud removal [18] and sustainability-related tasks, from poverty prediction [8, 1, 19, 2, 29] to infrastructure measurement [3] to forest and water quality monitoring [5] to the mapping of informal settlements [12]. Compared to coarser (10-30m) publicly-available imagery [4], high-resolution ($< 1m$) imagery has proven particularly useful for these tasks because is often able to resolve specific objects or features that are critical for downstream tasks but that are undetectable in coarser imagery.

*Equal Contribution

Preprint. Under review.

For example, recent work demonstrated an approach for predicting local-level consumption expenditure using object detection on high-resolution daytime satellite imagery [1], showing how this approach can yield interpretable predictions and also outperform previous benchmarks that rely on lower-resolution, publicly-available satellite imagery [4]. This additional information, however, typically comes at a cost, as high-resolution satellite imagery must be purchased from private providers. Additionally, processing high-resolution images is computationally more expensive than the coarser resolution ones [25, 30, 13, 10, 27, 14, 6]. Given these costs, deploying these models at scale using high-resolution imagery quickly becomes cost-prohibitive for most organizations and research teams, inhibiting the broader development and deployment of machine-learning based tools and insights based on these data.

To address this problem, we propose a reinforcement learning approach that uses coarse, freely-available public imagery to dynamically identify where to acquire costly high-resolution images, prior to conducting an object detection task. This concept leverages publicly available Sentinel-2 [4] images (10-30m) to sample smaller amount of high-resolution images (<1m). Our framework is inspired from the recent studies in computer vision literature that perform conditional inference to reduce computational complexity of convolutional networks in test time [22, 28].

We apply our approach to the domain of poverty prediction, and show how our approach can substantially reduce the cost of previous methods that used deep learning on high-resolution images to predict poverty [1] while maintaining or even improving their accuracy. In our study country of Uganda, we show how our approach can reduce the number of high-resolution images needed by 80%, in turn reducing the cost of making a country-wide poverty map using this approach by an estimate $2.9 million. We leave the exploration of our cost-aware adaptive framework for other computer vision tasks using high-resolution satellite images as a future work.

2 Poverty Mapping from Remote Sensing Imagery

Poverty is typically measured using consumption expenditure, the value of all the goods and services consumed by a household in a given period. A household or individual is said to be poverty stricken if their measured consumption expenditure falls below a defined threshold (currently $1.90 per capita per day). We focus on this consumption expenditure as our outcome of interest, using “poverty" as shorthand for “consumption expenditure" throughout the paper. While typical household surveys measure consumption expenditure at the household level, publicly available data typically only release geo-coordinate information at the “cluster" level – which is a village in rural areas and a neighborhood in urban areas. Efforts to predict poverty have thus focused on predicting at the cluster level (or more aggregated levels) [1].

Earlier work [1] demonstrated state-of-the-art results for predicting village-level poverty using high-resolution satellite imagery, and showed how such predictions could be made with an interpretable model. In particular, this work trained an object detector to obtain classwise object counts (buildings, trucks, passenger vehicles, railway vehicles, etc.) in high-resolution images, and then used these counts in a regression model to predict poverty. Not only were these categorical features predictive of poverty, but their counts had clear and intuitive relationships with the outcome of interest. The cost of this accuracy and interpretability was the high-resolution imagery, which typically must be purchased for $10-20 per km^2 from private providers.

We build on these earlier approaches here. Let $\{(\mathcal{H}_i, \mathcal{L}_i, y_i, c_i)\}_{i=1}^{N}$ be a set of N villages surveyed, where $c_i = (c_i^{lat}, c_i^{long})$ is the latitude and longitude coordinates for cluster i, and $y_i \in \mathbb{R}$ is the corresponding average poverty index for a particular year. For each cluster i, we can acquire both high-resolution and low-resolution satellite imagery corresponding to the survey year, $\mathcal{H}_i \in \mathbb{R}^{W \times H \times B}$, a $W \times H$ image with B channels, and $\mathcal{L}_i \in \mathbb{R}^{W/D \times H/D \times B}$, a $W/D \times H/D$ image with B channels. Here D represents a scalar to show the resolution difference between low-resolution and high-resolution images. Following [1], our goal is to learn a regressor f_r to predict the poverty index y_i using $\mathcal{L}_i$ and only limited informative regions of $\mathcal{H}_i$.

3 Dataset

Socio-economic Data. Our ground truth dataset consists of data on consumption expenditure (poverty) from Living Standards Measurement Study (LSMS) survey conducted in Uganda by the

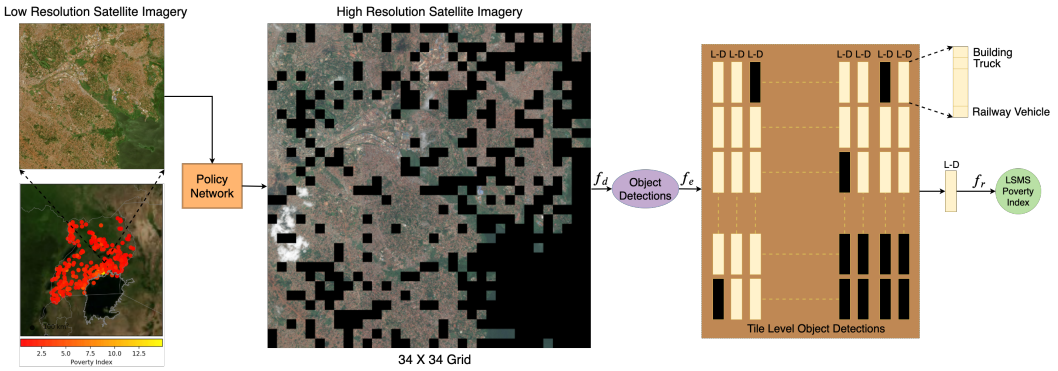


Figure 1: Schematic overview of the proposed approach. The Policy Network uses cheaply available Sentinel-2 low-resolution image representing a cluster to output a set of actions representing unique 1000×1000 px high-resolution tiles in the 34×34 grid. Then object detection is performed on the sampled HR tiles (black regions represent dropped tiles) to obtain the corresponding class-wise object counts (L-dimensional vectors). Finally, the classwise object counts vectors corresponding to the acquired HR tiles are added element-wise to get the final feature vector representing the cluster. Our reinforcement learning approach dynamically identifies where to acquire high-resolution images, conditioned on cheap, low-resolution data, before performing object detection, whereas the previous work [1] exhaustively uses all the HR tiles representing a cluster for poverty mapping, making their method expensive and less practical.

Uganda Bureau of Statistics between 2011 and 2012 [21]. The survey consists of data from 2,716 households in Uganda, which are grouped into unique locations called clusters. The latitude and longitude location, $c_i = (c_i^{lat}, c_i^{long})$, of a cluster $i = \{1, 2, \ldots, N\}$ is given, with noise of up to 5 km added in each direction by the surveyers to protect respondent privacy. Individual household locations in each cluster i are also withheld to preserve anonymity. We have N=320 clusters in the survey which we use to test the performance of our method in terms of predicting the average poverty index, y_i, for a group i. For each c_i, the survey measures the poverty level by the per capital daily consumption in dollars which we refer to as the "LSMS poverty score" for simplicity like [1]. Fig. 1 (bottom left corner) visualizes the surveyed locations on the map along with their corresponding LSMS poverty scores, revealing that a high percentage of surveyed locations have relatively low consumption expenditure values.

Satellite Imagery. We acquire both high-resolution and low-resolution satellite imagery for Uganda. The high-resolution satellite imagery, $\mathcal{H}_i$, corresponding to cluster c_i (roughly, a village or neighborhood) is represented by T=34×34=1156 images of 1000×1000 pixels each with 3 channels, arranged in a 34×34 square grid. This corresponds to a 10km×10km spatial neighborhood centered at c_i. A large neighborhood is considered to deal with up-to 5km of random noise in the cluster coordinates that has been added by the survey organization to protect respondent privacy. These high-resolution images come from DigitalGlobe satellites with 3 bands (RGB) and 30cm pixel resolution. Formally, we represent all the high-resolution images corresponding to c_i as a sequence of T tiles as $\mathcal{H}_i = \{H_i^j\}_{j=1}^T$.

We also acquire low-resolution satellite imagery, $\mathcal{L}_i$, corresponding to cluster c_i is represented by a single image of 1014×1014 pixels with 3 channels. These mages come from Sentinel-2 with 3 bands (RGB) and 10m pixel resolution and are freely available to the public. Each image corresponds to the same 10km×10km spatial neighborhood centered at c_i, however the resolution is much lower – each Sentinel-2 pixel corresponds to roughly 1000 pixels from the high-resolution imagery. Because of this low-resolution, it is not possible to perform fine-grained object detection just using these images. Fig. 1 illustrates an example cluster from Uganda.

4 Fine-grained Object Detection on High-Resolution Satellite Imagery

Similar to [1], we use an intermediate object detection phase to obtain categorical features (classwise object counts) from high-resolution tiles of a cluster. Due to lack of object annotations for satellite images from Uganda, we use the same transfer learning strategy as in [1] by training an object

detector (YOLOv3 [15]) on xView [9], one of the largest and most diverse publicly available overhead imagery datasets for object detection with 10 parent-level and 60 child-level classes. Earlier work [1] studied both parent-level and child-level detectors and empirically find that not only the parent-level object detection features are better for poverty regression but at the same time are more suited for interpretability due to household level descriptions. Thus, we train YOLOv3 detector using parent-level classes (see x-axis labels of Fig. 2).

As described in Section 3, each $\mathcal{H}_i$ representing a cluster is a set of T high-resolution images, $\{H_i^j\}_{j=1}^T$. To obtain a baseline model that uses all the high-resolution imagery available, we follow the protocol in [1] and run the trained YOLOv3 object detector on each 1000×1000px tile (*i.e.* H_i^j) to get the correspoding set of object detections and each detection is denoted by a tuple (x_c, y_c, w, h, l, s), where x_c and y_c represent the center coordinates of the bounding box, w and h represent the width and height of the bounding box, l and s represent the object class label and class confidence score. For each tile H_i^j of $\mathcal{H}_i$, we get n_i^j object detections. Similar to [1], we use these n_i^j object detections to generate a L-dimensional vector, $\mathbf{v}_i^j \in \mathbb{R}^L$ (where L=10 is the number of object labels/classes), by counting the number of detected objects in each class. This process results in T L-dimensional vectors, $\mathbf{v}_i = \{\mathbf{v}_i^j\}_{j=1}^T$ which can be aggregated into a single L-dimensional categorical feature vector $\mathbf{m}_i$ by summing over the tiles: $\mathbf{m}_i = \sum_{j=1}^T \mathbf{v}_i^j$. This classwise object counts can be used in a regression model for poverty estimation [1]. [1] exhaustively uses all T=1156 HR tiles of a cluster for poverty estimation. In contrast, we propose to use a method that adaptively selects informative regions for high-resolution acquisition conditioned on the publicly available, low-resolution data. We describe our solution in the next section.

5 Adaptive Tile Selection

Due to the large acquisition cost of HR images, it is non-trivial and expensive to deploy models based on HR imagery at scale. For this reason, we propose an efficient tile selection framework to capture relevant fine level information such as classwise object counts for downstream tasks. We represent the HR image covering a spatial cluster i centered at $c_i = (c_i^{lat}, c_i^{lon})$ as $\mathcal{H}_i \in R^{W \times H \times B}$ where W, H and B represent height width and number of bands. Additionally, we represent the LR image of the same spatial cluster i as $\mathcal{L}_i \in R^{W/D, H/D, B}$ where D represents a scalar for the number of pixels in width and height. For example, in the case of Sentinel-2 (10 m GSD), we have $D = 30$ times smaller number of pixels than the high-resolution DigitalGlobe images (0.3m GSD). With an adaptive approach, our task is to acquire only small subset of $\mathcal{H}_i$ conditionally on $\mathcal{L}_i$ while not hurting the performance in our downstream tasks that uses object counts from the cluster i. This adaptive method is formulated as a two-step episodic Markov Decision Process (MDP), similar to [26]. In the first step, we adaptively sample HR tiles and in the second step, we run them through a pre-trained detector.

Adaptive Selection. The first module of our framework finds tiles to sample/acquire, conditioned on the low spatial resolution image covering a cluster. In this direction, a cluster-level HR image $\mathcal{H}_i = (H_i^1, H_i^2, \ldots, H_i^T)$ is divided into equal-size non-overlapping tiles, where T is the number of tiles. In this set up, we model $\mathcal{H}_i$ as a latent variable as it is not directly observed and it is inferred from the random variable $\mathcal{L}_i$. We associate each tile, H_i^j, of $\mathcal{H}_i$ with an L-dimensional classwise object counts feature represented as $\mathbf{v}_i = \{\mathbf{v}_i^1, \ldots, \mathbf{v}_i^T\}$. We then model the policy network to only choose tiles where there is desirable number of object counts as: **Acquire** H_i^j if $||\mathbf{v}_i^j||_1 > P$ where P is determined by the policy network that uses a reward function characterized by the user. Similar to $\mathcal{H}_i$, we decompose the random variable $\mathcal{L}_i$ as $\mathcal{L}_i = (l_i^1, l_i^2, \ldots, l_i^T)$ where l_i^j represents the lower spatial resolution version (from Sentinel-2) of H_i^j.

Modeling the Policy Network's Input and Output. In a simple scenario, we can take a single binary action for each H_i^j whether to acquire it or not conditioned on l_i^j. However, we believe that choosing multiple actions representing different disjoint subtiles of tile H_i^j can help us avoid sampling areas of tile H_i^j where there are no objects of interest. In another setup, we can use a large T to have smaller tiles and take a single action to sample tile H_i^j or not. However, this introduces run-time complexity since we need to run the policy network more number of times to cover a cluster. For these reasons, we divide tile H_i^j into S number of disjoint subtiles as $H_i^j = (h_i^{j,1}, h_i^{j,2}, ..., h_i^{j,S})$.

In the first step of the MDP, the agent observes l_i^j and outputs a binary action array, $\mathbf{a}_i^j \in \{0,1\}^S$, where $a_i^{j,k} = 1$ represents acquisition of the HR version of the k-th subtile of H_i^j *i.e.* $h_i^{j,k}$. The subtile sampling policy, parameterized by θ_p, is formulated as

$$\pi(\mathbf{a}_i^j | l_i^j; \theta_p) = p(\mathbf{a}_i^j | l_i^j; \theta_p) \tag{1}$$

where $\pi(l_i^j; \theta_p)$ is a function mapping the observed LR image to a probability distribution over subtile sampling actions $\mathbf{a}_i^j$. The joint probability distribution over the random variables l_i^j, $\mathbf{v}_i^j$, H_i^j, and action $\mathbf{a}_i^j$, can be written as

$$p(H_i^j, l_i^j, \mathbf{v}_i^j, \mathbf{a}_i^j) = p(H_i^j)\, p(\mathbf{v}_i^j | H_i^j)\, p(l_i^j | H_i^j)\, p(\mathbf{a}_i^j | l_i^j; \theta_p) \tag{2}$$

Object Detection. In the second step of the MDP, the agent runs the object detection on the selected HR subtiles. Conditioned on $\mathbf{a}_i^j$, it observes HR subtiles if necessary and produces $\hat{\mathbf{v}}_i^j$, a L-dimensional classwise object counts vector. We find the object counts with our adaptive framework using a pre-trained object detector f_d (parameterized by θ_d) as:

$$\hat{\mathbf{v}}_i^{j,k} = \begin{cases} f_d(h_i^{j,k}) & \text{if} \quad a_i^{j,k} = 1 \\ \mathbf{0} & \text{else} \end{cases} \tag{3}$$

Then, we compute the tile level object counts as $\hat{\mathbf{v}}_i^j = \sum_{k=1}^{S} \hat{\mathbf{v}}_i^{j,k}$. Finally, we define our overall cost function J as:

$$\max_{\theta_p} J(\theta_p, \theta_d) = \mathbb{E}_p[R(\mathbf{a}_i^j, \hat{\mathbf{v}}_i^j, \mathbf{v}_i^j)], \tag{4}$$

where the reward depends on $\mathbf{a}_i^j$, $\hat{\mathbf{v}}_i^j$, $\mathbf{v}_i^j$. Our goal is to learn the parameters θ_p given a pre-trained object detector θ_d to maximize the objective being a function of the reward function. We detail the reward function in Section 6.

6 Modeling and Optimization of the Policy Network

Modeling the Policy Network. In the previous section, in high level we formulated the task of efficient HR subtile selection as a two step episodic MDP. In this section, we model how to learn the policy distribution for subtile sampling. In this study, we have $T = 1156$ number of tiles as we have a 34×34 grid of images. In this case, each tile consists of 1000×1000 pixels. As mentioned in the previous section, we divide each tile into S=4 subtiles of 250×250 pixels each. In this study, similar to [26] we model the action likelihood function of the policy network, f_p, using the product of bernoulli distributions as:

$$\pi_c(\mathbf{a}_i^j | l_i^j; \theta_p) = \prod_{k=1}^{S} (s_i^{j,k})^{a_i^{j,k}} (1 - s_i^{j,k})^{(1 - a_i^{j,k})} \tag{5}$$

$$s_i^j = f_p(l_i^j; \theta_p) \tag{6}$$

We use a sigmoid function to transform logits to probabilistic values, $s_i^{j,k} \in [0, 1]$.

Optimization of the Policy Network. Next we detail optimization procedure for the policy network. Previously defined objective function as shown in Eq. 4 is not differentiable w.r.t the policy network parameters, θ_p. This is because we discretize continuous action probabilities from the policy network to perform binary action of acquiring or not acquiring a subtile. To overcome this, we use one of the model-free reinforcement learning algorithms called Policy Gradient [20]. Our final objective function as shown below includes the reward function as well as action likelihood distribution which can be differentiated w.r.t θ_p.

$$\nabla_{\theta_p} J = \mathbb{E}\left[R(\mathbf{a}_i^j, \hat{\mathbf{v}}_i^j, \mathbf{v}_i^j) \nabla_{\theta_p} \log \pi_{\theta_p}(\mathbf{a}_i^j | l_i^j) \right], \tag{7}$$

Our objective function relies on mini-batch Monte-Carlo sampling to approximate the expectation. Especially, in scenarios where we can not afford large mini-batches, we can have highly oscillating

expectations which results in large variance. As this can de-stabilize the optimization, we use the self-critical baseline [16], A, to reduce the variance.

$$\nabla_{\theta_p} J = \mathbb{E}\left[A \sum_{k=1}^{S} \nabla_{\theta_p} \log(s_i^{j,k}\mathbf{a}_i^{j,k} + (1 - s_i^{j,k})(1 - \mathbf{a}_i^{j,k}))\right] \tag{8}$$

$$A(\mathbf{a}_i^j, \overline{\mathbf{a}}_i^j) = R(\mathbf{a}_i^j, \hat{\mathbf{v}}_i^j, \mathbf{v}_i^j) - R(\overline{\mathbf{a}}_i^j, \overline{\mathbf{v}}_i^j, \mathbf{v}_i^j) \tag{9}$$

where $\overline{\mathbf{a}}_i^j$ represents the baseline action vector. To get it, we use the most likely action vector proposed by the policy network: *i.e.*, $\overline{\mathbf{a}}_i^{j,k} = 1$ if $s_i^{j,k} > 0.5$ and $\overline{\mathbf{a}}_i^{j,k} = 0$ otherwise. Finally, in this study we use temperature scaling [20] to adjust exploration/exploitation trade-off during optimization time as

$$s_i^{j,k} = \alpha s_i^{j,k} + (1 - \alpha)(1 - s_i^{j,k}). \tag{10}$$

Setting α to a large value results in sampling from the learned policy whereas the small values lead to sampling from random policy.

Modeling the Reward Function. The proposed framework uses the policy gradient reinforcement learning algorithm to learn the parameters of the policy network f_p, adjusting weights θ_p to increase the expected reward value. Thus, it is crucial to design a reward function reflecting the desired characteristics of an efficient subtile selection method from a cluster representing a large area. The desired outcome from our adaptive strategy is to reduce the *image acquisition cost* drastically by sampling smaller subset of tiles. Taking this into account, we design a dual reward function that encourages dropping as many subtiles as possible while successfully approximating the classwise object counts. We define R as follows:

$$R = R_{acc}(\hat{\mathbf{v}}_i^j, \mathbf{v}_i^j) + R_{cost}(\mathbf{a}_i^j) \tag{11}$$

$$R_{acc} = -||\mathbf{v}_i^j - \hat{\mathbf{v}}_i^j||_1 \tag{12}$$

$$R_{cost} = \lambda(1 - |\mathbf{a}_i^j|_1/S) \tag{13}$$

where R_{acc} is object counts approximation accuracy and R_{cost} represents the image acquisition cost with λ as its coefficient. The R_{acc} term encourages acquiring a subtile when the counts difference between the object counts from fixed HR subtile sampling policy and the adaptive policy is positive. We increase the reward *linearly* with the smaller number of acquired subtiles for the cost component. See appendix for the pseudocode and other implementation details.

7 Experiments

Poverty Estimation. Previous work [1] exhaustively performed object detection on all the HR tiles representing a cluster i to obtain T L-dimensional vectors, $\mathbf{v}_i = \{\mathbf{v}_i^j\}_{j=1}^T$, which are then aggregated into a single L-dimensional categorical feature vector, $\mathbf{m}_i$, by summing over the tiles *i.e.* $\mathbf{m}_i = \sum_{j=1}^T \mathbf{v}_i^j$. This was subsequently used in a regression model to predict poverty score for cluster i. Using our adaptive method, we obtain $\hat{\mathbf{m}}_i = \sum_{j=1}^T \hat{\mathbf{v}}_i^j$, which is an approximate classwise counts vector for cluster i. Following [1], we consider Gradient Boosting Decision Trees as the regression model to estimate the poverty index, y_i, given the cluster level categorical feature vector (classwise object counts), $\mathbf{m}_i$/$\hat{\mathbf{m}}_i$. We use Pearson's r^2 to quantify the model performance. Invariance under seperate changes in scale between two variables allows Pearson's r^2 to provide insights into the ability of the model at distinguishing poverty levels.

Training and Evaluation. We have N=320 clusters in the survey. We divide the dataset into a 80-20 train-test split. We train a GBDT model using object counts features ($\mathbf{m}_i$) based on all HR tiles of the clusters in the trainset. We use the clusters in the trainset to train the policy network for adaptive tile selection. The trained policy network is then used to acquire informative HR tiles for each test cluster *i.e* for a test cluster i, the policy network selects HR tiles (subsequently used to obtain $\hat{\mathbf{m}}_i$) conditioned on low-resolution input representing the cluster. The obtained $\hat{\mathbf{m}}_i$ is then passed through the trained GBDT model to get the poverty score y_i. See appendix for more implementation details.

Baselines and State-of-the-Art Models. We compare our method with the following: (a) *No Patch Dropping*, where we simply use all the HR tiles in $\mathcal{H}_i$ to get the classwise object counts features

	No Dropping [1]	Fixed-18	Random-25	Stochastic-25	Nightlights	Ours (Dry sea.)	Ours (Wet sea.)
r^2	0.53	0.43	0.34	0.26	0.45	0.51	**0.62**
HR Acquisition	1.0	0.18	0.25	0.25	0.12	0.19	**0.19**

Table 1: LSMS poverty score prediction results in Pearson's r^2 for various methods. *HR Acquisition* represents the fraction of HR tiles acquired.

(same as [1]), (b) *Fixed Policy-X* samples $X\%$ HR tiles from the center of a cluster, (c) *Random Policy-X* samples $X\%$ HR tiles randomly from a cluster, (d) *Stochastic Policy-X*, samples $X\%$ HR tiles where the survival likelihood of a tile decays w.r.t the euclidean distance from the cluster center, and (e) *Nightlights*, where we use Nightlight Images (48×48 px) representing the clusters in Uganda and sample only those HR tiles which have non-zero nighttime light intensities.

Additionally, since Sentinel-2 imagery is freely available, we perform a comparative analysis of the effect of season on the ability of the policy network at approximating classwise object counts. We thus acquired two sets of low-resolution imagery, one from dry-season (Dec - Feb) in Uganda and other from wet season (March-May, Sept-Nov) corresponding to the survey year. Seasonality is likely highly relevant in our agrarian setting, where crops are grown during the wet season and much related market activity is highly seasonal. We hypothesize that greenery in low-resolution imagery during wet season will better indicate which patches might contain useful economic information.

Quantitative Analysis. Fig. 2 compares the ability of various methods at approximating the classwise object counts. It shows the number of objects missed on an average across clusters for each parent class, where we can see that our method (using wet season imagery) is better able to approximate the "true object counts" (we use object detector predictions on all the HR tiles as a proxy for true values) compared to baseline methods and our method (using dry season imagery). Table 1 presents the corresponding HR acquisition fractions revealing that out method is able to identify informative tiles leading to a lower HR requirement compared to various baselines. Table 1 also shows the results of poverty prediction in Uganda for our proposed method against these baselines and previous benchmarks. Our model (wet season) achieves **0.62** r^2 and substantially outperforms the published state-of-the-art results [1] (**0.53** r^2) while using around **80%** fewer satellite images. We similarly outperform other baselines as well. A scatter plot of GBDT LSMS poverty score predictions v.s. ground truth is shown in Fig. 3. It can be seen that the GBDT model can maintain explainability of a large fraction of the variance based on object counts identified from the sampled HR tiles using our method, compared to [1] that exhaustively uses all HR tiles.

The superior performance of our approach relative to other baselines and to previous work that uses all tiles suggests that our model is learning to sample the correct regions in a large image. The previous work [1] show that *Trucks* had a higher impact on LSMS poverty score prediction and

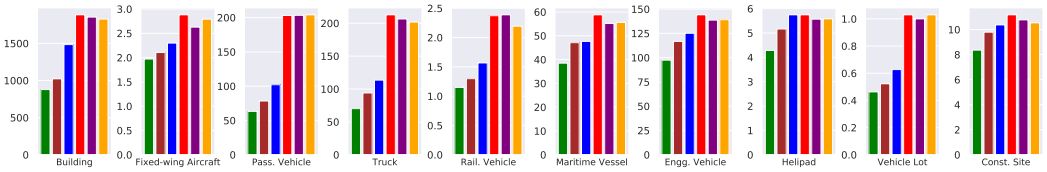
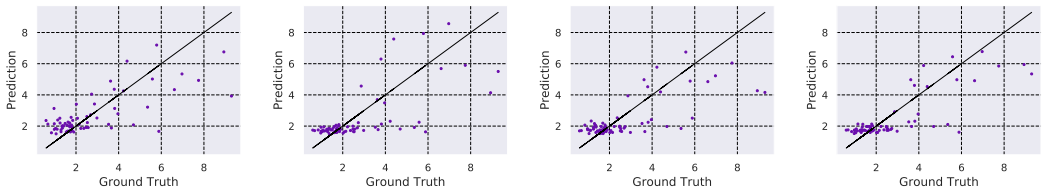


Figure 2: Number of objects missed on an average across clusters for each parent-level class. The colored bars in each subplot from left-right are: Ours (wet season), Ours (dry season), Nightlight, Fixed-18, Random-25, Stochastic-25.



(a) No Dropping (b) Nightlights (c) Ours (Dry Season) (d) Ours (Wet Season)

Figure 3: LSMS poverty score regression results of GBDT.

(a) (b) (c) (d) (e)

Figure 4: (a) High-Resolution Satellite Imagery representing a cluster. (b) Sentinel-2 Imagery of the cluster from dry season. (c) Corresponding HR acquisitions when dry-season imagery is input to the Policy Network. (d) Sentinel-2 Imagery of the cluster from wet season. (e) Corresponding HR acquisitions when wet-season imagery is input to the Policy Network.

explained that regions with good transport connectivity tend to have higher *#Trucks*. Fig. 4 (d) and (e) present an example highlighting the ability of the policy network (conditioned on wet season imagery) to identify such regions leading to a more accurate approximation of *#Trucks* (see Fig. 2) thus leading to an improved performance.

Analysis based on Season. We observe that presence of greenery during wet season allows the policy network to better identify the informative regions containing objects, compared to when trained with dry season Sentinel-2 imagery as input to the network. Figure 4 presents an example cluster, where it can be seen that training the policy network using wet season imagery better assists the network at sampling informative tiles whereas the one trained using dry season imagery misses out on some important tiles thus hindering performance in a downstream task. See Appendix for more visuals.

Performance/Sampling Trade-off. Next, we analyze the trade-off between accuracy (GBDT regression performance) and HR sampling rate controlled by hyperparameter λ in Eq. 13. We intentionally increase/decrease λ to quantify the effect on the policy network. As seen in Fig. 5, the policy network samples less HR tiles (0.09) when we increase λ to 2.0 and the r^2 goes down to 0.48. On the other hand, when we set λ to 1.0, we get optimal results in r^2 at 0.18 HR acquisition fraction.

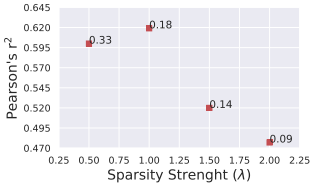
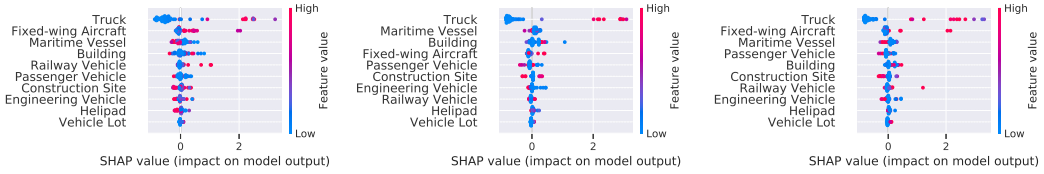


Figure 5: Trade-off between Pearson's r^2 and coefficent of image acquisition cost (λ). Text accompanying the points represents HR acquisition fraction.

Impact on Interpretability. An important contribution of [1] was to introduce model interpretability allowing successful application of such methods in many policy domains. They use Tree SHAP (Tree SHapley Additive exPlanations) [11], a game theoretic approach to explain the output of tree-based models, to explain the effect of individual features on poverty predictions and show that the presence of trucks appeared to be particularly useful for measuring local scale poverty. Here, we show that in addition to closely approximating the classwise object counts, our method retains the same findings in terms of interpretability as that of [1]. Fig. 6 shows the plots of SHAP values of every feature for every cluster for three different methods. The features are sorted by the sum of SHAP value magnitudes over all samples. It can be seen that our method still maintains that *#Trucks* tends to have a higher impact on the model's output. We also observe that



(a) No Dropping (b) Ours (Dry Season) (c) Ours (Wet Season)

Figure 6: Summary of the effects of all features using SHAP, showing the distribution of the impacts each feature has on the model output. Color represents the feature value (red high, blue low).

ordering of features in terms of SHAP values is fairly similar between the *No Dropping* approach [1] and our method (using wet season imagery) giving strong evidence that wet season imagery is better for such adaptive solutions.

Cost saving. Current pricing for high-resolution (30cm) three-band (RGB) imagery is $10-20 per km^2. Given that Uganda is 240k km^2 in land area, creating a poverty map using our method would save roughly $2.9 million if imagery costs $15 per km^2, given that we would only need 20% of the country to be tiled. This represents a potentially large cost saving if our approach is to be scaled at country or continent scale.

8 Conclusion

In this study, we increase the efficiency of recent methods of predicting consumption expenditure using object counts from high-resolution satellite images. To achieve this, we proposed a novel reinforcement learning setup to conditionally acquire high-resolution tiles. We designed a cost-aware reward function to reflect real-world constraints – i.e. budget and GPU availability – and then trained a policy network to approximate object counts in a given location as closely as possible given these constraints. We show that our approach reduces the number of high-resolution images needed by 80% while improving downstream poverty estimation performance relative to multiple other approaches, including a method that exhaustively uses all high-resolution images from a location. Future work includes application of our adaptive method to other sustainability-related computer vision tasks using high-resolution images at large scale.

References

[1] Kumar Ayush, Burak Uzkent, Marshall Burke, David Lobell, and Stefano Ermon. Generating interpretable poverty maps using object detection in satellite images. *arXiv preprint arXiv:2002.01612*, 2020.

[2] Joshua Blumenstock, Gabriel Cadamuro, and Robert On. Predicting poverty and wealth from mobile phone metadata. *Science*, 350(6264):1073–1076, 2015.

[3] Gabriel Cadamuro, Aggrey Muhebwa, and Jay Taneja. Assigning a grade: Accurate measurement of road quality using satellite imagery. *arXiv preprint arXiv:1812.01699*, 2018.

[4] M. Drusch, U. Del Bello, S. Carlier, O. Colin, V. Fernandez, F. Gascon, B. Hoersch, C. Isola, P. Laberinti, P. Martimort, A. Meygret, F. Spoto, O. Sy, F. Marchese, and P. Bargellini. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote Sensing of Environment*, 120:25 – 36, 2012.

[5] Jonathan RB Fisher, Eileen A Acosta, P James Dennedy-Frank, Timm Kroeger, and Timothy M Boucher. Impact of satellite imagery spatial resolution on land use classification accuracy and modeled water quality. *Remote Sensing in Ecology and Conservation*, 4(2):137–149, 2018.

[6] Mingfei Gao, Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. Dynamic Zoom-in Network for Fast Object Detection in Large Images. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6926–6935, Salt Lake City, UT, USA, June 2018. IEEE.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

[9] Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018.

[10] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[11] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[12] Ron Mahabir, Arie Croitoru, Andrew T Crooks, Peggy Agouris, and Anthony Stefanidis. A critical review of high and very high-resolution remote sensing approaches for detecting and mapping slums: Trends, challenges and emerging opportunities. *Urban Science*, 2(1):8, 2018.

[13] Zibo Meng, Xiaochuan Fan, Xin Chen, Min Chen, and Yan Tong. Detecting small signs from large images. In *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 217–224. IEEE, 2017.

[14] Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, Honolulu, HI, USA, July 2017. IEEE.

[15] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[16] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

[17] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015.

[18] Vishnu Sarukkai, Anirudh Jain, Burak Uzkent, and Stefano Ermon. Cloud removal from satellite images using spatiotemporal generator networks. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1796–1805, 2020.

[19] Evan Sheehan, Chenlin Meng, Matthew Tan, Burak Uzkent, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Predicting economic development using geolocated wikipedia articles. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2698–2706, 2019.

[20] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[21] Uganda Bureau of Statistics UBOS. Uganda national panel survey 2011/2012. *Uganda*, 2012.

[22] Burak Uzkent and Stefano Ermon. Learning when and where to zoom with deep reinforcement learning. *arXiv preprint arXiv:2003.00425*, 2020.

[23] Burak Uzkent, Aneesh Rangnekar, and Matthew Hoffman. Aerial vehicle tracking by adaptive fusion of hyperspectral likelihood maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 39–48, 2017.

[24] Burak Uzkent, Aneesh Rangnekar, and Matthew J Hoffman. Tracking in aerial hyperspectral videos using deep kernelized correlation filters. *IEEE Transactions on Geoscience and Remote Sensing*, (99):1–13, 2018.

[25] Burak Uzkent, Evan Sheehan, Chenlin Meng, Zhongyi Tang, Marshall Burke, David Lobell, and Stefano Ermon. Learning to interpret satellite images in global scale using wikipedia. *arXiv preprint arXiv:1905.02506*, 2019.

[26] Burak Uzkent, Christopher Yeh, and Stefano Ermon. Efficient object detection in large images using deep reinforcement learning. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1824–1833, 2020.

[27] Christian Wojek, Gyuri Dorkó, André Schulz, and Bernt Schiele. Sliding-windows for rapid object class localization: A parallel technique. In *Joint Pattern Recognition Symposium*, pages 71–81. Springer, 2008.

[28] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8817–8826, 2018.

[29] Christopher Yeh, Anthony Perez, Anne Driscoll, George Azzari, Zhongyi Tang, David Lobell, Stefano Ermon, and Marshall Burke. Using publicly available satellite imagery and deep learning to understand economic well-being in africa. *Nature Communications*, 11(1):1–11, 2020.

[30] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2110–2118, 2016.

A Pseudocode

Input: $(\mathcal{L}_i, \mathcal{H}_i) \quad i = \{1, 2, ..., N\}$
for $j \leftarrow 1 \; to \; T$ **do**
  $s_i^j \leftarrow f_p(l_i^j; \theta_p)$
  $s_i^j \leftarrow \alpha + (1 - s_i^j)(1 - \alpha)$
  $\mathbf{a}_i^j \sim \pi(\mathbf{a}_i^j | s_i^j)$
  for $k \leftarrow 1 \; to \; S$ **do**
    $\hat{\mathbf{v}}_i^{j,k} = f_d(h_i^{j,k}) \bigodot \mathbf{a}_i^{j,k}$
  end
  $\hat{\mathbf{v}}_i^j = \sum_{k=1}^{S} \hat{\mathbf{v}}_i^{j,k}$
  Evaluate Reward $R(\mathbf{a}_i^j, \hat{\mathbf{v}}_i^j, \mathbf{v}_i^j)$
  $\theta_p \leftarrow \theta_p + \nabla \theta_p$
end

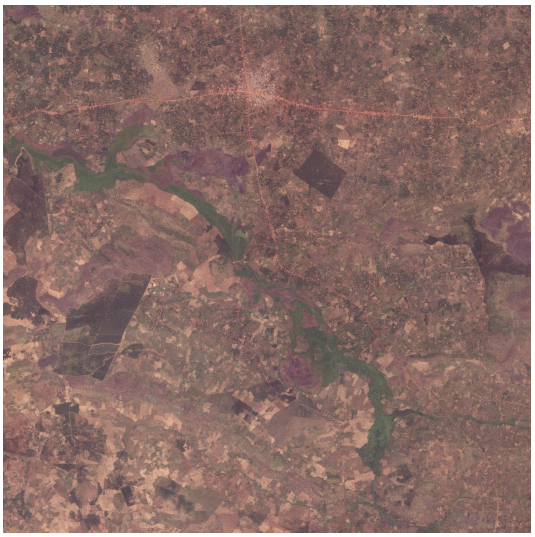
Algorithm 1: Pseudo-code for the Proposed Adaptive Algorithm. T and S represent the number of tiles and subtiles.

B Implementation Details

Policy Networks. To parameterize the policy network, we use ResNet [7] with 32 layers pretrained on the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) dataset [17]. We train the policy network using 2 NVIDIA 1080ti GPUs.

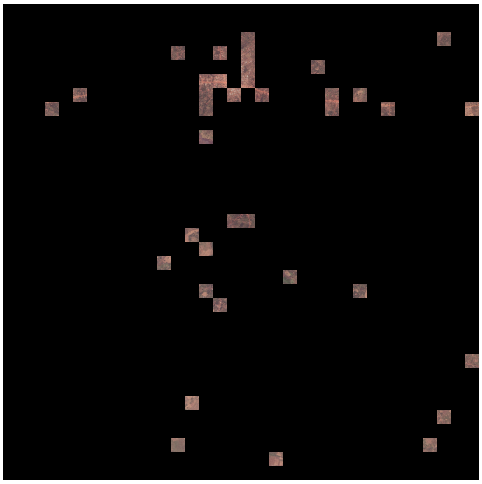
Object Detectors. Our object detector use the YOLOv3 architecture [15], chosen for its reasonable trade off between accuracy on small objects and run-time performance. The backbone network, DarkNet-53, is pre-trained on ImageNet. Following [1], we perform transfer learning by training the detector on xView dataset and running it on the Uganda HR patches.

C Visualizations

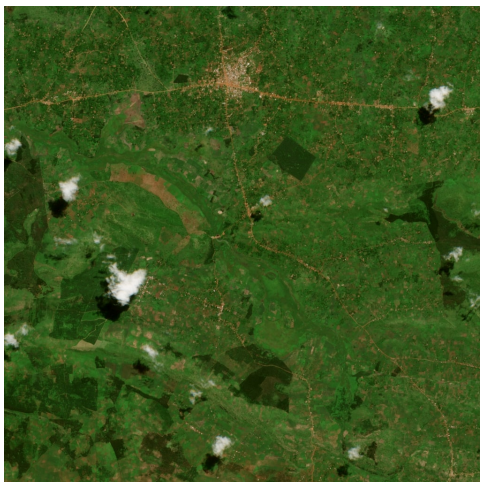


(a) High-Resolution Satellite Imagery (downsampled by 34 for visualization).

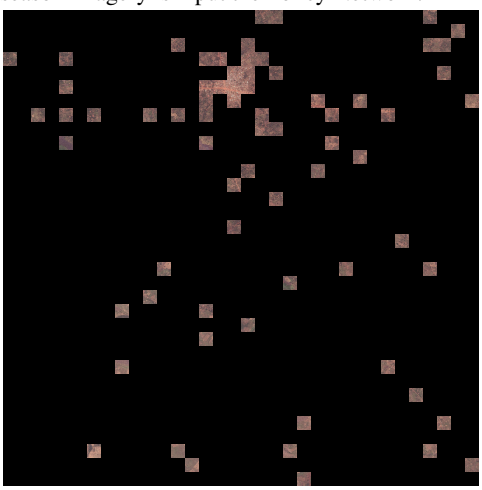
(b) Sentinel-2 Imagery for a cluster from dry season.



(c) Corresponding HR acquisitions when dry-season imagery is input the Policy Network.

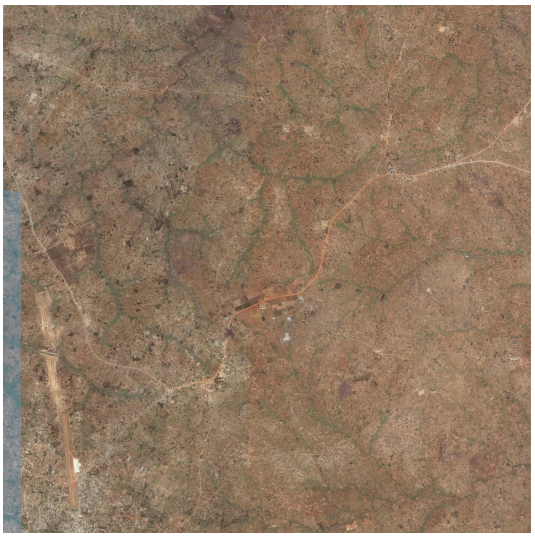


(d) Sentinel-2 Imagery for a cluster from wet season.



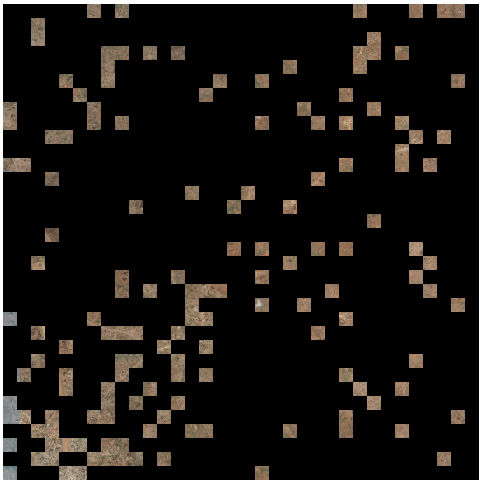
(e) Corresponding HR acquisitions when wet-season imagery is input the Policy Network.

Figure 7: Example 1

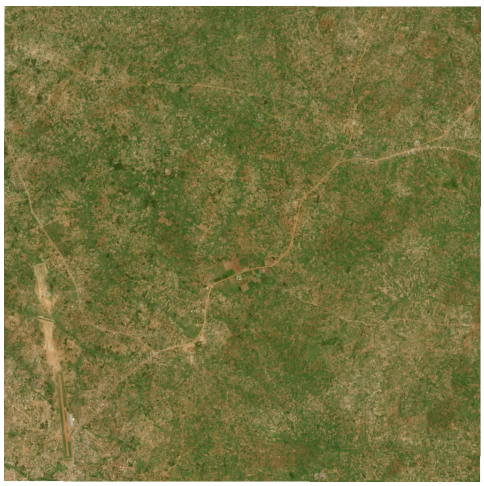


(a) High-Resolution Satellite Imagery (downsampled for visualization).

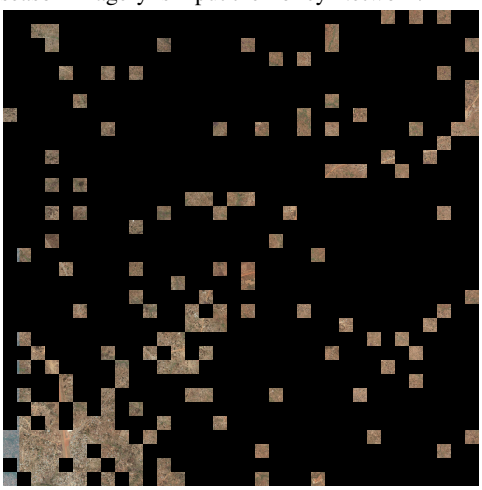
(b) Sentinel-2 Imagery for a cluster from dry season.



(c) Corresponding HR acquisitions when dry-season imagery is input the Policy Network.



(d) Sentinel-2 Imagery for a cluster from wet season.



(e) Corresponding HR acquisitions when wet-season imagery is input the Policy Network.

Figure 8: Example 2