COMBINING MACHINE LEARNING AND SATELLITE IMAGERY FOR SUSTAINABILITY CHALLENGES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Kumar Ayush
August 2022

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Master of Science.

(Stefano Ermon) Principal Co-Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Master of Science.

(Marshall Burke) Principal Co-Adviser

Approved for the Stanford University Committee on Graduate Studies

Abstract

The lack of reliable data in developing countries is a major obstacle to sustainable development, food security, and disaster relief. Poverty data, for example, is typically scarce, sparse in coverage, and labor-intensive to obtain. Remote sensing data such as high-resolution satellite imagery, on the other hand, is becoming increasingly available and inexpensive. Recent computer vision advances in using satellite imagery to predict economic indicators like poverty have shown increasing accuracy, but they do not generate features that are interpretable to policymakers, inhibiting adoption by practitioners. My research goal in this thesis is to develop an efficient, explainable, and transferable method that uses object detection from high-resolution satellite imagery for sustainable development tasks like poverty, even in settings with limited training data. 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 further 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.

Object detection, image classification, and semantic segmentation models are important components of an effective computational framework for various sustainability related tasks. The performance of these models are highly dependent on the appropriate pre-training of their backbone networks. A main purpose of such pre-training is to learn good representations (i.e., features) that can be transferred to these downstream tasks of detection, segmentation, classification, etc., by fine-tuning on limited training data. In this thesis, we also propose novel methods to improve unsupervised/self-supervised learning (especially for network pre-training) that can effectively boost performance of such downstream tasks. Specifically, we propose novel training methods that exploit the spatio-temporal structure of remote sensing data in a contrastive learning framework. We also propose a negative data augmentation strategy in a contrastive learning framework for self-supervised representation learning on images and videos, achieving improved performance on downstream image classification, object detection, and action recognition tasks. Additionally, we also propose methods to drastically reduce pre-training cost and provide strong performance boosts.

Acknowledgments

First and foremost, I would like to express my utmost gratitude to my advisor, Stefano Ermon. I have had the good fortune of working with him since the onset of my master's degree. I still remember my elation when I received a positive response from him to my email exploring the opportunity to work with him. I still remember how start-struck I was when I met Stefano for the first time during the first lecture of his course on Deep Generative Models. Since then I have been fortunate to cherish a long list of experiences with him involving diverse research projects. His high level vision and low level guidance have helped me grow both as a researcher as well as a mentor for other students. Every meeting with him ends ups teaching me a new way of thinking which has greatly helped me in my time as a student researcher at Stanford. I find myself truly lucky to have Stefano as my advisor. Thank you for this experience which I will treasure forever.

This letter of acknowledgement would be incomplete without expressing my gratitude to my co-advisors, Marshall Burke and David Lobell. They are stalwarts in their field of sustainability. I am amazed by the amount of economic information they possess about the world. With their patient advising, and clarity in thought, every research meeting ends on a note of optimism. Working with them has allowed me to broaden my horizon outside of pure computer science and AI. Their sense of humour always offers a great comic relief during the meetings.

My experience at Stanford would be incomplete without the company I enjoyed in both the Sustainability and Artificial Intelligence Lab and the Ermon Group. Thank you to my research mentors: Burak Uzkent and Jiaming Song for always being available with the perfect solutions to all my challenges. Burak has been a brilliant mentor and became a close friend along the way. I am always in awe of Jiaming. Every conversation with him gives me intellectual satisfaction and brings a smile on my face.

I would like to thank all my co-authors whose work is featured in this thesis (in alphabetical order): Abhishek Sinha, Chenlin Meng, Evan Sheehan, Kelly He, Kumar Tanmay, and Shuvam Chakraborty. Thank you to all the members of both the Stanford Sustainability and Artificial Intelligence Lab and the Ermon Group for the helpful discussions that contributed to this work. This experience has nothing been short of amazing and I feel so privileged to work with such brilliant minds. While I wrap up my journey at Stanford, I sincerely hope to be in touch with each one of you.

I would be remiss in not acknowledging my friends from Stanford who have become my family away from home: Abhishek, Prabhat, Shreya, Chetanya, Sarthak, Ayush Kanodia, Nishant, Sundar, Soham, Ashwin,

Aditya Gera, Kopal, Advay, Prerna, Aparna, Praveen, Varun, and Trisha. Thank you for making this time at Stanford so memorable and supporting me through all the highs and lows. Finally, I would like to thank my younger brother, Tanmay, for being my buddy and for always being a comic relief. Last and definitely the most, I will forever be indebted to my parents, Manoj and Preeti, for their unconditional love and giving me the liberty to follow my dreams and supporting me with all means possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The number one goal of the United Nations Sustainable Development Goals is to "end poverty, in all its forms, everywhere." While much has been done to achieve this objective, there still remain vast regions of the world where extreme poverty continues to be a persistent and endemic problem. Progress is hampered by a stubborn lack of data regarding key social, environmental, and economic indicators that would inform research and policy. Many nations lack the governmental, social, physical, and financial capabilities to conduct large-scale data gathering operations and costly on the-ground surveys to identify and distribute aid to the most needy communities [6]. Unfortunately, this data scarcity is typical of other SDGs as well: healthcare and mortality data [7], infrastructure and transportation statistics, economic well-being and educational achievement information, food security [8] and wealth inequality assessment, among many others. This lack of detailed and consistent information contributes to delayed or suboptimal financial and physical responses from both regional governments and international aid organizations. There have been numerous attempts to remedy this shortage of socioeconomic information through the combination of machine learning with cheap, globally available data streams such as social media or remotely sensed data. In particular, with regard to healthcare data, [9] use Twitter as an information-rich source from which to track and predict disease levels and the public's concerns regarding pathogens, while [10] pursue a similar task via the analysis of Google trends. [11] mine the tweets of Indonesian citizens in order to understand food shortages as well as analyze food security and predict food prices on a granular level. [12] also utilize social media posts to track adverse reactions to drugs with the hope of increasing the scale of data available in the healthcare space. Finally, [13] use mobile phone metadata to attempt to predict poverty levels for data-sparse regions of Rwanda, and [14] successfully perform detailed traffic prediction using the open-source geospatial dataset OpenStreetMap (OSM) [15].

Remote sensing, particularly satellite imagery, is perhaps the only cost-effective technology able to provide data at a global scale. Within ten years, commercial services are expected to provide sub-meter resolution

images everywhere at a fraction of current costs. This level of temporal and spatial resolution could provide a wealth of data towards sustainable development. There has been some works like [16] that show how a transfer learning approach that uses coarse information from nighttime satellite images to extract features from daytime high-resolution imagery can also predict asset wealth variation across multiple African countries. However, research in exploiting machine learning and satellite imagery for sustainability related tasks is still in the initial stages and much work is yet to be done.

In the spirit of leveraging machine learning and satellite imagery for various sustainable development tasks, we analyse the drawbacks of existing systems and propose novel methods to tackle the existing challenges that create a substantial hurdle to the efficient scaling and widespread adoption of high-resolution satellite imagery based approaches.

1.1.1 Thesis Outline

In this thesis, we highlight the disadvantages of traditional surveying methods for organizations to measure the well-being of developing regions and other sustainability related tasks. We attempt to remedy this shortage of socioeconomic information through the combination of machine learning with globally available data such as remotely sensed data and hope that our methods can be employed as a cheap but effective alternative to measure various economic and livelihood indicators. The thesis proceeds as follows: In **Chapter 2**, we attempt to predict consumption expenditure from high resolution satellite images. We propose an efficient, explainable, and transferable method that combines object detection and regression. **Chapter 3** discusses disadvantages of our method in **Chapter 2** and proposes a novel reinforcement learning setup to conditionally acquire high-resolution tiles to increase the efficiency of our method of predicting consumption expenditure. Image classification, object detection, and semantic segmentation models are important components of an effective computational framework for various sustainability related tasks. The performance of these models are highly dependent on the appropriate pre-training of their backbone networks. A main purpose of such pre-training is to learn good representations (i.e., features) that can be transferred to these downstream tasks of detection, segmentation, classification, etc., by fine-tuning on limited training data. In the next three chapters, we explore and propose novel methods to improve unsupervised/self-supervised learning (for pre-training) that can effectively boost performance of such downstream tasks. **Chapter 4** focuses on providing a self-supervised learning framework for remote sensing data, where unlabeled data is often plentiful but labeled data is scarce. By leveraging spatially aligned images over time to construct temporal positive pairs in contrastive learning and geo-location in the design of pre-text tasks, we are able to close the gap between self-supervised and supervised learning on image classification, object detection and semantic segmentation on remote sensing and other geo-tagged image datasets. **Chapter 5** provides a new form of data augmentation, called negative data augmentation as a method to incorporate prior knowledge through out-of-distribution (OOD) samples. We leverage NDA for unsupervised representation learning in images (including satellite imagery) and videos and show improved results on image and action recognition various benchmark datasets. **Chapter 6** presents filtering methods to efficiently pre-train on large scale datasets conditioned on the transfer

learning task. Our methods drastically reduce pre-training cost and provide strong performance boosts in downstream tasks. The thesis concludes in **Chapter 7** where we summarize our contributions and briefly list potential future directions and applications.

1.1.2 Previously Published Papers

Most contributions in this thesis have first appeared as various publications/preprints. These publications/preprints are: [4] (**IJCAI 2020**, Chapter 2), [17] (**AAAI 2021**, Chapter 3), [18] (Under review in **ICCV 2021**, Chapter 4), [19] (**ICLR 2021**, Chapter 5) and [20] (Under review in **ICCV 2021**, Chapter 6). My other work [21] (Under review in **ICCV 2021**) is out of the context of this thesis.

1.2 Background and Related Work

In this section, we dive into literature related to estimation of economic indicators (like poverty) from satellite imagery, self-supervised and unsupervised representation learning methods, data augmentation, and pre-training methods.

1.2.1 Poverty Prediction from Imagery

Multiple studies have sought to use various types of satellite imagery for local-level prediction of economic livelihoods. [16] train a CNN to extract features in high-resolution daytime images using low-resolution nighttime images as labels, and then use the extracted features to predict asset wealth and consumption expenditure across five African countries. [22] train a CNN to predict African asset wealth from lower-resolution (30m) multi-spectral satellite imagery, achieving similar performance to [16]. These approaches provide accurate methods for predicting local-level asset wealth, but the CNN-extracted features used to make predictions are not easily interpretable, and performance is substantially lower when predicting consumption expenditure rather than asset wealth.

Two related works use object detection approaches to predicting economic livelihoods from imagery. [23] show how information on the make and count of cars detected in Google Streetview imagery can be used to predict socioeconomic outcomes at local level in the US. This work is promising in a developed world context where streetview imagery is available, but challenging to employ in the developing world where such imagery is very rare, and where car ownership is uncommon.

1.2.2 Self-Supervised Learning

Self-supervised methods use unlabeled data to learn representations that are transferable to downstream tasks (image classification, object detection, semantic segmentation). Two commonly seen self-supervised methods are *pre-text task* and *contrastive learning*.

Pre-text tasks Pre-text task based learning [24, 25, 26, 27, 28, 29] can be used to learn feature representations when data labels are not available. [30] rotates an image and then trains a model to predict the rotation angle. [31] trains a network to perform colorization of a grayscale image. [32] represents an image as a grid, permuting the grid and then predicting the permutation index. In this study, we use *geo-location classification* as a pre-text task, in which a deep network is trained to predict a coarse geo-location of where in the world the image might come from.

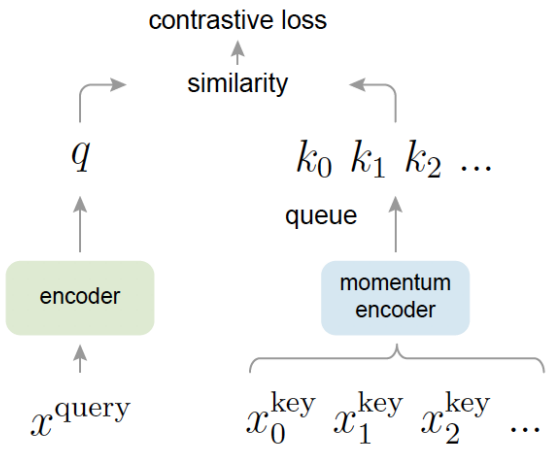


Figure 1.1: Momentum Contrast (MoCo) trains a visual representation encoder by matching an encoded query q to a dictionary of encoded keys using a contrastive loss. The dictionary keys $\{k_0, k_1, k_2, ...\}$ are defined on-the-fly by a set of data samples. The dictionary is built as a queue, with the current mini-batch enqueued and the oldest mini-batch dequeued, decoupling it from the mini-batch size. The keys are encoded by a slowly progressing encoder, driven by a momentum update with the query encoder. This method enables a large and consistent dictionary for learning visual representations. Figure taken from [3].

Contrastive Learning Recent self-supervised contrastive learning approaches such as MoCo [2] (see Figure 1.1), MoCo-v2 [5], SimCLR [33], PIRL [24], and FixMatch [34] have demonstrated superior performance and have emerged as the fore-runner on various downstream tasks. The intuition behind these methods are to learn representations by pulling positive image pairs from the same instance closer in latent space while pushing negative pairs from difference instances further away. These methods, on the other hand, differ in the type of contrastive loss, generation of positive and negative pairs, and sampling method.

Although growing rapidly in self-supervised learning area, contrastive learning methods have not been explored on large-scale remote sensing dataset. In this work, we provide a principled and effective approach for improving representation learning using MoCo-v2 [2] for remote sensing data as well geo-located conventional datasets.

1.2.3 Representation Learning in Remote Sensing

Unsupervised Learning in Remote Sensing Images Unlike in traditional computer vision areas, unsupervised learning on remote sensing domain has not been studied comprehensively. Most of the studies utilize small-scale datasets specific to a small geographical region [35, 36, 37, 38, 39], a few classes [40] or a highly-specific modality, i.e. hyperspectral images [41, 42]. Most of these studies focus on the UCM-21 dataset [43] consisting of less than 1,000 images from 21 classes. A more recent study [44] proposes large-scale weakly supervised learning using a multi-modal dataset consisting of satellite images and paired geo-located wikipedia articles. While being effective, this method requires each satellite image to be paired to its corresponding article, limiting the number of images that can be used.

Geography-aware Computer Vision Geo-location data has been studied extensively in prior works. Most of these studies utilizes geo-location of an image as a prior to improve image recognition accuracy [45, 46, 47, 48, 49]. Other studies [50, 51, 52, 53] use geo-tagged training datasets to learn how to predict the geo-location of previously unseen images at test time. In our study, we leverage geo-tag information to improve unsupervised and self-supervised learning methods.

1.2.4 Negative Sampling and Data Augmentation

In several machine learning settings, negative samples are produced from a statistical generative model. [54] aim to generate negative data using GANs for semi-supervised learning and novelty detection while we are concerned with efficiently creating negative data to improve generative models and self-supervised representation learning. [55] also propose an alternative theoretical framework that relies on access to an oracle which classifies a sample as valid or not, but do not provide any practical implementation. [56] use adversarial training to generate hard negatives that fool the discriminator for NLP tasks whereas we obtain NDA data from positive data to improve image generation and representation learning. [57] use a GAN to learn the negative data distribution with the aim of classifying positive-unlabeled (PU) data whereas we do not have access to a mixture data but rather generate negatives by transforming the positive data.

In contrastive unsupervised learning, common negative examples are ones that are assumed to be further than the positive samples semantically. Word2Vec [58] considers negative samples to be ones from a different context and CPC-based methods [59] such as momentum contrast [3], the negative samples are data augmentations from a different image. Our work considers a new aspect of "negative samples" that are neither generated from some model, nor samples from the data distribution. Instead, by applying negative data augmentation (NDA) to existing samples, we are able to incorporate useful inductive biases that might be difficult to capture otherwise [60].

1.2.5 Transfer Learning

Unconditional Transfer Learning The success of deep learning on datasets with increased sample complexity has brought transfer learning to the attention of the research community. Pre-training networks on

ImageNet-1k has been shown to be a very effective way of initializing weights for a target task with small sample size [61, 62, 63, 64, 65]. However, all these studies use unconditional pre-training as they employ the weights pre-trained on the full source dataset, which can be computationally infeasible for future large scale datasets.

Conditional Transfer Learning [66, 67, 68], on the other hand, filter the pre-training dataset conditioned on target tasks. [67, 69] use greedy class-specific clustering based and learn image representations with an encoder trained on the massive JFT-300M dataset [70], which dramatically increases cost. [66] trains a number of expert models on many subsets of the pre-training dataset and uses their performance to weight source images, however this method is naturally quite computationally expensive. Our methods differ from the past works as we take into account pre-training dataset filtering efficiency, adaptability to different tasks and settings, and target task performance.

Chapter 2

Interpretable Poverty Estimation from Satellite Images

2.1 Introduction

Accurate measurements of poverty and related human livelihood outcomes critically shape the decisions of governments and humanitarian organizations around the world, and the eradication of poverty remains the first of the United Nations Sustainable Development Goals [71]. However, reliable local-level measurements of economic well-being are rare in many parts of the developing world. Such measurements are typically made with household surveys, which are expensive and time consuming to conduct across broad geographies, and as a result such surveys are conducted infrequently and on limited numbers of households. For example, Uganda (our study country) is one of the best-surveyed countries in Africa, but surveys occur at best every few years, and when they do occur often only survey a few hundred villages across the whole country (Fig. 2.1). Scaling up these ground-based surveys to cover more regions and more years would likely be prohibitively expensive for most countries in the developing world [72]. The resulting lack of frequent, reliable local-level information on economic livelihoods hampers the ability of governments and other organizations to target assistance to those who need it and to understand whether such assistance is having its intended effect.

To tackle this data gap, an alternative strategy has been to try to use passively-collected data from non-traditional sources to shed light on local-level economic outcomes. Such work has shown promise in measuring certain indicators of economic livelihoods at local level. For instance, [13] show how features extracted from cell phone data can be used to predict asset wealth in Rwanda, and [73] show how applying NLP techniques to Wikipedia articles can be used to predict asset wealth in multiple developing countries, and [16] show how a transfer learning approach that uses coarse information from nighttime satellite images to extract features from daytime high-resolution imagery can also predict asset wealth variation across multiple African countries.

These existing approaches to using non-traditional data are promising, given that they are inexpensive and inherently scalable, but they face two main challenges that inhibit their broader adoption by policymakers. The first is the outcome being measured. While measures of asset ownership are thought to be relevant metrics for understanding longer-run household well-being [74], official measurement of poverty requires data on consumption expenditure (i.e. the value of all goods consumed by a household over a given period), and existing methods have either not been used to predict consumption data or perform much more poorly when predicting consumption than when predicting other livelihood indicators such as asset wealth [16]. Second, interpretability of model predictions is key for whether policymakers will adopt machine-learning based approaches to livelihoods measurement, and current approaches attempt to maximize predictive performance rather than interpretability. This tradeoff, central to many problems at the interface of machine learning and policy [75], has yet to be navigated in the poverty domain.

Here we demonstrate an interpretable computational framework for predicting local-level consumption expenditure using object detection on high-resolution (30cm) daytime satellite imagery. We focus on Uganda, a country with existing high-quality ground data on consumption where performance benchmark are available. We first train a satellite imagery object detector on a publicly available, global scale object detection dataset, called xView [76], which avoids location specific training and provides a more general object detection model. We then apply this detector to high resolution images taken over hundreds of villages across Uganda that were measured in an existing georeferenced household survey, and use extracted counts of detected objects as features in a final prediction of consumption expenditure. We show that not only does our approach substantially outperform previous performance benchmarks on the same task, it also yields features that are immediately and intuitively interpretable to the analyst or policy-maker.

2.2 Poverty Estimation from Remote Sensing Data

The outcome of interest in this paper is consumption expenditure, which is the metric used to compute poverty statistics; a household or individual is said to be poor or in poverty if their measured consumption expenditure falls below a defined threshold (currently $1.90 per capita per day). Throughout the paper we use “poverty" as shorthand for “consumption expenditure", although we emphasize that the former is computed from the latter. 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), and we do the same here. Let $\{(x_i, y_i, c_i)\}_{i=1}^{N}$ 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 high resolution satellite imagery corresponding to the survey year $x_i \in \mathcal{I} = \mathbb{R}^{W \times H \times B}$, a $W \times H$ image with B channels. Following [16], our goal is to learn a regressor $f : \mathcal{I} \to \mathbb{R}$ to predict the poverty index y_i from x_i. Here our goal is to find a regressor that is both accurate

and *interpretable*, where we use the latter to mean a model that provides insight to a policy community on why it makes the predictions it does in a given location.

2.2.1 Dataset

Socio-economic Data. The dataset comes from field Living Standards Measurement Study (LSMS) survey conducted in Uganda by the Uganda Bureau of Statistics between 2011 and 2012 [77]. The LSMS survey we use here 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 privacy. Individual household locations in each cluster i are also withheld to preserve anonymity. We use all $N = 320$ clusters in the survey 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. For simplicity, in this study, we name the per capital daily consumption in dollars as LSMS poverty score. We visualize the chosen locations on the map as well as their corresponding LSMS poverty scores in Fig. 2.1. From the figure, we can see that the surveyed locations are scattered near the border of states and high percentage of these locations have relatively low poverty scores.

	Building	Fixed-Wing Aircraft	Passenger Vehicle	Truck	Railway Vehicle	Maritime Vessel	Engineering Vehicle	Helipad	Vehicle Lot	Construction Site
AP	0.40	0.59	0.42	0.27	0.39	0.24	0.17	0.0	0.012	0.0003
AR	0.62	0.65	0.76	0.56	0.49	0.47	0.37	0.0	0.06	0.006

Table 2.1: Class wise performance (average precision and recall) of YOLOv3 when trained using parent level classes (10 classes).

Uganda Satellite Imagery. The satellite imagery, x_i corresponding to cluster c_i is represented by $K = 34 \times 34 = 1156$ images of $W = 1000 \times H = 1000$ pixels with $B = 3$ channels, arranged in a 34×34 square grid. This corresponds to a 10 km $\times$ 10 km spatial neighborhood centered at c_i. We consider a large neighborhood to deal with the noise in the cluster coordinates.

The images come from DigitalGlobe satellites with three bands (RGB) and 30cm pixel resolution. Fig. 2.1 illustrates an example cluster from Uganda. Formally, we represent all the images corresponding to c_i as a sequence of K tiles as $x_i = \{x_i^j\}_{j=1}^K$.

2.3 Fine-grained Detection on Satellite Images

Contrary to existing methods for poverty mapping which perform end-to-end learning [16, 73, 22], we use an intermediate object detection phase to first obtain interpretable features for subsequent poverty prediction. However, we do not have object annotations for satellite images from Uganda. Therefore, we perform transfer learning by training an object detector on a different but related source dataset $\mathcal{D}^s$.

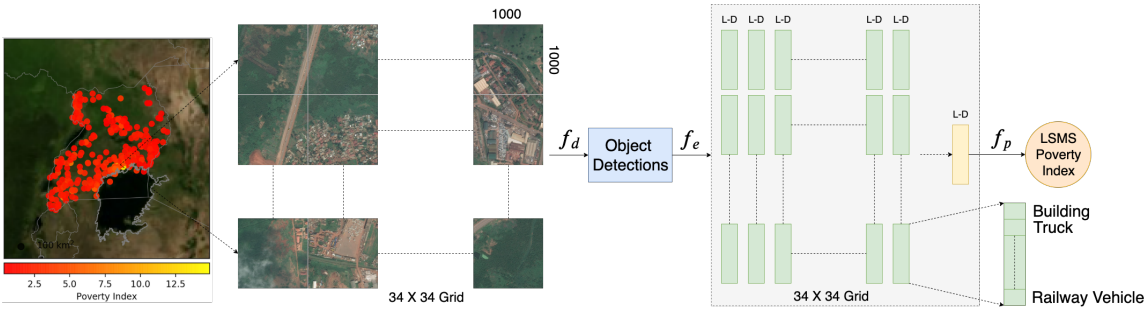


Figure 2.1: Pipeline of the proposed approach. For each cluster we acquire 1156 images, arranged in a 34×34 grid, where each image is an RGB image of 1000×1000 px. We run an object detector on its 10×10 km^2 neighborhood and obtain the object counts for each xView class as a L-dimensional categorical feature vector. This is done by running the detector on every single image in a cluster, resulting in a $34 \times 34 \times L$ dimensional feature vector. Finally, we perform summation across the first two dimensions and get the feature vector representing the cluster, with each dimension containing the object counts corresponding to an object class. Given the cluster level feature vector, we regress the LSMS poverty score.

2.3.1 Object Detection Dataset

We use xView [76], as our source dataset. It is one of the largest and most diverse publicly available overhead imagery datasets for object detection. It covers over $1,400$ km^2 of the earth's surface, with 60 classes and approximately 1 million labeled objects. The satellite images are collected from DigitalGlobe satellites at 0.3 m GSD, aligning with the GSD of our target region satellite imagery $\{x_i\}_{i=1}^{N}$. Moreover, xView uses a tree-structured ontology of classes. The classes are organized hierarchically similar to [78, 79] where children are more specific than their parents (e.g., *fixed-wing aircraft* as a parent of *small aircraft* and *cargo plane*). Overall, there are 60 child classes and 10 parent classes.

2.3.2 Training the Object Detector

Models. Since we work on very large tiles ($\sim$3000$\times$3000 pixels), we only consider single stage detectors. Considering the trade off between run-time performance and accuracy on small objects, YOLOv3 [80] outperforms other single stage detectors [81, 82] and performs almost on par with RetinaNet [83] but 3.8 $\times$ faster [80] on small objects while running significantly faster than two-stage detectors [84, 85]. Therefore, we use YOLOv3 object detector with a DarkNet53 [80] backbone architecture.

Dataset Preparation. The xView dataset consists of 847 large images (roughly 3000×3000 px). YOLOv3 is usually used with an input image size of 416×416 px. Therefore, we randomly chip 416×416 px tiles from the xView images and discard tiles without any object of interest. This process results in 36996 such tiles of which we use 30736 tiles for training and 6260 tiles for testing.

Training and Evaluation. We use the standard per-class average precision, mean average precision (mAP), and per-class recall, mean average recall (mAR) metrics [80, 83] to evaluate our trained object detector. We fine-tune the weights of the YOLOv3 model, pre-trained on the ImageNet, using the training split of the

xView dataset. Since xView has an ontology of parent and child level classes, we train two YOLOv3 object detectors using parent level and child level classes seperately.

After training the models, we validate their performance on the test set of xView. The detector trained using parent level classes (10 classes) achieves mAP of 0.248 and mAR of 0.42. On the other hand, the one trained on child classes achieves mAP of 0.082 and mAR of 0.163. Table 2.1 shows the class-wise performance of the parent-level object detector on the test set. For comparison, lam2018xview report 0.14 mAP, but they use a separate validation and test set in addition to the training set (which are not publicly available) so the models are not directly comparable. While not state of the art, our detector reliably identifies objects, especially at the parent level.

2.3.3 Object Detection on Uganda Satellite Images

As described in Section 2.2.1, each x_i is represented by a set of K images, $\{x_i^j\}_{j=1}^K$. Each 1000×1000 px tile (i.e. x_i^j) is further chipped into 9 416×416 px small tiles (with overlap of 124 px) and fed to YOLOv3. Although the presence of objects across tile borders could decrease performance, this method is highly parallelizable and enables us to scale to very large regions. We perform object detection on $320 \times 1156 \times 9$ chips (more than 3 million images), which takes about a day and a half using 4 NVIDIA 1080Ti GPUs. In total, we detect 768404 objects. 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. In Section 2.4.1, we explain how we use these details to create interpretable features. Additionally, we experiment with object detections obtained at different confidence thresholds which we discuss in Section 2.5.1.

Transfer performance in Uganda. The absence of ground truth object annotations for our Uganda imagery $\{x_i^j\}_{j=1}^K$ prevents us from quantitatively measuring the detector's performance on Uganda satellite imagery. However, we manually annotated 10 images from the Uganda dataset together with the detected bounding boxes to measure the detector's performance on building and truck classes. We found that the detector achieves about 50%, and 45% AR for Building and Truck which is slightly lower than the AR scores for the same classes on the xView test set. We attribute this slight difference to the problem of domain shift and we plan to address this problem via domain adaptation in a future work. To qualitatively test the robustness of our xView-trained object detector, we also visualize its performance on two representative tiles in Fig. 2.2 and 2.3. The detection results prove the effectiveness of transferring the YOLOv3 model to DigitalGlobe imagery it has not been trained on.

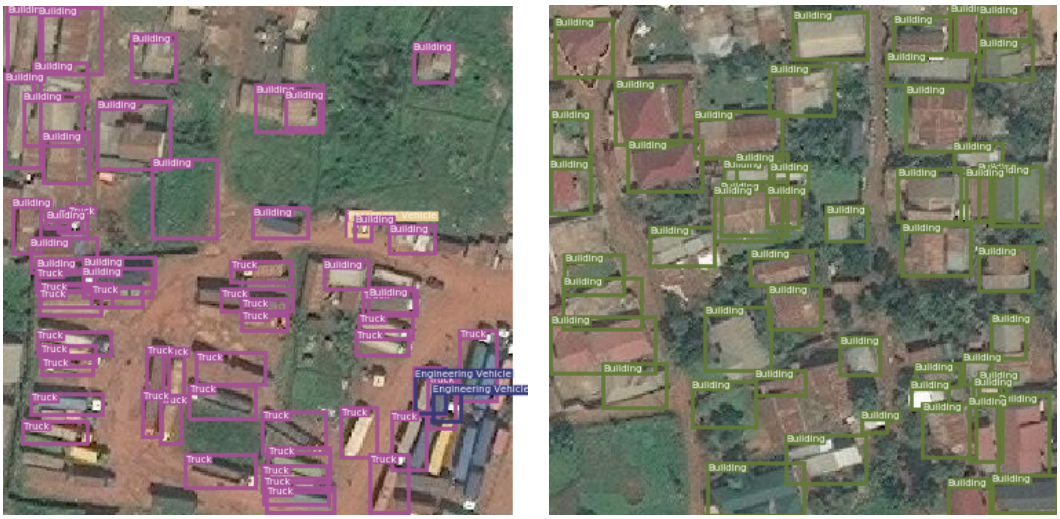


Figure 2.2: Sample detection results from Uganda. Zoom-in is recommended to visualize the bounding box classes.

Figure 2.3: Additional detection results from Uganda. Zoom-in is recommended to visualize the bounding boxes and labels. The detector can reliably detect buildings and trucks with reasonable overlap. However, it misses some of the small cars. Detecting small cars is a very challenging task due to very small number of representative pixels .

2.4 Fine-level Poverty Mapping

2.4.1 Feature Extraction from Clusters

Our object detection pipeline outputs n_i^j object detections for each tile x_i^j of x_i. We use the n_i^j object detections to generate a L-dimensional vector, $\mathbf{v}_i^j \in \mathbb{R}^L$ (where L is the number of object labels/classes), by counting the number of detected objects in each class with each object weighted by its confidence score or size or their combination (details below). This process results in K L-dimensional vectors $\mathbf{v}_i = \{\mathbf{v}_i^j\}_{k=1}^K$. Finally, we aggregate these K vectors into a single L-dimensional categorical feature vector $\mathbf{m}_i$ by summing over tiles: $\mathbf{m}_i = \sum_{j=1}^K \mathbf{v}_i^j$. While many other options are possible, in this work we explore four types of features:

Counts. *Raw object counts corresponding to each class*. Here, each dimension represents an object class and contains the number of objects detected corresponding to that class.

Confidence×Counts. *Each detected object is weighted by its class confidence score*. The intuition is to reduce the contributions of less confident detections. Here each dimension corresponds to the sum of class confidence scores of the detected objects of that class.

Size×Counts. *Each detected object is weighted by its bounding box area*. We posit that weighting based on area coverage of an object class can be an important factor. For example, an area with 10 big buildings might have a different wealth level than an area with 10 small buildings. Each dimension in $\mathbf{m}_i$ contains the sum of areas of the bounding boxes of the detected objects of that class.

(Confidence, Size)×Counts. *Each detected object is weighted by its class confidence score and the area of its bounding box*. We concatenate the *Confidence* and *Size* based features to create a $2L$-dimensional vector.

2.4.2 Models, Training and Evaluation

Given the cluster level categorical feature vector, $\mathbf{m}_i$, we estimate its poverty index, y_i with a regression model. Since we value interpretability, we consider Gradient Boosting Decision Trees, Linear Regression, Ridge Regression, and Lasso Regression. As we regress directly on the LSMS poverty index, we quantify the performance of our model using the square of the Pearson correlation coefficient (Pearson's r^2). Pearson's r^2, provides a measure of how well observed outcomes are replicated by the model. This metric was chosen so that comparative analysis could be performed with previous literature [16]. Pearson's r^2 is invariant under separate changes in scale between the two variables. This allows the metric to provide insight into the ability of the model to distinguish between poverty levels. This is relevant for many downstream poverty tasks, including the distribution of program aid under a fixed budget (where aid is disbursed to households starting with the poorest, until the budget is exhausted), or in the evaluation of anti-poverty programs, where outcomes are often measured in terms of percentage changes in the poverty metric. Due to small size of the dataset, we use a Leave-one-out cross validation (LOOCV) strategy. Since nearby clusters could have some geographic overlap, we remove clusters which are overlapping with the test cluster from the train split to avoid leaking information to the test point.

2.5 Experiments

2.5.1 Poverty Mapping Results

Quantitative Analysis. Table 2.2 shows the results of LSMS poverty prediction in Uganda. The detections are obtained using a 0.6 confidence threshold (the effect of this hyper-parameter is evaluated below). The best result of 0.539 Pearson's r^2 is obtained using GBDT trained on parent level *object counts* features (red color entry). A scatter plot of GBDT predictions v.s. ground truth is shown in Fig. 2.4. It can be seen that our

GBDT model can explain a large fraction of the variance in terms of object counts automatically identified in high resolution satellite images. To the best of our knowledge, this is the first time this capability has been shown with a rigorous and reproducible out-of-sample evaluation (see however the related but unpublished paper by [86]).

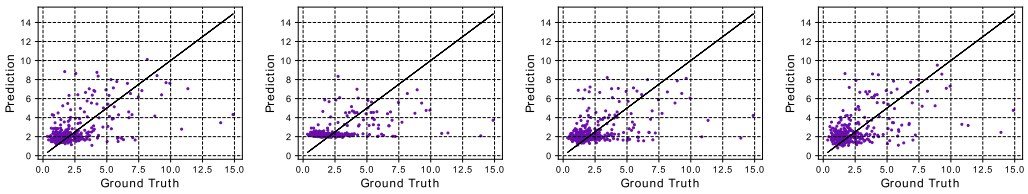
We observe that GBDT performs consistently better than other regression models across the four features we consider. As seen in Table 2.2, object detection based features deliver positive r^2 with a simple linear regression method which suggests that they have positive correlation with LSMS poverty scores. However, the main drawback of linear regression against GBDT is that it predicts negative values, which is not reasonable as poverty indices are non-negative. In general, the features are useful, but powerful regression models are still required to achieve better performance.

We also find that child-level object detections can perform better than the coarser ones (second and third best) in some cases. This is likely because although they convey more information, detection and classification is harder and less accurate at the finer level. Additionally, parent level features are more suited for interpretability, due to household level descriptions, which we show later.

— Best — Second Best — Third Best

Features/Method	**GBDT**		**Linear**		**Lasso**		**Ridge**	
	Parent	**Child**	**Parent**	**Child**	**Parent**	**Child**	**Parent**	**Child**
Counts	0.539	0.508	0.311	0.324	0.312	0.46	0.311	0.329
Confidence × Counts	0.466	0.485	0.305	0.398	0.305	0.461	0.305	0.409
Size × Counts	0.455	0.535	0.363	0.47	0.363	0.476	0.363	0.47
(Conf., Size) × Counts	0.495	0.516	0.411	0.369	0.418	0.343	0.411	0.476

Table 2.2: LSMS poverty score prediction results in Pearson's r^2 using parent level features (YOLOv3 trained on 10 classes) and child level features (YOLOv3 trained on 60 classes).



(a) Counts (b) Confidence × Counts (c) Size × Counts (d) (Conf., Size) × Counts

Figure 2.4: Regression result of GBDT using parent level counts.

Comparison to Baselines and State-of-the-Art. We compare our method with two baselines and a state-of-the-art method: (a) **NL-CNN** where we regress the LSMS poverty scores using a 2-layer CNN with Nightlight Images (48×48 px) representing the clusters in Uganda as input, (b) **RGB-CNN** where we regress the LSMS poverty scores using ImageNet [78] pretrained ResNet-18 [87] model with central tile representing c_i as input, and (c) **Transfer Learning with Nightlights**, [16] proposed a transfer learning approach where nighttime light intensities are used as a data-rich proxy.

Results are shown in Table 2.3. Our model substantially outperforms all three baselines, including published state-of-the-art results on the same task in [16]. We similarly outperform the NL-CNN baseline, a simpler version of which (scalar nightlights) is often used for impact evaluation in policy work [88]. Finally, the performance of the RGB-CNN baseline reveals the limitation of directly regressing CNNs on daytime images, at least in our setting with small numbers of labels. As discussed below, these performance improvements do not come at the cost of interpretability – rather, our model predictions are much more interpretable than each of these three baselines.

Method	**RGB-CNN**	**NL-CNN**	**[16]**	**Ours**
r^2	0.04	0.39	0.41	**0.54**

Table 2.3: Comparison with baseline and state-of-the-art methods.

Impact of Detector's Confidence Threshold. Finally, we analyze the effect of confidence threshold for object detector on the poverty prediction task in Fig. 2.5. We observe that when considering only *Counts* features, we get the best performance at 0.6 threshold. However, even for very small thresholds, we achieve around 0.3-0.5 Pearson's r^2 scores. We explore this finding in Fig. 2.5 (Right), and observe that the *ratio of classes in terms of number of bounding boxes remain similar* across different thresholds. These results imply that the ratio of object counts is perhaps more useful than simply the counts themselves – an insight also consistent with the substantial performance boost from GBT over unregularized and regularized linear models in Table 1.

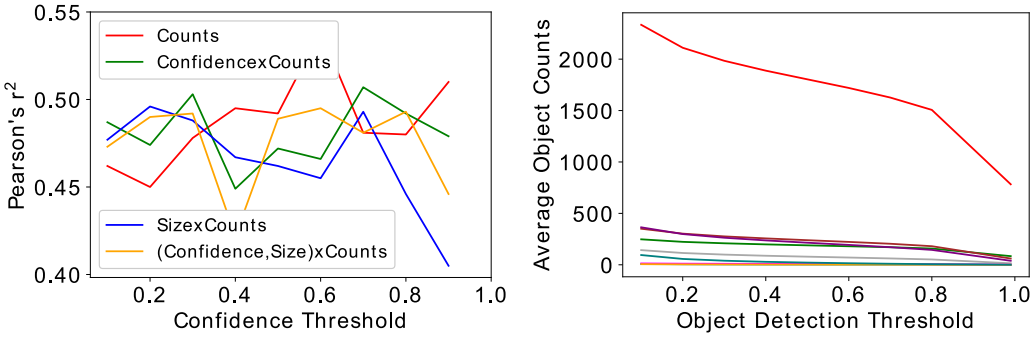


Figure 2.5: **Left:** Regression results (GBDT) using object detection features (parent level classes) at different confidence thresholds. **Right:** Average object counts across clusters for each parent class (see Table 2.1 for color coding) at difference confidence thresholds.

2.5.2 Interpretability

Existing approaches to poverty prediction using unstructured data from satellites or other sources have understandably sought to maximize predictive performance [16, 22, 73], but this has come at the cost of interpretability, as most of the extracted features used for prediction do not have obvious semantic meaning. While no quantitative data have been collected on the topic, our personal experience on multiple continents over many years is that the lack of interpretability of CNN-based poverty predictions can make policymakers understandably reluctant to trust these predictions and to use them in decision-making. Enhancing the interpretability of ML-based approaches more broadly is thought to be a key component of successful application in many policy domains [89].

Relative to an end-to-end deep learning approach, our two-step approach provides categorical features that can be easily interpreted. We now explore whether these features also have an intuitive mapping to poverty outcomes in three analyses.

Explanations via SHAP. In this section, we explain the effect of individual features on poverty score predictions using SHAP (SHapley Additive exPlanations) [90]. SHAP is a game theoretic approach to explain the output of any machine learning model. We particularly use TreeSHAP [91] which is a variant of SHAP for tree-based machine learning models. TreeSHAP significantly improves the interpretability of tree-based models through a) a polynomial time algorithm to compute optimal explanations based on game theory, b) explanations that directly measure local feature interaction effects, and c) tools for understanding global model structure based on combining many local explanations of each prediction.

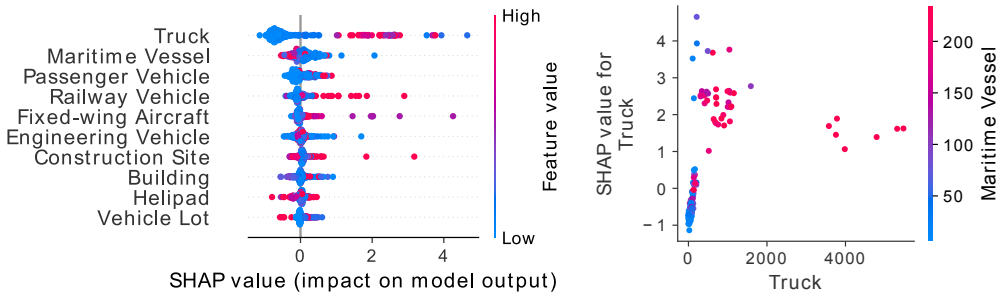


Figure 2.6: **Left**: Summary of the effects of all the features. **Right**: Dependence plot showing the effect of a single feature across the whole dataset. In both figures, the color represents the feature value (red is high, blue is low)

To get an overview of which features are most important for a model we plot the SHAP values of every feature for every sample. The plot in Figure 2.6 (left) sorts features by the sum of SHAP value magnitudes over all samples, and uses SHAP values to show the distribution of the impacts each feature has on the model output. The color represents the feature value (red high, blue low). We find that *Truck* tends to have a high impact on the model's output. Higher *#Trucks* pushes the output to a higher value and low *#Trucks* has a negative impact on the output, thereby lowering the predicted value.

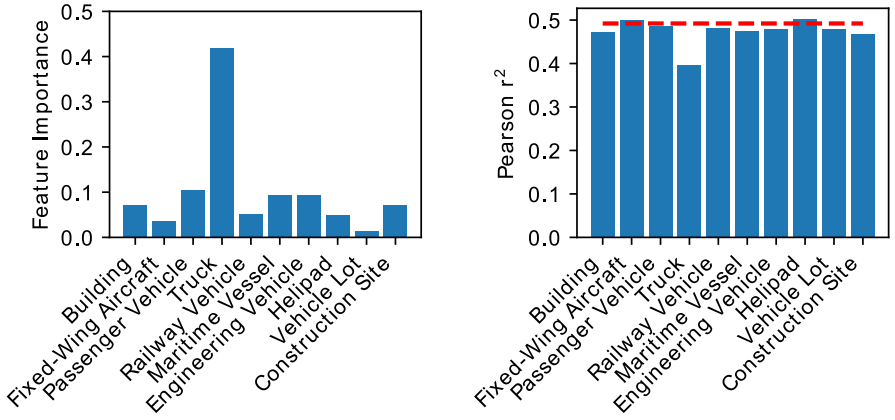


Figure 2.7: **Left**: Feature Importance of parent classes in the GBDT model. **Right**: Ablation analysis where the red line represents the GBDT's performance when including all the parent classes.

To understand how the Truck feature effects the output of the model we plot the SHAP value of *Truck* feature vs. the value of the Truck feature for all the examples in the dataset. Since SHAP values represent a feature's responsibility for a change in the model output, the plot in Figure 2.6 (right) represents the change in predicted poverty score as *Truck* feature changes and also reveals the interaction between *Truck* feature and *Maritime Vessel* feature. We find that for small #*Trucks*, low #*Maritime Vessels* decreases the *Truck* SHAP value. This can be seen from the set of points that form a vertical line (towards bottom left) where the color changes from blue (low #*Maritime Vessels*) to red (high #*Maritime Vessels*) as Truck SHAP value increases.

Feature Importance. We also plot the sum of SHAP value magnitudes over all samples for the various features (feature importance). Figure 2.7 (left) shows the importance of the 10 features (parent level features) in poverty prediction. *Truck* has the highest importance. It is followed by *Passenger Vehicle*, *Maritime Vessel*, and *Engg. Vehicle*.

Ablation Analysis. Finally, we run an ablation study by training the regression model using all the categorical features in the train set and at test time we eliminate a particular feature by collapsing it to zero. We perform this ablation study with the parent level features as it provides better interpretability. Consistent with the feature importance scores, in Figure 2.7 we find that when *Truck* feature is eliminated at test time, the Pearson's r^2 value is impacted most.

2.6 Additional Results

Here we provide additional analysis to understand the effects of various features on the output of the model. Similar to Figure 2.6 (right) we plot (Figure 2.8) the SHAP value of a feature vs. the value of that feature for all the examples in the dataset.

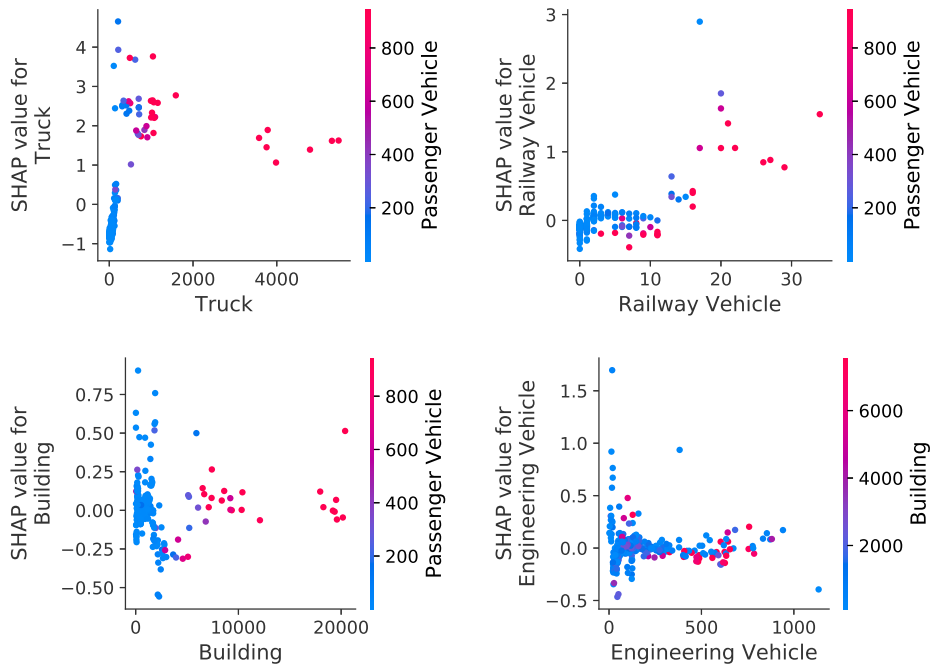


Figure 2.8: Dependence plots showing the effect of a single feature across the whole dataset. In both figures, the color represents the feature value (red is high, blue is low)

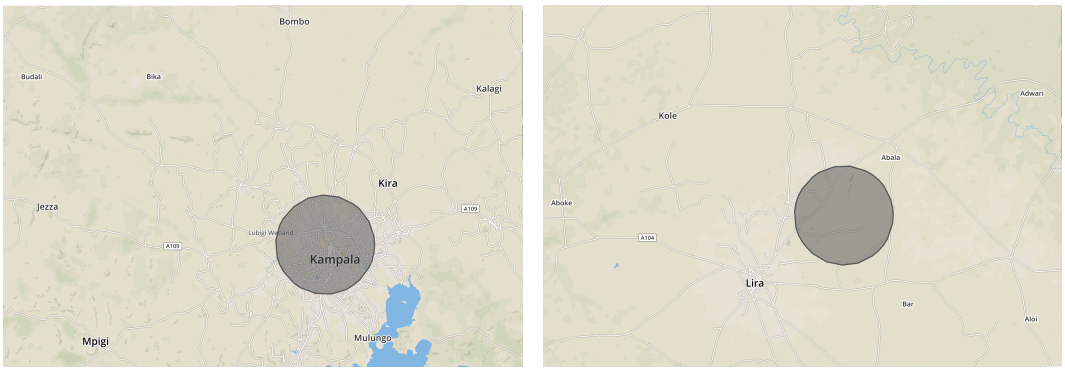


Figure 2.9: **Left**: A region with high wealth level and high truck numbers. **Right**: A region with low wealth level and low truck numbers.

Figure 2.8 represents the change in predicted poverty score as the feature value under consideration changes and also reveals the interaction between that feature and another feature. Top left figure shows that regions with high #*Trucks* also have high #*Passenger Vehicles*. From top right figure, we find that higher #*Railway Vehicles* pushes the output to a higher value and low #*Railway Vehicles* has a negative impact on the output, thereby lowering the predicted value. We also observe that regions with high #*Railway Vehicles* also have high #*Passenger Vehicles*. On the other, we find (bottom left and bottom right figures) that *Buildings*

and *Engineering Vehicles* do not tend to show much impact on the prediction value as their value increases.

Figure 2.9 compares an example region (left) with low poverty level and high #*Trucks* against a region with high poverty level and low #*Trucks*. We find that regions with high truck numbers have better road network and transportation connectivity to nearby regions, thereby resulting in better wealth index in those regions as transportation and connectivity play a vital role in the economic growth of a region.

2.7 Conclusion

In this chapter, we attempt to predict consumption expenditure from high resolution satellite images. We propose an efficient, explainable, and transferable method that combines object detection and regression. This model achieves a Pearson's r^2 of 0.54 in predicting village level consumption expenditure in Uganda, even when the provided locations are affected by noise (for privacy reasons) and the overall number of labels is small ($\sim$300). The presence of trucks appears to be particularly useful for measuring local scale poverty in our setting. We also demonstrate that our features achieve positive results even with simple linear regression models. Our results offer a promising approach for generating interpretable poverty predictions for important livelihood outcomes, even in settings with limited training data.

Chapter 3

Efficient Poverty Estimation from Satellite Images

3.1 Introduction

When combined with machine learning, high-resolution satellite imagery has proven broadly useful for a range of sustainability-related tasks, from poverty prediction [16, 4, 73, 13, 92] to infrastructure measurement [93] to forest and water quality monitoring [94] to the mapping of informal settlements [95]. Compared to coarser (10-30m) publicly-available imagery [96], high-resolution ($< 1m$) imagery has proven particularly useful for these tasks because it is often able to resolve specific objects or features that are undetectable in coarser imagery.

For example, recent work demonstrated an approach for predicting local-level consumption expenditure using object detection on high-resolution daytime satellite imagery [4], showing how this approach can yield interpretable predictions and also outperform previous benchmarks that rely on lower-resolution, publicly-available satellite imagery [96]. 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 [44, 97, 98, 99, 100, 101, 102]. 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 [96] 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 [103, 104].

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 [4] 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.

3.2 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) [4].

[4] 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, they 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.

Problem statement. Let $\{(\mathcal{H}_i, \mathcal{L}_i, y_i, c_i)\}_{i=1}^{N}$ be a set of N villages surveyed, where $c_i = (c_i^{lat}, c_i^{lon})$ 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 (at a cost) and low-resolution (free of charge) 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. Our goal is to learn (1) a regressor f_r to predict the poverty index y_i using $\mathcal{L}_i$ and parts of $\mathcal{H}_i$ (the informative regions) selected by (2) an adaptive data acquisition scheme based on $\mathcal{L}_i$. This adaptive data acquisition scheme is optimized to minimize cost (which depends on the number of selected regions) while maximizing the accuracy of f_r.

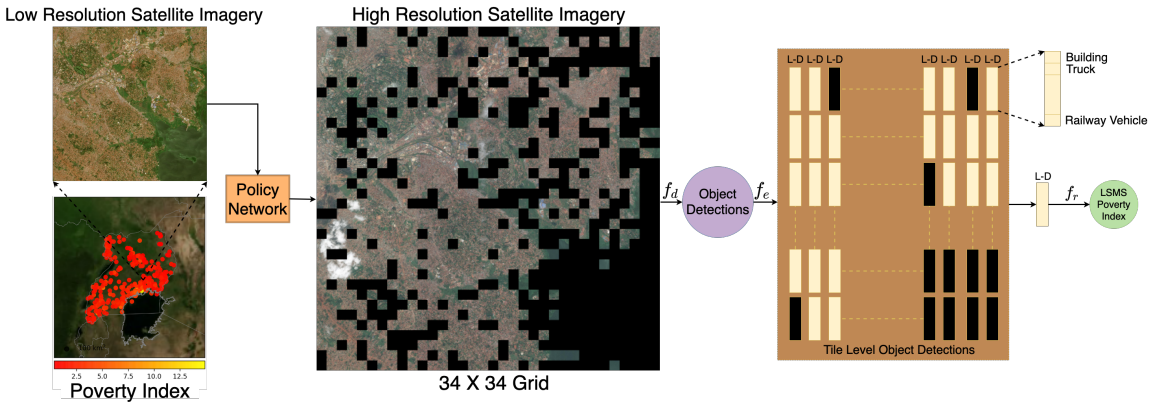


Figure 3.1: Schematic overview of the proposed approach. The Policy Network uses cheaply available Sentinel-2 low-resolution image representing a cluster to output a set of actions representing unique 1000×1000 px high-resolution tiles in the 34×34 grid. Then object detection is performed on the sampled 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 [4] exhaustively uses all the HR tiles representing a cluster for poverty mapping, making their method expensive and less practical.

3.3 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 Uganda Bureau of Statistics between 2011 and 2012 [77]. The survey consists of data from 2,716 households in Uganda, grouped into unique locations called clusters. The latitude and longitude, $c_i = (c_i^{lat}, c_i^{long})$, 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 method performance 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 [4]. Fig. 3.1 (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 acquire all the high-resolution tiles representing a cluster for comparison with [4]. However, in real-word scenario our method requires only a small fraction of HR tiles in test time unlike [4] that acquires HR tiles exhaustively.

We also acquire low-resolution satellite imagery, $\mathcal{L}_i$, corresponding to cluster c_i and represented by a single image of 1014×1014 pixels with 3 channels. These images 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. 3.1 illustrates an example cluster from Uganda.

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

Similar to [4], 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 [4] by training an object detector (YOLOv3 [80]) on xView [76], 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 [4] 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. 3.3).

As described in previous section, 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 [4] 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.

Similar to [4], we use these 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 class-wise object counts can be used in a regression model for poverty estimation [4].

[4] 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. Thus, we reduce the dependency on HR images that are expensive to acquire thereby reducing the costs of poverty prediction models that use HR images exhaustively [4] making their method costly and less practical. We describe our solution in the next section.

3.5 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 [105]. In the first step, we adaptively sample HR tiles and in the second step, we run them through a pre-trained detector.

Task Definition. The first module of our framework finds HR tiles to sample/acquire, conditioned on the low spatial resolution image covering a cluster (which is always acquired). However, a cluster is represented by 34000×34000 px HR images. Directly learning actions with reinforcement learning on such a large area can be very challenging and unstable. For this reason, we decompose our task to many independent sub-tasks where each sub-task focuses on sampling the important parts of the corresponding area with HR images. Following this, we divide a cluster-level HR image $\mathcal{H}_i = (H_i^1, H_i^2, \ldots, H_i^T)$ into equal-size non-overlapping tiles, where T is the number of tiles. Similar to $\mathcal{H}_i$, we decompose $\mathcal{L}_i$ 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. 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 observation $\mathcal{L}_i$. We associate each tile, H_i^j, of $\mathcal{H}_i$ with an L-dimensional classwise object counts feature represented as $\mathbf{v}_i^j$.

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.

For this reason, we divide tile H_i^j into S number of disjoint subtiles as $H_i^j = (h_i^{j,1}, h_i^{j,2}, ..., h_i^{j,S})$. We then define our task as learning a policy network conditioned on l_i^j to only choose HR sub-tiles from H_i^j where there is desirable number of objects characterized by a reward function. Once we learn the policy network, in test time we run it on each l_i^j of a cluster i to sample HR images and run them through detector to find out the cluster-level object counts.

1st Step of MDP. In the first step, 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 $\pi(\mathbf{a}_i^j | l_i^j; \theta_p) = p(\mathbf{a}_i^j | l_i^j; \theta_p)$

where $\pi(l_i^j; \theta_p)$ is a function mapping the observed LR image to a probability distribution over subtile sampling actions $\mathbf{a}_i^j$.

2nd Step of MDP. In the second step, 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:

$$\hat{\mathbf{v}}_i^{j,k} = \begin{cases} f_d(h_i^{j,k}) & \text{if} \quad a_i^{j,k} = 1 \\ \mathbf{0} & \text{else} \end{cases} \tag{3.1}$$

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:

$$\max_{\theta_p} J(\theta_p, \theta_d) = \mathbb{E}_p[R(\mathbf{a}_i^j, \hat{\mathbf{v}}_i^j, \mathbf{v}_i^j)], \tag{3.2}$$

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.

The Reward Function. 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:

$$R = R_{acc}(\hat{\mathbf{v}}_i^j, \mathbf{v}_i^j) + R_{cost}(\mathbf{a}_i^j) \tag{3.3}$$

$$R_{acc} = -||\mathbf{v}_i^j - \hat{\mathbf{v}}_i^j||_1 \tag{3.4}$$

$$R_{cost} = \lambda(1 - ||\mathbf{a}_i^j||_1/S) \tag{3.5}$$

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.

3.6 Modeling and Optimization of 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.

Modeling the Policy Network. In this study, we have $T = 1156$ number of tiles as we have a 34×34 grid of images. In this case, each grid consists of 2000×2000 pixels. As mentioned in the previous section, we divide each tile into S=4 subtiles of 1000×1000 pixels each (higher values of S led to unstable training with higher variance and less sparse selections).

In this study, similar to [105] we model the action likelihood function of the policy network, f_p, using the

product of bernoulli distributions as:

$$\pi(\mathbf{a}_i^j|l_i^j;\theta_p) = \prod_{k=1}^{S}(s_i^{j,k})^{a_i^{j,k}}(1-s_i^{j,k})^{(1-a_i^{j,k})} \tag{3.6}$$

$$s_i^j = f_p(l_i^j;\theta_p) \tag{3.7}$$

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

Optimization of the Policy Network. The previously defined objective function as shown in Eq. 3.2 is not differentiable w.r.t the policy network parameters, θ_p, because acquistion actions are discrete. To overcome this, we train using Policy Gradient [106]. 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.

$$\nabla_{\theta_p} J = \mathbb{E}\left[R(\mathbf{a}_i^j, \hat{\mathbf{v}}_i^j, \mathbf{v}_i^j)\nabla_{\theta_p}\log\pi_{\theta_p}(\mathbf{a}_i^j|l_i^j)\right], \tag{3.8}$$

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 [107], A, to reduce the variance.

$$\nabla_{\theta_p} J = \mathbb{E}\left[A\sum_{k=1}^{S}\nabla_{\theta_p}\log(s_i^{j,k}\mathbf{a}_i^{j,k} + (1-s_i^{j,k})(1-\mathbf{a}_i^{j,k}))\right] \tag{3.9}$$

$$A(\mathbf{a}_i^j, \mathbf{a}_i'^j) = R(\mathbf{a}_i^j, \hat{\mathbf{v}}_i^j, \mathbf{v}_i^j) - R(\mathbf{a}_i'^j, \hat{\mathbf{v}}_i'^j, \mathbf{v}_i^j) \tag{3.10}$$

where $\mathbf{a}_i'^j$ represents the baseline action vector. To get $\mathbf{a}_i'^j$, we use the most likely action vector proposed by the policy network: *i.e.*, $a_i'^{j,k} = 1$ if $s_i^{j,k} > 0.5$ and $a_i'^{j,k} = 0$ otherwise.

Finally, in this study we use temperature scaling [106] to adjust exploration/exploitation trade-off during optimization time as

$$s_i^{j,k} = \alpha s_i^{j,k} + (1-\alpha)(1-s_i^{j,k}). \tag{3.11}$$

Setting α to a large value results in sampling from the learned policy whereas the small values lead to sampling from random policy. We present the pseudocode and implementation details later.

3.7 Experiments

3.7.1 Training and Testing the Policy Network on xView

Our goal is to learn policies to reduce the dependency on HR images in approximating object counts in a geocluster while successfully predicting the downstream index (poverty prediction). Since our downstream

dataset (Uganda) does not contain object bounding boxes, it is not possible to assess how well we approximate true object counts. To achieve this, we train our policy network on the xView dataset where our object detector is trained on. We use 2000×2000 px images and their corresponding 224×224 px LR images to train the policy network on each point. As proposed earlier, the action space has 4 units representing the top left, top right, bottom left, and bottom right part (1000×1000 px) of the full area. The detector is only run on the part chosen by the policy network. We train the policy network on 1249 points and test it on 200 points and show the results in Table 3.1.

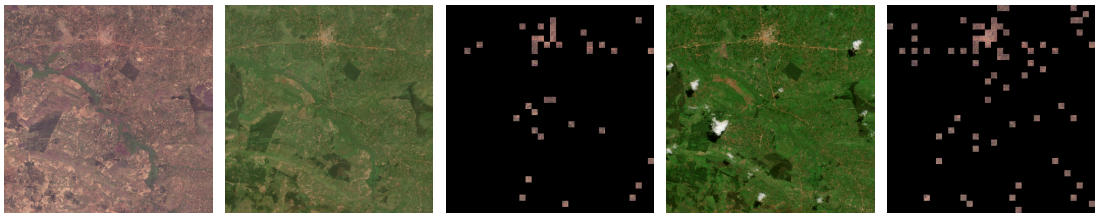
Our policy network uses 42.3% HR images while approximating the fixed approach in mean Average Precision (mAP) and mean Average Recall (mAR) metrics [80]. This results indicate that the policy network learns to successfully choose regions where there are objects of interest and eliminate the regions with no objects of interest.

	mAP	**mAR**	**HR**	**Run-time**
No Dropping	24.3%	42.5%	100.0%	2890 ms
RL Method	26.3%	41.1%	42.3%	1510 ms

Table 3.1: Results on the xView test set.

	No Dropping	**Fixed-18**	**Random-25**	**Stochastic-25**	**Green**	**Counts Pred.**	**Sett. Layer**	**Nightlights**	**Ours (Dry sea.)**	**Ours (Wet sea.)**
r^2	0.53	0.43	0.34	0.26	0.33	0.49	0.45	0.45	0.51 ± 0.01	$\mathbf{0.61 \pm 0.01}$
MSE	1.86	2.20	2.67	3.13	2.56	1.91	2.16	2.17	1.89 ± 0.02	$\mathbf{1.46 \pm 0.02}$
Explained Variance	0.54	0.43	0.33	0.27	0.36	0.48	0.46	0.45	0.50 ± 0.01	$\mathbf{0.63 \pm 0.02}$
HR Acquisition.	1.0	0.18	0.25	0.25	0.19	0.19	0.19	0.12	0.19	**0.19**

Table 3.2: LSMS poverty score prediction results in Pearson's r^2 (and two other metrics) for various methods. *HR Acquisition* represents the fraction of HR tiles acquired. We report the mean and std of our RL model across 7 runs with different seeds.



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

Figure 3.2: (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.

3.7.2 Testing the Policy Network on Poverty Prediction

Previously, we trained and tested the policy network to quantify how well we approximate the true object counts. In this section, we train and test the policy network on Uganda dataset where we have only cluster-level poverty labels.

Poverty Estimation. Previous work [4] 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 [4], 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), $\mathbf{m}_i$ or $\hat{\mathbf{m}}_i$.

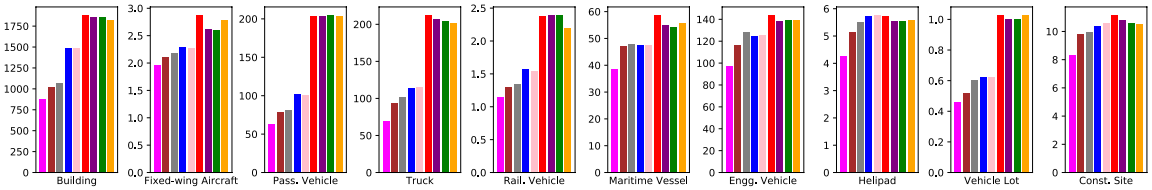
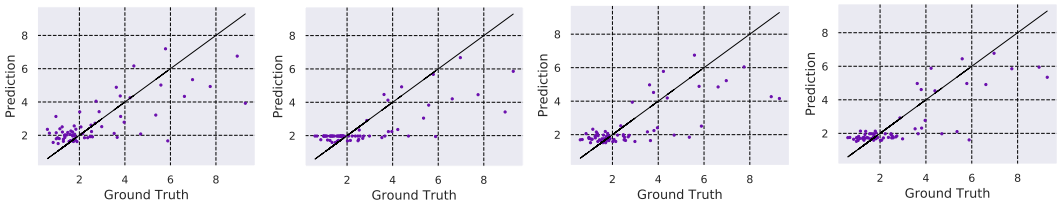
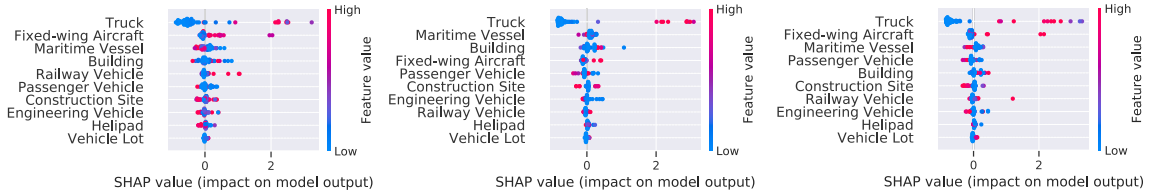


Figure 3.3: Number of objects missed on average across clusters for each class. Colored bars in each subplot from left-right are: Ours (wet season), Ours (dry season), Counts Pred., Nightlight, Settlement, Fixed-18, Random-25, Green Tiles, Stochastic-25.



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

Figure 3.4: LSMS poverty score regression results of GBDT.



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

Figure 3.5: 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).

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 training set. We use the clusters in the training set 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 a later section for more details. To evaluate the models, 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. We also report mean squared error (MSE) and Explained Variance [108]. Explained variance measures the discrepancy between a model and actual data. Higher explained variance indicates a stronger strength of association thus meaning better predictions.

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 (same as [4]), (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, (f) *Green Tiles*, where we compute the average green channel value for a low-res tile and select bottom K tiles for HR acquisition with least average green channel value, where K is the number of tiles selected by the policy network for a particular cluster, (g) *Counts Prediction*, where we train a CNN (Resnet-50 backbone) to regress object counts given low-res tile as input. We find that the object counts in a tile vary from 0-500. Instead of regressing directly on raw object counts, we create 100 bins such that a tile with counts between 5i-(5i+1) has label 5i+2.5 (e.g. a tile with counts 0-5 has label 2.5, 5-10 has label 7.5 and so on). We use this network to select top K HR tiles based on predicted object counts, (h) *Settlement Layer*, where we select HR tiles based on their population density. We used the HR settlement layer maps[1] and selected top K tiles based on population density, 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 rural 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. 3.3 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

[1] https://research.fb.com/downloads/high-resolution-settlement-layer-hrsl/

we can see that our method (using wet season imagery) can better approximate the "true object counts" (we use object detector predictions on all the HR tiles as a proxy for true values) compared to baselines and our method (using dry season imagery). Table 3.2 shows the results of poverty prediction in Uganda. Our model (wet season) achieves **0.61** r^2 and substantially outperforms the published state-of-the-art results [4] (**0.53** r^2) while using around **80%** fewer HR images.

It is interesting that we can outperform *No Dropping* method when sampling only 20% of HR tiles. Qualitatively, we observed that it is due to false positives proposed by the object detector on the tiles with no true objects of interest in it. Unfortunately, since we do not have ground truth bounding boxes for Uganda, we can not quantify it. However, our experiments on xView (Table 3.1) show that our approach achieves higher AP than the *No Dropping* approach, suggesting our approach is able to remove false positives.

In comparison to the baselines relying on external data layers such as settlement and nightlights, our method achieves around 0.16 higher r^2. This is because such maps assume that objects are located in the tiles with large nightlight intensity or settlement index, however, some objects, i.e trucks, passenger vehicles etc., do not necessarily exist in these areas. Additionally, our approach outperforms the counts prediction model by 0.12 in r^2. This might be because the counts predictor is trained to directly regress very noisy object counts thus making it a difficult task.

Next, a scatter plot of GBDT LSMS poverty score predictions v.s. ground truth is shown in Fig. 3.4. 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 [4] that exhaustively uses all HR tiles.

Performance/Sampling Trade-off. We analyze the trade-off between accuracy (regression performance) and HR sampling rate controlled by the hyperparameter λ in the reward Eq. 3.5. We intentionally change λ to quantify the effect on the policy network. As seen in Fig. 3.6, the policy network samples less HR tiles (a 0.09 fraction) 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 terms of r^2, while acquiring only a 0.18 fraction of HR imagery.

Cost saving. Current pricing for high-resolution (30cm) RGB imagery is 10-20$ per km^2. Given that

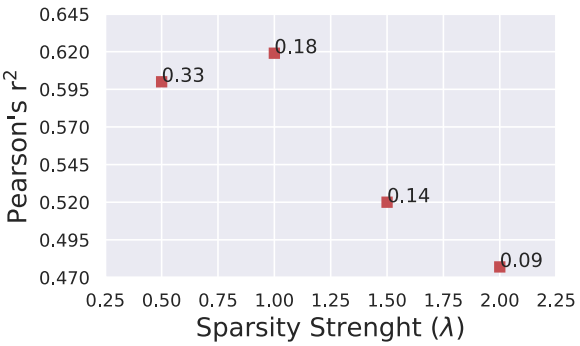


Figure 3.6: Trade-off between Pearson's r^2 and coefficent of image acquisition cost (λ). Text accompanying the points represents HR acquisition fraction.

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. This represents a potentially large cost saving if our approach is scaled at country or continent scale.

Analysis based on Season. 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. Fig. 3.7 presents an example cluster, where it is seen that training the policy network using wet season imagery better assists the network at sampling informative tiles.

Impact on Interpretability. An important contribution of [4] was to introduce model interpretability allowing successful application of such methods in many policy domains. They use Tree SHAP (Tree SHapley Additive exPlanations) [90], a game theoretic approach to explain the output of tree-based models, to explain the effect of individual features on poverty predictions. Here, we show that in addition to closely approximating the classwise object counts, our method retains the same findings for interpretability as that of [4]. Fig. 3.5 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 ordering of features in terms of SHAP values is fairly similar between the *No Dropping* approach [4] and our method.

3.8 Pseudocode

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

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

3.9 Implementation Details

Policy Network. To parameterize the policy network, we use ResNet-32 [87] pretrained on the ImageNet dataset [109]. We train the policy network using 2 NVIDIA 1080ti GPUs and use the following hyperparameters: learning rate = 1e-4, #epochs = 300, batch size = 289, coefficient of image acquisition cost *i.e.* λ=1.0, temperature scaling α is gradually increased from 0.6 to 0.95.

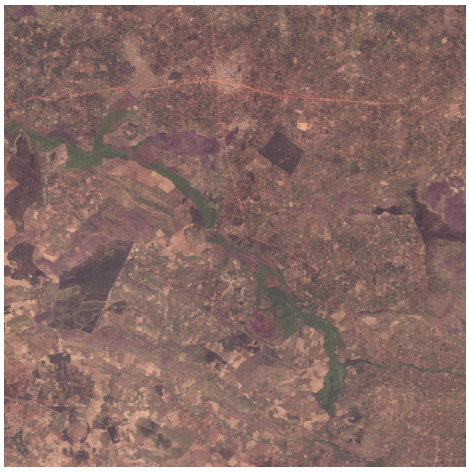
Object Detector. We use YOLOv3 architecture [80] as the object detector, chosen for its reasonable trade off between accuracy on small objects and run-time performance. The backbone network, DarkNet-53, is pretrained on ImageNet. Following [4], we perform transfer learning by training the detector on xView dataset and running it on the Uganda HR patches. The object detections are obtained at 0.6 confidence threshold following [4].

3.10 Additional Qualitative Results

In this section, we present additional qualitative results. Figures 3.7 and 3.8 emphasize that the presence of greenery during wet season allows the policy network to better identify the informative regions containing objects (better sampling of regions containing buildings, trucks, vehicles, etc.), compared to when dry season Sentinel-2 imagery is used as input to the network. Figures 3.9, 3.10, and 3.11 show additional results when wet season imagery is used as input to the policy network.

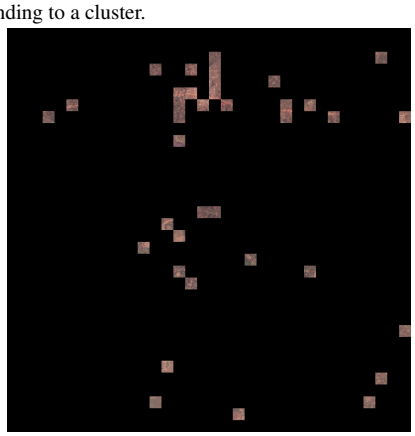
3.11 Conclusion

In this chapter, 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.

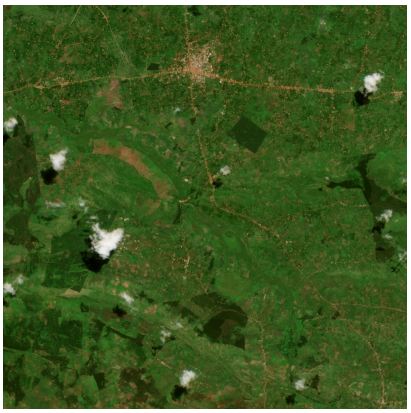


(a) High-Resolution Satellite Imagery (downsampled for visualization) corresponding to a cluster.

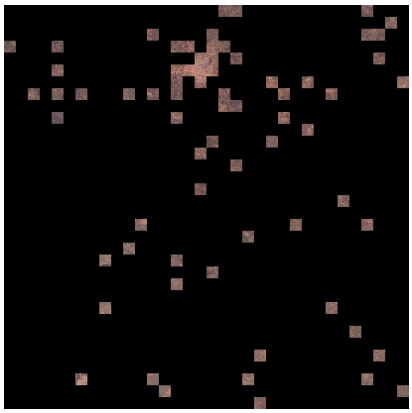
(b) Low Resolution Sentinel-2 Imagery for the cluster from dry season.



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

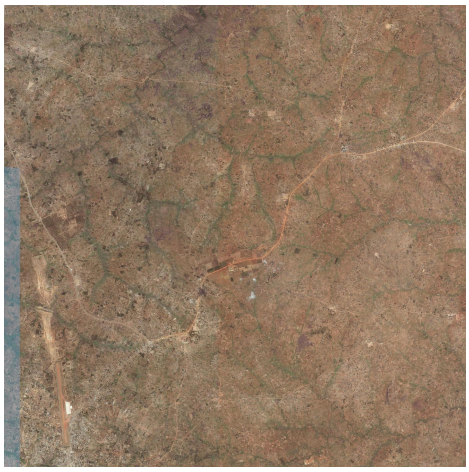


(d) Low Resolution Sentinel-2 Imagery for the cluster from wet season.

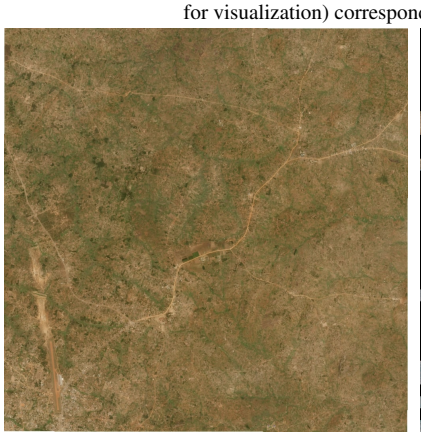


(e) Corresponding HR acquisitions when wet-season imagery is input to the Policy Network.

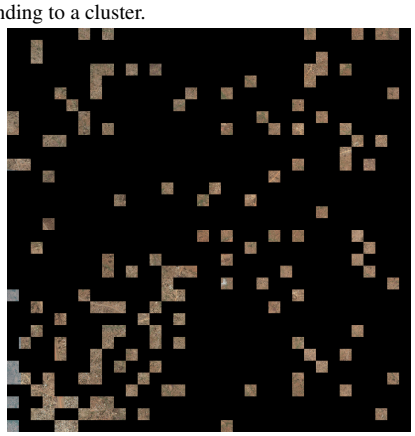
Figure 3.7: Comparison between sampling ability of the policy network when trained with low-resolution imagery from two different seasons.



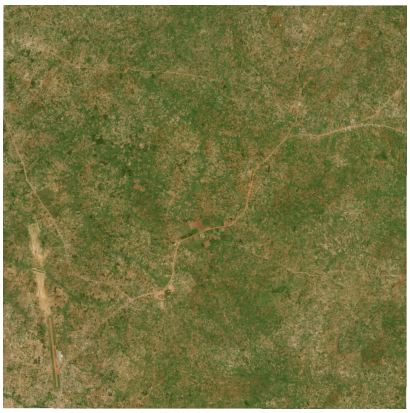
(a) High-Resolution Satellite Imagery (downsampled for visualization) corresponding to a cluster.



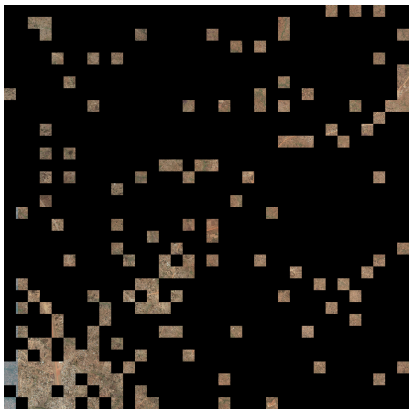
(b) Low Resolution Sentinel-2 Imagery for the cluster from dry season.



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



(d) Low Resolution Sentinel-2 Imagery for the cluster from wet season.

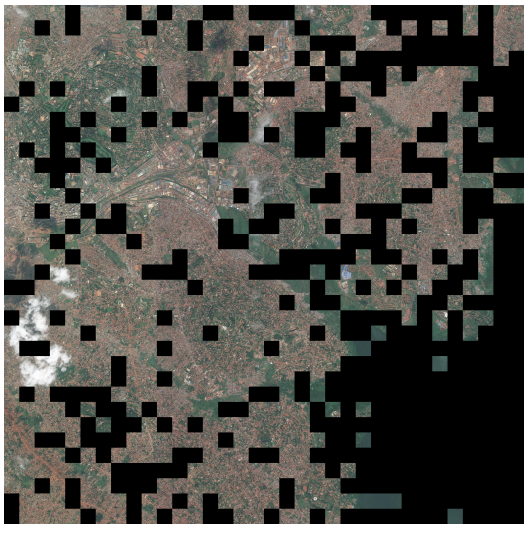


(e) Corresponding HR acquisitions when wet-season imagery is input to the Policy Network.

Figure 3.8: Comparison between sampling ability of the policy network when trained with low-resolution imagery from two different seasons.

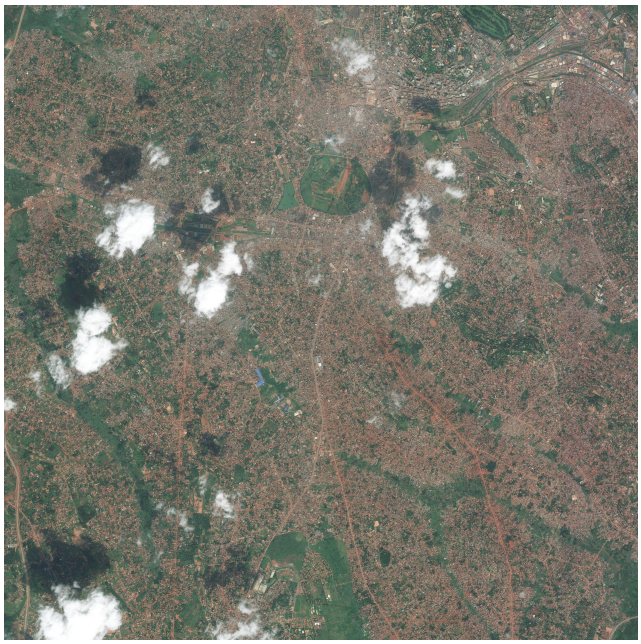
(a) High-Resolution Satellite Imagery (from Digital Globe Systems) corresponding to a cluster.

(b) Low-Resolution Sentinel-2 Satellite Imagery corresponding to the same cluster.



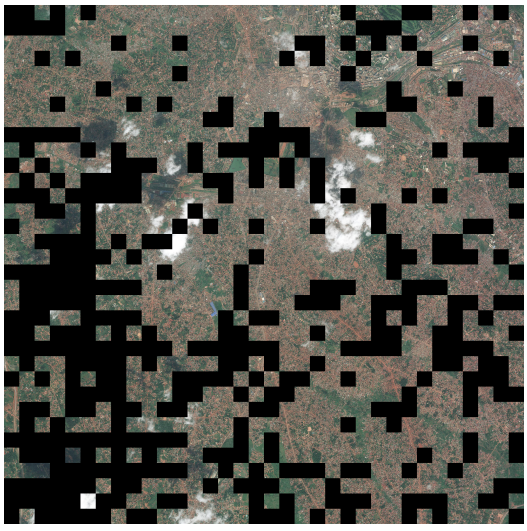
(c) Sampled Regions for HR acquisition which are subsequently used for Poverty Prediction.

Figure 3.9: Additional results when wet season imagery is used as input to the policy network.



(a) High-Resolution Satellite Imagery (from Digital Globe Systems) corresponding to a cluster.

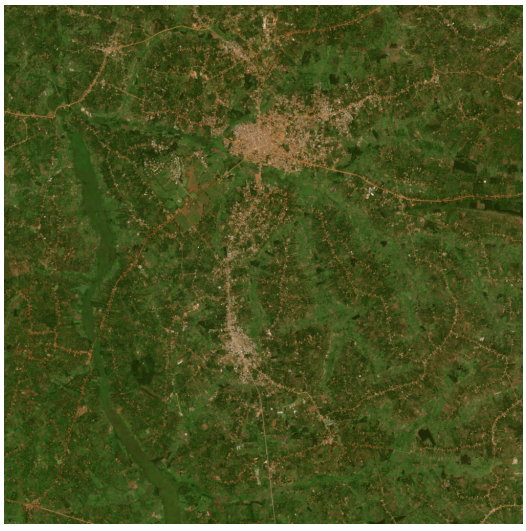
(b) Low-Resolution Sentinel-2 Satellite Imagery corresponding to the same cluster.



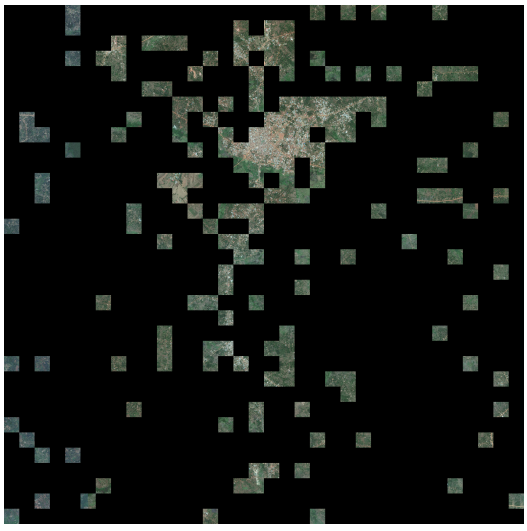
(c) Sampled Regions for HR acquisition which are subsequently used for Poverty Prediction.

Figure 3.10: Additional results when wet season imagery is used as input to the policy network.

(a) High-Resolution Satellite Imagery (from Digital Globe Systems) corresponding to a cluster.



(b) Low-Resolution Sentinel-2 Satellite Imagery corresponding to the same cluster.



(c) Sampled Regions for HR acquisition which are subsequently used for Poverty Prediction.

Figure 3.11: Additional results when wet season imagery is used as input to the policy network.

Chapter 4

Geography-Aware Self-Supervised Learning

4.1 Introduction

Inspired by the success of self-supervised learning methods [5, 2], we explore their application to large-scale remote sensing datasets (satellite images) and geo-tagged natural image datasets. It has been recently shown that self-supervised learning methods perform comparably well or even better than their supervised learning counterpart on image classification, object detection, and semantic segmentation on traditional computer vision datasets [110, 111, 2, 5, 33]. However, their application to remote sensing images is largely unexplored, despite the fact that collecting and labeling remote sensing images is particularly costly as annotations often require domain expertise [44, 76, 112].

In this direction, we first experimentally evaluate the performance of an existing self-supervised contrastive learning method, MoCo-v2 [2], on remote sensing datasets, finding a performance gap with supervised learning using labels. For instance, on the Functional Map of the World (fMoW) image classification benchmark [112], we observe an 8% gap in top 1 accuracy between supervised and self-supervised methods.

To bridge this gap, we propose geography-aware contrastive learning to leverage the spatio-temporal structure of remote sensing data. In contrast to typical computer vision images, remote sensing data are often geo-located and might provide multiple images of the same location over time. Contrastive methods encourage closeness of representations of images that are likely to be semantically similar (positive pairs). Unlike contrastive learning for traditional computer vision images where different views (augmentations) of the same image serve as a positive pair, we propose to use *temporal positive pairs* from spatially aligned images over time. Utilizing such information allows the representations to be invariant to subtle variations over time (e.g., due to seasonality). This can in turn result in more discriminative features for tasks focusing on spatial variation, such as object detection or semantic segmentation (but not necessarily for tasks involving

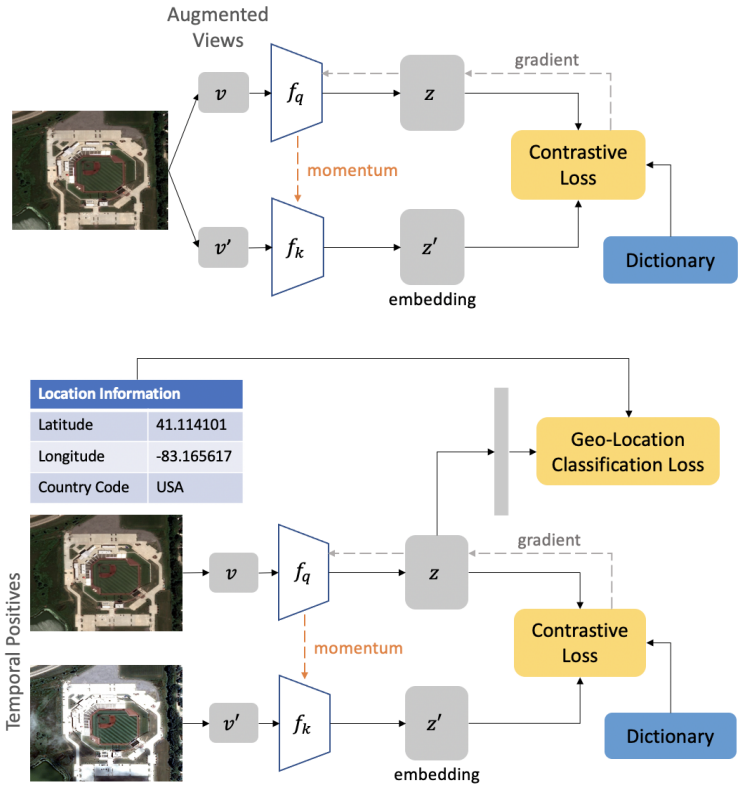


Figure 4.1: **Top** shows the original MoCo-v2 [5] framework. **Bottom** shows the schematic overview of our approach.

temporal variation such as change detection). In addition, we design a novel unsupervised learning method that leverages geo-location information, i.e., knowledge about where the images were taken. More specifically, we consider the pretext task of predicting where in the world an image comes from, similar to [51, 52]. This can complement the information-theoretic objectives typically used by self-supervised learning methods by encouraging representations that reflect geographical information, which is often useful in remote sensing tasks. Finally, we integrate the two proposed methods into a single geography-aware contrastive learning objective.

Our experiments on the functional Map of the World [112] dataset consisting of high spatial resolution satellite images show that we improve MoCo-v2 baseline significantly. In particular, we improve it by $\sim 8\%$ classification accuracy when testing the learned representations on image classification, $\sim 2\%$ AP on object detection, $\sim 1\%$ mIoU on semantic segmentation, and $\sim 3\%$ top-1 accuracy on land cover classification. Interestingly, our geography-aware learning can even outperform the supervised learning counterpart on temporal data classification by $\sim 2\%$. To further demonstrate the effectiveness of our geography-aware learning approach, we extract the geo-location information of ImageNet images using FLICKR API similar to [113], which provides us with a subset of 543,435 geo-tagged ImageNet images. We extend the proposed

approaches to geo-located ImageNet, and show that geography-aware learning can improve the performance of MoCo-v2 by $\sim 2\%$ on image classification, showing the potential application of our approach to any geo-tagged dataset. Figure 4.1 shows our contributions in detail.

4.2 Related Work

Self-supervised methods use unlabeled data to learn representations that are transferable to downstream tasks (image classification, object detection, semantic segmentation). Two commonly seen self-supervised methods are *pre-text task* and *contrastive learning*.

Pre-text tasks Pre-text task based learning [24, 25, 26, 27, 28, 29] can be used to learn feature representations when data labels are not available. [30] rotates an image and then trains a model to predict the rotation angle. [31] trains a network to perform colorization of a grayscale image. [32] represents an image as a grid, permuting the grid and then predicting the permutation index. In this study, we use *geo-location classification* as a pre-text task, in which a deep network is trained to predict a coarse geo-location of where in the world the image might come from.

Contrastive Learning Recent self-supervised contrastive learning approaches such as MoCo [2], MoCo-v2 [5], SimCLR [33], PIRL [24], and FixMatch [34] have demonstrated superior performance and have emerged as the fore-runner on various downstream tasks. The intuition behind these methods are to learn representations by pulling positive image pairs from the same instance closer in latent space while pushing negative pairs from difference instances further away. These methods, on the other hand, differ in the type of contrastive loss, generation of positive and negative pairs, and sampling method.

"gsd":	2.10264849663	2.06074237823	1.9968634	2.2158575	1.24525177479	1.4581833	1.2518295
"img_width":	2421	2410	2498	2253	4016	3400	4003
"img_height":	2165	2156	2235	2015	3592	3041	3581
"country_code":	IND	IND	IND	IND	IND	IND	IND
"cloud_cover":	6	0	1	0	0	2	0
“timestamp":	2015-11-02 T05:44:14Z	2016-03-09 T05:25:30Z	2017-02-02 T05:47:02Z	2017-02-27 T05:24:30Z	2015-04-09 T05:36:04Z	2016-12-28 T05:57:06Z	2017-04-12 T05:51:49Z

Figure 4.2: Images over time concept in the fMoW dataset. The metadata associated with each image is shown underneath. We can see changes in contrast, brightness, cloud cover etc. in the images. These changes render spatially aligned images over time useful for constructing additional positives.

Although growing rapidly in self-supervised learning area, contrastive learning methods have not been explored on large-scale remote sensing dataset. In this work, we provide a principled and effective approach for improving representation learning using MoCo-v2 [2] for remote sensing data as well geo-located conventional datasets.

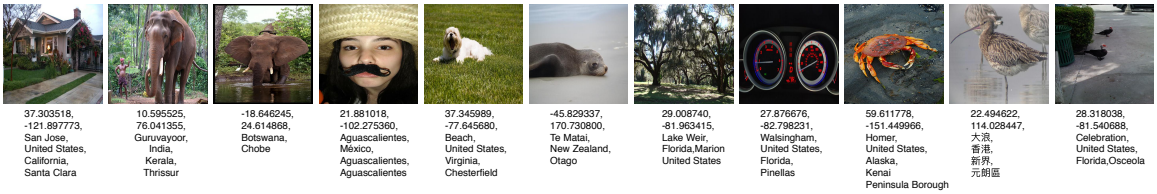


Figure 4.3: Some examples from GeoImageNet dataset. Below each image, we list their latitudes, longitudes, city, country name. In our study, we use the latitude and longitude information for unsupervised learning. We recommend readers to zoom-in to visualize the details of the pictures.

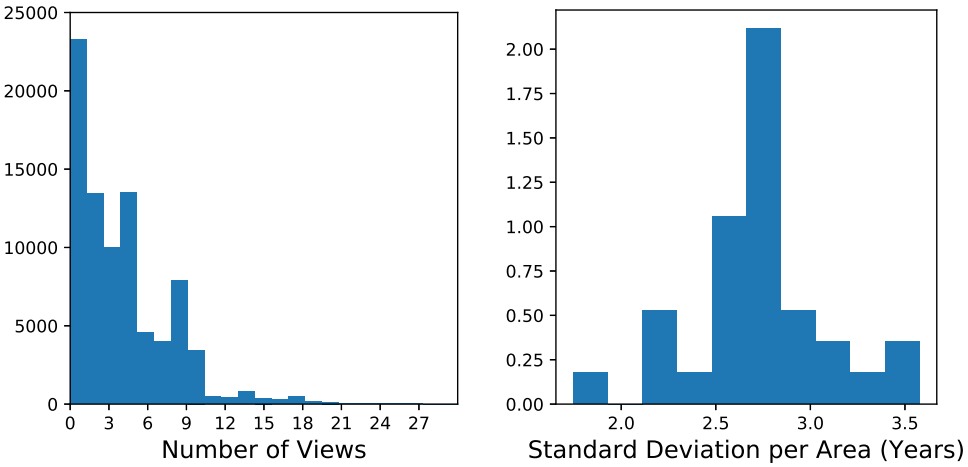


Figure 4.4: **Left** The histogram of number of views. **Right** the histogram of standard deviation in years per area in fMoW.

Unsupervised Learning in Remote Sensing Images Unlike in traditional computer vision areas, unsupervised learning on remote sensing domain has not been studied comprehensively. Most of the studies utilize small-scale datasets specific to a small geographical region [35, 36, 37, 38, 39], a few classes [40] or a highly-specific modality, i.e. hyperspectral images [41, 42]. Most of these studies focus on the UCM-21 dataset [43] consisting of less than 1,000 images from 21 classes. A more recent study [44] proposes large-scale weakly supervised learning using a multi-modal dataset consisting of satellite images and paired geo-located wikipedia articles. While being effective, this method requires each satellite image to be paired to its corresponding article, limiting the number of images that can be used.

Geography-aware Computer Vision Geo-location data has been studied extensively in prior works. Most of these studies utilizes geo-location of an image as a prior to improve image recognition accuracy [45, 46, 47, 48, 49]. Other studies [50, 51, 52, 53] use geo-tagged training datasets to learn how to predict the geo-location of previously unseen images at test time. In our study, we leverage geo-tag information to improve unsupervised and self-supervised learning methods.

4.3 Problem Definition

We consider a geo-tagged visual dataset $\{((x_i^1, \cdots, x_i^{T_i}), \text{lat}_i, \text{lon}_i)\}_{i=1}^N$, where the ith datapoint consists of a sequence of images $\mathcal{X}_i = (x_i^1, \cdots, x_i^{T_i})$ at a shared location, with latitude and longitude equal to $\text{lat}_i, \text{lon}_i$ respectively, over time $t_i = 1, ..., T_i$. When $T_i > 1$, we refer to the dataset to have temporal information or structure. Although temporal information is often not available in natural image datasets (ImageNet), it is common in remote sensing. While the temporal structure is similar to that of conventional videos, there are some key differences that we exploit in this work. First, we consider relatively short temporal sequences, where the time difference between two consecutive "frames" could range from months to years. Additionally unlike conventional videos we consider datasets where there is no viewpoint change across the image sequence.

Given our setup, we want to obtain visual representations $z_i^{t_i}$ of images $x_i^{t_i}$ such that the learned representation can be transferred to various downstream tasks. We do not assume access to any labels or human supervision beyond the $\text{lat}_i, \text{lon}_i$ geo-tags. The quality of the representations is measured by their performance on various downstream tasks. Our primary goal is to improve the performance of self-supervised learning by utilizing the geo-coordinates and the unique temporal structure of remote sensing data.

4.3.1 Functional Map of the World

Functional Map of the World (fMoW) is a large-scale publicly available remote sensing dataset [112] consisting of approximately 363,571 training images and 53,041 test images across 62 highly granular class categories. It provides images (temporal views) from the same location over time $(x_i^1, \cdots, x_i^{T_i})$ as well as geo-location metadata $(\text{lat}_i, \text{lon}_i)$ for each image. Fig. 4.4 shows the histogram of the number of temporal views in fMoW dataset. We can see that most of the areas have multiple temporal views where T_i can range from 1 to 21, and on average there is about 2.5-3 years of difference between the images from an area. Also, we show examples of spatially aligned images in Fig. 4.2. As seen in Fig. 4.5, fMoW is a global dataset consisting of images from seven continents which can be ideal for learning global remote sensing representations.

4.3.2 GeoImageNet

Following [113], we extract geo-coordinates for a subset of images in ImageNet [78] using the FLICKR API. More specifically, we searched for geo-tagged images in ImageNet using the FLICKR API, and were able to find 543,435 images with their associated coordinates $(\text{lat}_i, \text{lon}_i)$ across 5150 class categories. This dataset is more challenging than ImageNet-1k as it is highly imbalanced and contains about 5$\times$ more classes. In the rest of the paper, we refer to this geo-tagged subset of ImageNet as *GeoImageNet*. Upon publication, we will release the GeoImageNet dataset publicly for the research community.

We show some examples from GeoImageNet in Fig. 4.3. As shown in the figure, for some images we have geo-coordinates that can be predicted from visual cues. For example, we see that a picture of a person

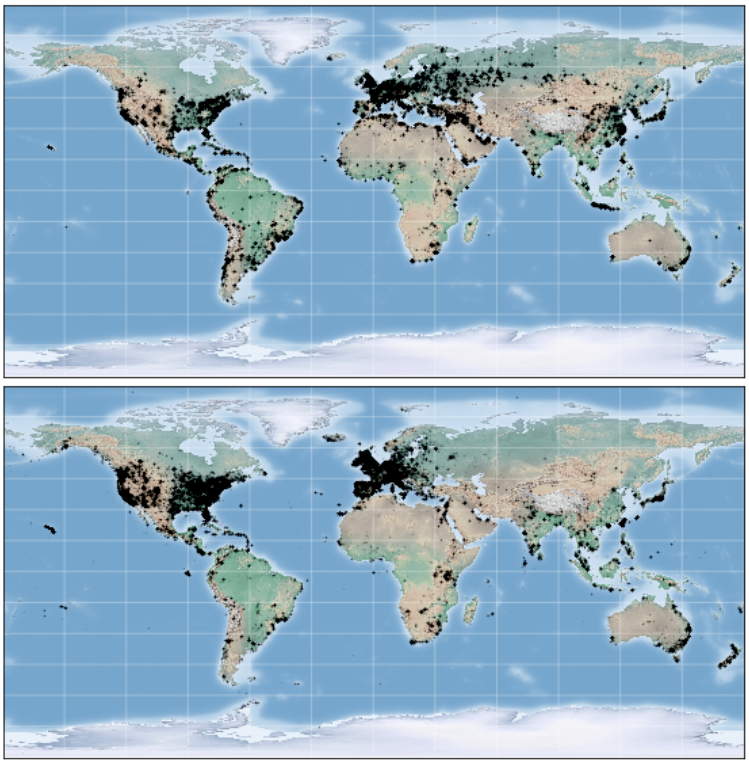


Figure 4.5: **Top** shows the distribution of the fMoW and **Bottom** shows the distribution of GeoImageNet.

with a Sombrero hat was captured in Mexico. Similarly, an Indian Elephant picture was captured in India, where there is a large population of Indian Elephants. Next to it, we show the picture of an African Elephant (which is larger in size). If a model is trained to predict where in the world the image was taken, it should be able to identify visual cues that are transferable to other tasks (e.g., visual cues to differentiate Indian Elephants from the African counterparts). Figure 4.5 shows the distribution of images in the GeoImageNet dataset.

4.4 Method

In this section, we briefly review contrastive loss functions for unsupervised learning and detail our proposed approach to improve Moco-v2 [5], a recent contrastive learning framework, on geo-located data.

4.4.1 Contrastive Learning Framework

Contrastive [2, 33, 5, 114, 59] methods attempt to learn a mapping $f_q : x_i^t \mapsto z_i^t \in \mathbb{R}^d$ from raw pixels x_i^t to semantically meaningful representations z_i^t in an unsupervised way. The training objective encourages representations corresponding to pairs of images that are known a priori to be semantically similar (positive

pairs) to be closer to each other than typical unrelated pairs (negative pairs). With similarity measured by dot product, recent approaches in contrastive learning differ in the type of contrastive loss and generation of positive and negative pairs. In this work, we focus on the state-of-the-art contrastive learning framework MoCo-v2 [5], an improved version of MoCo [2], and study improved methods for the construction of positive and negative pairs tailored to remote sensing applications.

The contrastive loss function used in the MoCo-v2 framework is InfoNCE [59], which is defined as follows for a given data sample:

$$L_z = -\log \frac{\exp(z \cdot \hat{z}/\lambda)}{\exp(z \cdot \hat{z}/\lambda) + \sum_{j=1}^{N} \exp(z \cdot k_j/\lambda)}, \tag{4.1}$$

where z and $\hat{z}$ are query and key representations obtained by passing the two augmented views of x_i^t (denoted v and v' in Fig. 4.1) through query and key encoders, f_q and f_k parameterized by θ_q and θ_k respectively. Here z and $\hat{z}$ form a positive pair. The N negative samples, $\{k_j\}_{j=1}^{N}$, come from a dictionary of representations built as a queue. We refer readers to [2] for details on this. $\lambda \in \mathbb{R}^+$ is the temperature hyperparameter.

The key idea here is to encourage representations of positive (semantically similar) pairs to be closer, and negative (semantically unrelated) pairs to be far apart as measured by dot product. The construction of positive and negative pairs plays a crucial role in this contrastive learning framework. MoCo and MoCo-v2 both use perturbations (also called “data augmentation") from the same image to create a positive example and perturbations from different images to create a negative example. Commonly used perturbations include random color jittering, random horizontal flip, and random grayscale conversion.

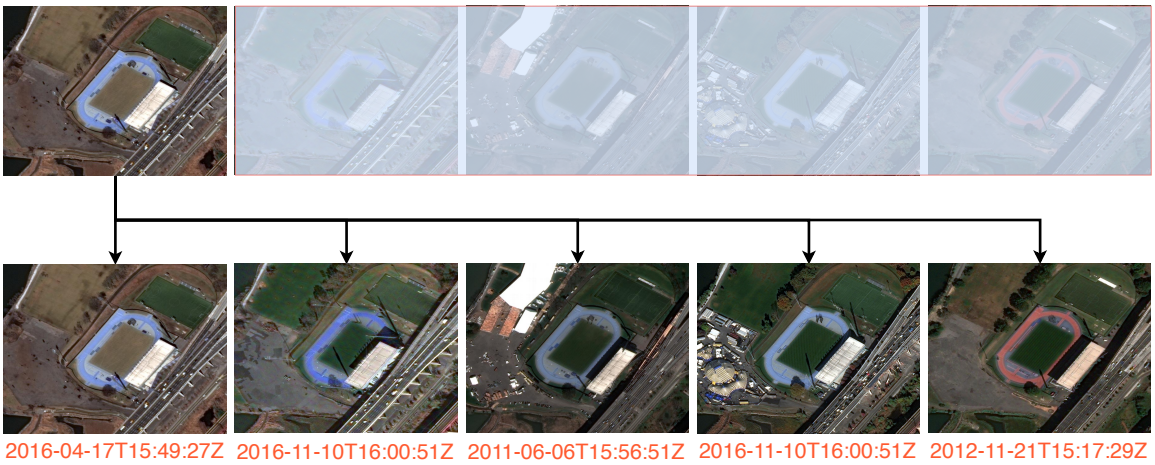


Figure 4.6: Demonstration of temporal positives in eq. 4.2. An image from an area is paired to the other images including itself from the same area captured at different time. We show the time stamps for each image underneath the images. We can see the color changes in the stadium seatings and surrounding areas.

Temporal Positive Pairs Different from many commonly seen natural image datasets, remote sensing datasets often have extra temporal information, meaning that for a given location $(\text{lat}_i, \text{lon}_i)$, there exists a sequence

of spatially aligned images $\mathcal{X}_i = (x_i^1, \cdots, x_i^{T_i})$ over time. Unlike in traditional videos where nearby frames could experience large changes in content (from a cat to a tree), in remote sensing the content is often more stable across time due to the fixed viewpoint. For instance, a place on ocean is likely to remain as ocean for months or years, in which case satellite images taken across time at the same location should share high semantic similarities. Even for locations where non-trivial changes do occur over time, certain semantic similarities could still remain. For instance, key features of a construction site are likely to remain the same even as the appearance changes due to seasonality.

Given these observations, it is natural to leverage temporal information for remote sensing while constructing positive or negative pairs since it can provide us with extra semantically meaningful information of a place over time. More specifically, given an image $x_i^{t_1}$ collected at time t_1, we can randomly select another image $x_i^{t_2}$ that is spatially aligned with $x_i^{t_1}$ ($x_i^{t_2} \in \mathcal{X}_i$). We then apply perturbations (random color jittering) as used in MoCo-v2 to the spatially aligned image pair $x_i^{t_1}$ and $x_i^{t_2}$, providing us with a *temporal positive pair* (denoted v and v' in Figure 4.1) that can be used for training the contrastive learning framework by passing them through query and key encoders, f_q and f_k respectively (see Fig. 4.1). Note that when $t_1 = t_2$, the *temporal positive pair* is the same as the positive pair used in MoCo-v2.

Given a data sample $x_i^{t_1}$, our TemporalInfoNCE objective function can be formulated as follows:

$$L_{z_i^{t_1}} = -\log \frac{\exp(z_i^{t_1} \cdot z_i^{t_2}/\lambda)}{\exp(z_i^{t_1} \cdot z_i^{t_2}/\lambda) + \sum_{j=1}^{N} \exp(z_i^{t_1} \cdot k_j/\lambda)}, \tag{4.2}$$

where $z_i^{t_1}$ and $z_i^{t_2}$ are the encoded representations of the randomly perturbed temporal positive pair $x_i^{t_1}, x_i^{t_2}$.

Similar as equation 4.1, N is number of negative samples, $\{k_j\}_{j=1}^N$ are the encoded negative pairs and $\lambda \in \mathbb{R}^+$ is the temperature hyperparameter. Again, we refer readers to [2] for details on construction of these negative pairs.

Note that compared to equation 4.1, we use two *real* images from the same area over time to create positive pairs. We believe that relying on real images for positive pairs encourages the network to learn better representations for real data than the one relying on synthetic images. On the other hand, our objective in equation 4.2 enforces the representations to be invariant to changes over time. Depending on the target task, such inductive bias can be desirable or undesirable. For example, for a change detection task, learning representations that are highly sensitive to temporal changes can be more preferable. However, for image classification or object detection task, learning temporally invariant features should not degrade the downstream performance.

4.4.2 Geo-location Classification as a Pre-text Task

In this section, we explore another aspect of remote sensing images, *geolocation metadata*, to further improve the quality of the learned representations. In this direction, we design a pre-text task for unsupervised learning. In our pre-text task, we cluster the images in the dataset using their coordinates $(\text{lat}_i, \text{lon}_i)$. We

use a clustering method to construct K clusters and assign an area with coordinates $(\text{lat}_i, \text{lon}_i)$ a categorical geo-label $c_i \in \mathcal{C} = \{1, \cdots, K\}$. Using the cross entropy loss function, we then optimize a geo-location predictor network f_c as

$$L_g = \sum_{k=1}^{K} -p(c_i = k) \log(\hat{p}(c_i = k | f_c(x_i^t)), \tag{4.3}$$

where p represent the probability of the cluster k representing the true cluster and $\hat{p}$ represents the predicted probabilities for K clusters. In our experiments, we represent f_c with a CNN parameterized by θ_c. With this objective, our goal is to learn location-aware representations that can potentially transfer well to different downstream tasks.

4.4.3 Combining Geo-location and Contrastive Learning Losses

In the previous section, we designed a pre-text task leveraging the geo-location meta-data of the images to learn location-aware representations in a standalone setting. In this section, we combine geo-location prediction and contrastive learning tasks in a single objective to improve the contrastive learning-only and geo-location learning-only tasks. In this direction, we first integrate the geo-location learning task into the contrastive learning framework using the cross-entropy loss function where the geo-location prediction network uses features z_i^t from the query encoder as

$$L_g = -\sum_{i=1}^{K} p(c_i = k) \log(\hat{p}(c_i = k | f_c(z_i^t)). \tag{4.4}$$

In the combined framework (see Fig. 4.1), f_c is represented by a linear layer parameterized by θ_c. Finally, we propose the objective for joint learning as the linear combination of TemporalInfoNCE and geo-classification loss with coefficients α and β representing the importance of contrastive learning and geo-location learning losses as

$$\underset{\theta_q, \theta_k, \theta_c}{\operatorname{argmin}} L_f = \alpha L_{z^{t_1}} + \beta L_g. \tag{4.5}$$

By combining two tasks, we learn representations to jointly maximize agreement between spatio-temporal positive pairs, minimize agreement between negative pairs and predict the geo-label of the images from the positive pairs.

4.5 Experiments

In this study, we perform unsupervised representation learning on fMoW and GeoImageNet datasets. We then evaluate the learned representations/pre-trained models on a variety of downstream tasks including image recognition, object detection and semantic segmentation benchmarks on remote sensing and conventional images.

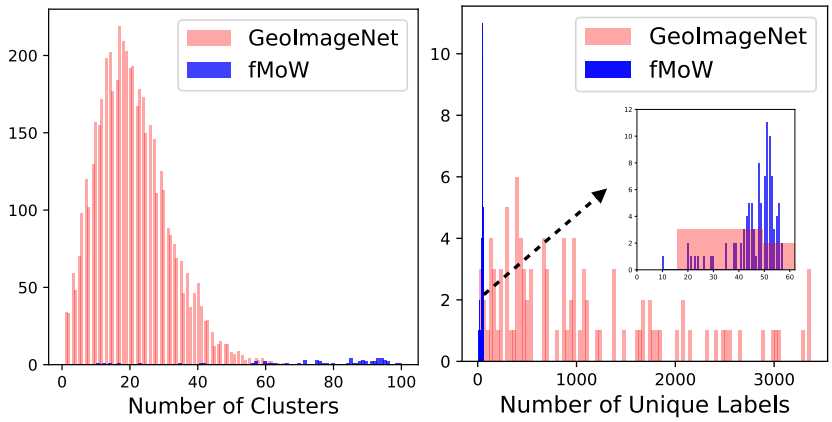


Figure 4.7: **Left** shows the number of clusters per label and **Right** shows the number of unique labels per cluster in fMoW and GeoImageNet. Labels represent the original classes in fMoW and GeoImageNet.

Implementation Details for Unsupervised Learning *For contrastive learning*, similar to MoCo-v2 [5], we use ResNet-50 to paramaterize the query and key encoders, f_q and f_k, in all experiments. We use following hyper-parameters in the MoCo-v2 pre-training step: learning rate of 1e-3, batch size of 256, dictionary queue of size 65536, temperature scaling of 0.2 and SGD optimizer. We use similar setups for both fMoW and

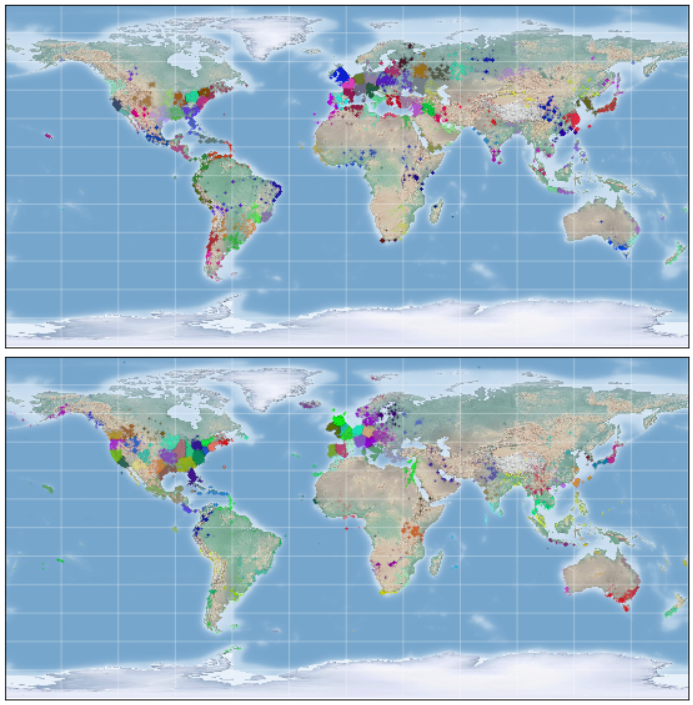


Figure 4.8: **Top** and **Bottom** show the distributions of the fMoW and GeoImageNet clusters.

GeoImageNet datasets. Finally, for each downstream experiment, we report results for the representations learned after 200 epochs.

For geo-location classification task, we run k-Means clustering algorithm to cluster fMoW and GeoImageNet into $K = 100$ geo-clusters given their latitude and longitude pairs. We show the clusters in Fig. 4.8. As seen in the figure, while both datasets have similar clusters there are some differences, particularly in North America and Europe. In Fig. 4.7 we analyze the clusters in GeoImageNet and fMoW. The figure shows that the number of clusters per class on GeoImageNet tend to be skewed towards smaller numbers than fMoW whereas the number of unique classes per cluster on GeoImageNet has more of a uniform distribution. For fMoW, we can conclude that each cluster contain samples from most of the classes. Finally, when adding the geo-location classification task into the contrastive learning we tune α and β to be 1.

Methods We compare our unsupervised learning approach to **supervised learning** for image recognition task. For object detection, and semantic segmentation we compare them to pre-trained weights obtained using (a) **supervised learning**, and (b) **random initilization** while fine-tuning on the targert task dataset. Finally, for ablation analysis we provide results using different combinations of our methods. When appending only geo-location classification task or temporal positives into **MoCo-v2** we use **MoCo-v2+Geo** and **MoCo-v2+TP**. When adding both of our approaches into **MoCo-v2** we use **MoCo-v2+Geo+TP**.

4.5.1 Experiments on fMoW

We first perform experiments on fMoW image recognition task. Similar to the common protocol of evaluating unsupervised pre-training methods [5, 2], we freeze the features and train a supervised linear classifier. However, in practice, it is more common to finetune the features end-to-end in a downstream task. For completeness and a better comparison, we report end-to-end finetuning results for the 62-class fMoW classification as well. We report both top-1 accuracy and F1-scores for this task.

Classifying Single Images In Table 4.1, we report the results on single image classification on fMoW. We would like to highlight that in this case we classify each image individually. In other words, we do not use the prior information that multiple images over the same area $(x_i^1, x_i^2, \ldots, x_i^{T_i})$ have the same labels $(y_i, y_i, \ldots, y_i)$. For evaluation, we use 53,041 images. Our results on this task (linear classification on frozen features) show that MoCo-v2 performs reasonably well on a large-scale dataset with 60.69% accuracy, 8% less than the supervised learning methods. *Sup. Learning (IN wts. init.)* and *Sup. Learning (Scratch)* correspond to supervised learning method starting from imagenet pre-trained weights and random weights respectively. This result aligns with MoCo-v2's performance on the ImageNet dataset [5]. Next, by incorporating geo-location classification task into MoCo-v2, we improve by 3.38% in top-1 classification accuracy. We further improve the results to 68.32% using temporal positives, bridging the gap between the MoCo-v2 baseline and supervised learning to less than 1%. However, when we perform end-to-end finetuning for the classification task, we observe that our method surpasses the supervised learning methods by more than 2%. For completeness, we also include results for MoCo-v2 pre-trained on Imagenet dataset (4th row in

	Backbone	F1-Score ↑ (Frozen/Finetune)	Accuracy ↑ (Frozen/Finetune)
Sup. Learning (IN wts. init.)*	ResNet50	-/64.72	-/69.07
Sup. Learning (Scratch)*	ResNet50	-/64.71	-/69.05
Geoloc. Learning*	ResNet50	48.96/52.23	52.40/56.59
MoCo-V2 (pre. on IN)	ResNet50	31.55/57.36	37.05/62.90
MoCo-V2	ResNet50	55.47/60.61	60.69/64.34
MoCo-V2+Geo	ResNet50	61.60/66.60	64.07/69.04
MoCo-V2+TP	ResNet50	**64.53/67.34**	**68.32/71.55**
MoCo-V2+Geo+TP	ResNet50	63.13/66.56	66.33/70.60

Table 4.1: Experiments on fMoW on classifying single images. * indicates a model trained up to epoch with the highest accuracy on the validation set. We use the same set up for Sup. Learning and Geoloc. Learning in the remaining experiments. **Frozen** corresponds to linear classification on frozen features. **Finetune** corresponds to end-to-end finetuning results for the fmow classification.

Table 4.1) and find that the distribution shift between Imagenet and downstream dataset leads to suboptimal performance.

Classifying Temporal Data In the next step, we change how we perform testing across multiple images over an area at different times. In this case, we predict labels from images over an area i.e. make a prediction for each $t \in \{1, \ldots, T_i\}$, and average the predictions from that area. We then use the most confident class prediction to get area-specific class predictions. In this case, we evaluate the performance on 11,231 unique areas that are represented by multiple images at different times. Our results in Table 4.2 show that doing area-specific inference improves the classification accuracies by 4-8% over image-specific inference. Even incorporating temporal positives, we can improve the accuracy by 6.1% by switching from image classification to temporal data classification. Overall, our methods outperform the baseline Moco-v2 by 4-6% and supervised learning by 1.2%. Here we only report temporal classification on top of frozen features. We present the end-to-end fine-tuning results in a later section.

4.5.2 Transfer Learning Experiments

Previously, we performed pre-training experiments on fMoW dataset and quantified the quality of the representations by supervised training a linear layer for image recognition on fMoW. In this section, we perform transfer learning experiments on different low level tasks.

Object Detection

For object detection, we use the xView dataset [76] consisting of high resolution satellite images captured with similar sensors to the ones in the fMoW dataset. The xView dataset consists of 846 very large

	Backbone	F1-Score ↑	Accuracy ↑
Sup. Learning (IN wts. init.)*	ResNet50	68.72 (+4.01)	73.22 (+4.15)
Sup. Learning (Scratch)*	ResNet50	68.73 (+4.02)	73.24 (+4.19)
Geoloc. Learning*	ResNet50	52.01 (+3.05)	56.12 (+3.72)
MoCo-V2 (pre. on IN)	ResNet50	35.93 (+4.38)	42.56 (+5.51)
MoCo-V2	ResNet50	63.96 (+8.49)	68.64 (+7.95)
MoCo-V2+Geo	ResNet50	66.93 (+5.33)	70.48 (+6.41)
MoCo-V2+TP	ResNet50	**70.11 (+5.58)**	**74.42 (+6.10)**
MoCo-V2+Geo+TP	ResNet50	69.56 (+6.43)	72.76 (+6.43)

Table 4.2: Experiments on fMoW on classifying temporal data. In the table, we compare the results to the ones on single image classification. Here we present results corresponding to linear classification on frozen features only. End-to-end finetuning results are present in a later section.

pre-train	AP_{50} ↑
Random Init.	10.75
Sup. Learning (IN wts. init.)	14.44
Sup. Learning (Scratch)	14.42
MoCo-V2	15.45 (+4.70)
MoCo-V2-Geo	15.63 (+4.88)
MoCo-V2-TP	17.65 (+6.90)
MoCo-V2-Geo+TP	**17.74** (+**6.99**)

Table 4.3: Object detection results on the xView dataset.

(∼2000×2000 pixels) satellite images with bounding box annotations for 60 different class categories including airplane, passenger vehicle, maritime vessel, helicopter etc.

Implementation Details We first divide the set of large images into 700 training and 146 test images. Then, we process the large images to create 416×416 pixels images by randomly sampling the bounding box coordinates of the small image and we repeat this process 100 times for each large image. In this process, we ensure that there is less than 25% overlap between any two bounding boxes from the same image. We then use RetinaNet [83] with pre-trained ResNet-50 backbone and fine-tune the full network on the xView training set. To train RetinaNet, we use learning rate of 1e-5 and a batch size of 4 and Adam optimizer.

Qualitative Analyis Table 4.3 shows the object detection performance on the xView test set. We achieve the best results with the addition of temporal positive pair, and geo-location classification pre-text task into MoCo-v2. With our final model, we can outperform the randomly initialized weights by 7% AP and the supervised learning on the fMoW by 3.3% AP.

Image Segmentation

In this section, we perform downstream experiments on the task of Semantic Segmentation on SpaceNet dataset [115]. The SpaceNet datasets consists of 5000 high resolution satellite images with segmentation masks for buildings.

Implementation Details We divide our SpaceNet dataset into training and test sets of 4000 and 1000 images respectively. We use PSAnet [116] network with ResNet-50 backbone to perform semantic segmentation. We train PSAnet network with a batch size of 16 and a learning rate of 0.01 for 100 epochs and use SGD optimizer.

Qualitative Analysis Table 4.4 shows the segmentation performance of differently initialized backbone weights on the SpaceNet test set. Similar to object detection, we achieve the best IoU scores with the addition of temporal positives and geo-location classification task. Our final model outperforms the randomly initialized weights and supervised learning by 3.58% and 2.94% IoU scores. We observe that the gap between the best and worst models shrinks going from the image recognition to object detection, and semantic segmentation task. This aligns with the performance of the MoCo-v2 pre-trained on ImageNet and fine-tuned on the Pascal-VOC object detection and semantic segmentation experiments [2, 5].

pre-train	mIOU ↑
Random Init.	74.93
Imagenet Init.	75.23
Sup. Learning (IN wts. init.)	75.61
Sup. Learning (Scratch)	75.57
MoCo-V2	78.05 (+3.12)
MoCo-V2-Geo	78.42 (+3.49)
MoCo-V2-TP	78.48 (+3.55)
MoCo-V2-Geo+TP	**78.51 (+3.58)**

Table 4.4: Semantic segmentation results on Space-Net.

pre-train	Top-1 Accuracy ↑
Random Init.	51.89
Imagenet Init.	53.46
Sup. Learning (IN wts. init.)	54.67
Sup. Learning (Scratch)	54.46
MoCo-V2	55.18 (+3.29)
MoCo-V2-Geo	**58.23 (+6.34)**
MoCo-V2-TP	57.10 (+5.21)
MoCo-V2-Geo+TP	57.63 (+5.74)

Table 4.5: Land Cover Classification on NAIP dataset.

Land Cover Classification

Finally, we perform transfer learning experiments on land cover classification across 66 land cover classes using high resolution remote sensing images obtained by the USDA's National Agricultural Imagery Program (NAIP). We use the images from the California's Central Valley for the year of 2016. Our final dataset consists of 100,000 training and 50,000 test images. Table 4.5 shows that our method outperforms the randomly initialized weights by 6.34% and supervised learning by 3.77%.

4.5.3 Experiments on GeoImageNet

After fMoW, we adopt our methods for unsupervised learning on fMoW for improving representation learning on the GeoImageNet. Unfortunately, since ImageNet does not contain images from the same area over time we are not able to integrate the temporal positive pairs into the MoCo-v2 objective. However, in our GeoImageNet experiments we show that we can improve MoCo-v2 by introducing geo-location classification pre-text task.

Qualitative Analysis Table 4.6 shows the top-1 and top-5 classification accuracy scores on the test set of GeoImageNet. Surprisingly, with only geo-location classification task we can achieve 22.26% top-1 accuracy. With MoCo-v2 baseline, we get 38.51 accuracy, about 3.47% more than supervised learning method. With the addition of geo-location classification, we can further improve the top-1 accuracy by 1.45%. These results are interesting in a way that MoCo-v2 (200 epochs) on ImageNet-1k performs 8% worse than supervised learning whereas it outperforms supervised learning on our *highly imbalanced* GeoImageNet with 5150 class categories which is about $5\times$ more than ImageNet-1k. We include results on object detection and semantic segmentation on PASCAL VOC dataset in a later section and show that our method leads to improved performance on these tasks.

	Backbone	**Top-1 (Accuracy)** ↑	**Top-5 (Accuracy)** ↑
Sup. Learning (Scratch)	ResNet50	35.04	54.11
Geoloc. Learning	ResNet50	22.26	39.33
MoCo-V2	ResNet50	38.51	57.67
MoCo-V2+Geo	ResNet50	**39.96**	**58.71**

Table 4.6: Experiments on GeoImageNet. We divide the dataset into 443,435 training and 100,000 test images across 5150 classes. We train MoCo-V2 and MoCo-V2+Geo for 200 epochs whereas **Sup. and Geoloc. Learning are trained until they converge**.

4.6 GeoImagenet Spatial Distribution Analysis

Our geo-location classification pre-text task involves learning where in the world certain objects are represented. For example, we want to learn representations good at predicting that the Indian elephants can be found in India. It would not be possible to learn useful representations if Indian elephants could be found all around the world. For this reason, we can expect to learn more meaningful representations with more number of objects represented only in certain places in the world. In this direction, in Figure 4.9 we show that certain animals in GeoImageNet are populated in specific regions of the world whereas some other classes are specific to certain countries.

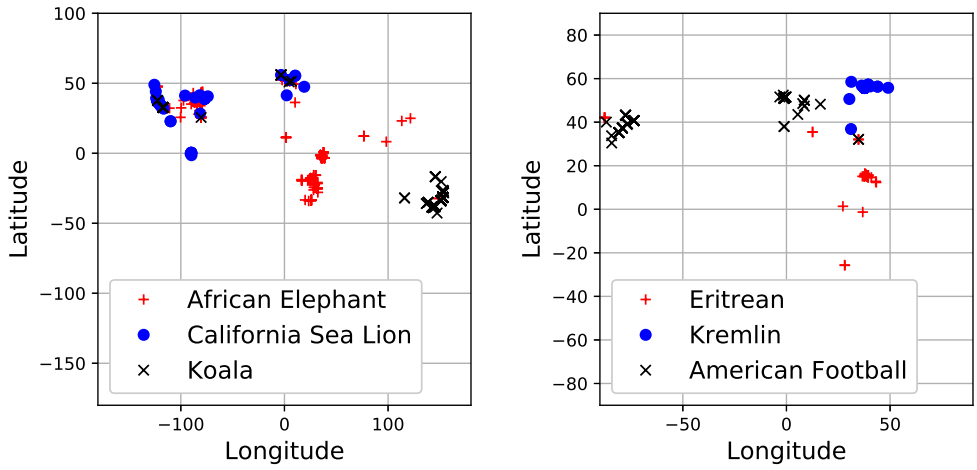


Figure 4.9: Examples of GeoImageNet classes specific to a region in the world. **Left** shows some animals mostly found in the specific regions of the world and **Right** shows some classes specific to certain countries. A small portion of the Koala and African Elephant pictures have been captured in zoos in North America. We note that we do not project coordinates to the world map in this figure.

4.7 Additional Method Details

Algorithm. We show the pseudocode for our proposed methodology (combining temporal positive pairs and geo-location classification pre-text task) in Algorithm 2. It must be noted that details about updating the dictionary queue (or memory bank) is similar to [2] and we refer the readers to the same reference.

Image Recognition Task Details. We would like to add some details regarding the image recognition tasks performed on fMoW and GeoImagenet, the results for which are already mentioned in the main paper. We follow the linear classification protocol similar to [2] for image recognition tasks. After unsupervised pre-training on fMoW/GeoImagenet, we freeze the features and train a supervised linear classifier (a fully-connected layer followed by softmax). We train this classifier on the global average pooling features of

Algorithm 2 Proposed Algorithm

Input: $(\mathcal{X}, \mathcal{C})$ $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_M\}$
m: momentum λ: temperature
for $i = 1, \ldots, M$ **do**
  sample t_1 and t_2 from $(1, 2, \ldots, T_i)$
  sample v and $v^{'}$ from $x_i^{t_1}, x_i^{t_2} \in \mathcal{X}_i$
  $z_i^{t_1} = f_q(v; \theta_q)$
  $z_i^{t_2} = f_k(v^{'}; \theta_k)$
  $l_{pos} = \exp(z_i^{t_1} \cdot z_i^{t_2} / \lambda)$
  $l_{neg} = 0$
  for $j = 1, \ldots, N$ **do**
    $l_{neg} = l_{neg} + \exp(z_i^{t_1} \cdot k_j / \lambda)$
  $L_{z^{t_1}} = -\log \frac{l_{pos}}{l_{pos} + l_{neg}}$
  L_g = cross-entropy$(f_c(z_i^{t_1}; \theta_c), c_i)$
  $L_f = \alpha L_g + \beta L_{z^{t_1}}$
$\theta_p \leftarrow \theta_p + \nabla_{L_f} \theta_p$
$\theta_c \leftarrow \theta_c + \nabla_{L_f} \theta_c$
$\theta_k \leftarrow m * \theta_k + (1 - m) * \theta_q$

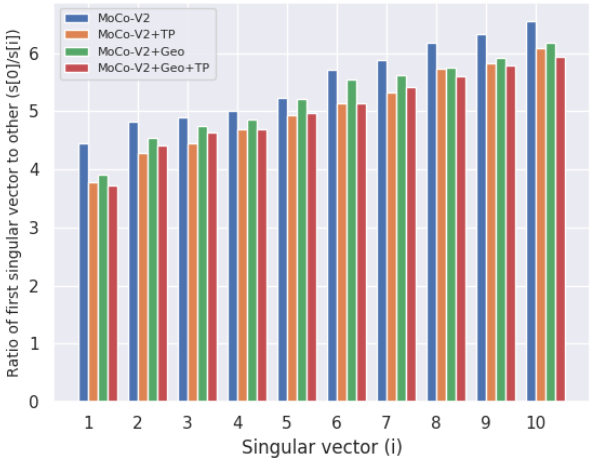


Figure 4.10: Plot of ratio of first singular value to other singular values over the feature space for different methods.

ResNet50. We report top-1 classification accuracy and/or F1-score on the validation set of fMoW/GeoImagenet. We use a learning rate of 1 for fMoW and GeoImagenet for training of the supervised linear classifier.

However, in practice, it is more common to finetune the features end-to-end in a downstream task. For completeness, we report end-to-end finetuning results for the 62-class fMoW classification as well. We report both top-1 accuracy and F1-scores for this task.

4.8 Additional Experiments

4.8.1 Classifying Temporal Data

End-to-End Finetuning Earlier in this chapter, we only report temporal classification (on fMoW) on top of frozen features (see Table 4.2). Here, we present the end-to-end fine-tuning results. Our results in Table 4.7 show that when we perform end-to-end finetuning for the temporal classification task, our method again surpasses the supervised learning methods by more than 2% top 1 accuracy. For completeness, we also include results for MoCo-v2 pre-trained on Imagenet dataset (4th row in Table 4.7) and find that the distribution shift between Imagenet and downstream dataset leads to suboptimal performance.

	Backbone	F1-Score ↑	Accuracy ↑
Sup. Learning (IN wts. init.)*	ResNet50	68.72	73.22
Sup. Learning (Scratch)*	ResNet50	68.73	73.24
Geoloc. Learning*	ResNet50	55.85	59.95
MoCo-V2 (pre. on IN)	ResNet50	61.23	65.95
MoCo-V2	ResNet50	66.63	70.01
MoCo-V2+Geo	ResNet50	70.59	74.23
MoCo-V2+TP	ResNet50	**71.23**	**75.35**
MoCo-V2+Geo+TP	ResNet50	71.01	75.09

Table 4.7: Experiments on fMoW on classifying temporal data. In the table, we compare the results to the ones on single image classification. Here we present results corresponding to **end-to-end finetuning**.

4.8.2 GeoImageNet

Visualizing Distributions of Predictions In this section, we show the distributions of correct and wrong predictions for GeoImageNet on a world map. In Figure 4.11 we show the correct and wrong predictions in top-5 accuracy from the **Supervised learning** and **MoCo-v2+Geo** models on the test set of the GeoImageNet. As we can see in the figure, both models have similar correct and wrong predictions distributions, however, we can see improvements by the **MoCo-v2+Geo** model by zooming-in to the specific parts of the world.

PASCAL VOC Object detection. We also perform experiments in the object detection setting for models pretrained on GeoImagenet. We utilize the Pascal VOC [117] dataset as our target task to perform object detection while pre-training on GeoImageNet with MoCo-v2 and MoCo-v2+Geo models.

Experimental Setup. We use a standard setup for object detection with a Faster R-CNN detector with a R50-C4 backbone as in [2, 118, 119]. We pre-train the backbone on GeoImageNet with MoCo-v2/MoCo-v2+Geo. We finetune for 24k iterations ($\sim$ 23 epochs) on trainval2007 ($\sim$ 5k images). We evaluate on the

pre-train	AP ↑	AP_{50} ↑	AP_{75} ↑
Random Init.	14.51	31.00	11.62
Sup. Learning (Scratch)	37.33	68.56	37.41
MoCo-V2	39.71	69.60	39.86
MoCo-V2-Geo	**41.26**	**70.65**	**42.45**

Table 4.8: Comparison of Pascal-VOC object detection performance. Evaluation is on test2007, fine-tuned end-to-end for 24k iterations (∼23 epochs) on trainval2007.

pre-train	mIOU ↑	mAcc ↑	allAcc ↑
Random Init.	0.45	0.55	0.82
Sup. Learning (Scratch)	0.57	0.65	0.86
MoCo-V2	0.62	0.70	0.88
MoCo-V2-Geo	**0.65**	**0.74**	**0.89**

Table 4.9: Comparison of Pascal-VOC 2012 Semantic Segmentation performance.

VOC test2007 set with the default metric AP50 and the more stringent metrics of COCO-style [120] AP and AP75.

Results. We present results in Table 4.8. It can be seen that with the addition of geo-location classification pre-text task, we can improve the object detection performance by 1.55%, 1.05%, 2.59% AP, AP_{50}, and AP_{75}.

PASCAL VOC Semantic Segmentation. We also perform experiments in the semantic segmentation setting for models pretrained on GeoImagenet. We utilize the Pascal VOC 2012 [117] dataset as our target task to perform semantic segmentation while pre-training on GeoImageNet with MoCo-v2 and MoCo-v2+Geo models.

Experimental Setup. We use PSAnet [116] network with ResNet-50 backbone to perform semantic segmentation. We train PSAnet network with a batch size of 16 and a learning rate of 0.01 for 100 epochs and use SGD optimizer. Similar to object detection, we pre-train the backbone on GeoImageNet with MoCo-v2/MoCo-v2+Geo and then we finetune the network using trainset of PASCAL VOC 2012 segmentation dataset. We evaluate on the testset with the following three metrics: (a) **mIOU**: standard segmentation metric, (b) **mAcc**: mean classwise pixel accuracy, (c) **allAcc**: total pixel accuracy.

Results. We present results in Table 4.9. It can be seen that with the addition of geo-location classification pre-text task, we can improve the segmentation performance by 3% mIOU.

4.8.3 Supervised Learning with Geo-Classification

For completeness we use geo-location classification as an auxiliary task in the supervised learning method. We see an improvement of 1.4% on fMoW image classification showing that using geolocation helps supervised learning method. However, it is still 1.08% less than our best performing model.

4.8.4 Analysis of Features

From our experiments, we find that incorporating spatio-temporal structure of remote-sensing data in a self-supervised contrastive learning framework leads to better performance on multiple downstream tasks. We believe that temporal information results in learning more discriminative features for tasks focusing on spatial variation, such as object detection or semantic segmentation. In addition, geo-location information can complement the information-theoretic objectives typically used by self-supervised learning methods by encouraging representations that reflect geographical information, which is often useful in remote sensing tasks.

We thus believe our proposed methodology is able to encode more bits of information crucial for tasks over remote sensing data. We empirically verify this via performing SVD on the feature matrix obtained over the test set of fMoW and then plot the ratio of 1st singular value to the top 10 singular values. Figure 4.10 show the corresponding plot. We find that the self-supervised features from our approaches have a lower ratio compared to features from vanilla MoCo-v2, which suggests that they are less centered on certain directions.

4.9 Conclusion

In this chapter, we provide a self-supervised learning framework for remote sensing data, where unlabeled data is often plentiful but labeled data is scarce. By leveraging spatially aligned images over time to construct temporal positive pairs in contrastive learning and geo-location in the design of pre-text tasks, we are able to close the gap between self-supervised and supervised learning on image classification, object detection and semantic segmentation on remote sensing and other geo-tagged image datasets. Improvement in performance of these downstream tasks could be useful in many sustainability-related tasks, including poverty prediction, infrastructure measurement, and forest monitoring that require satellite imagery.

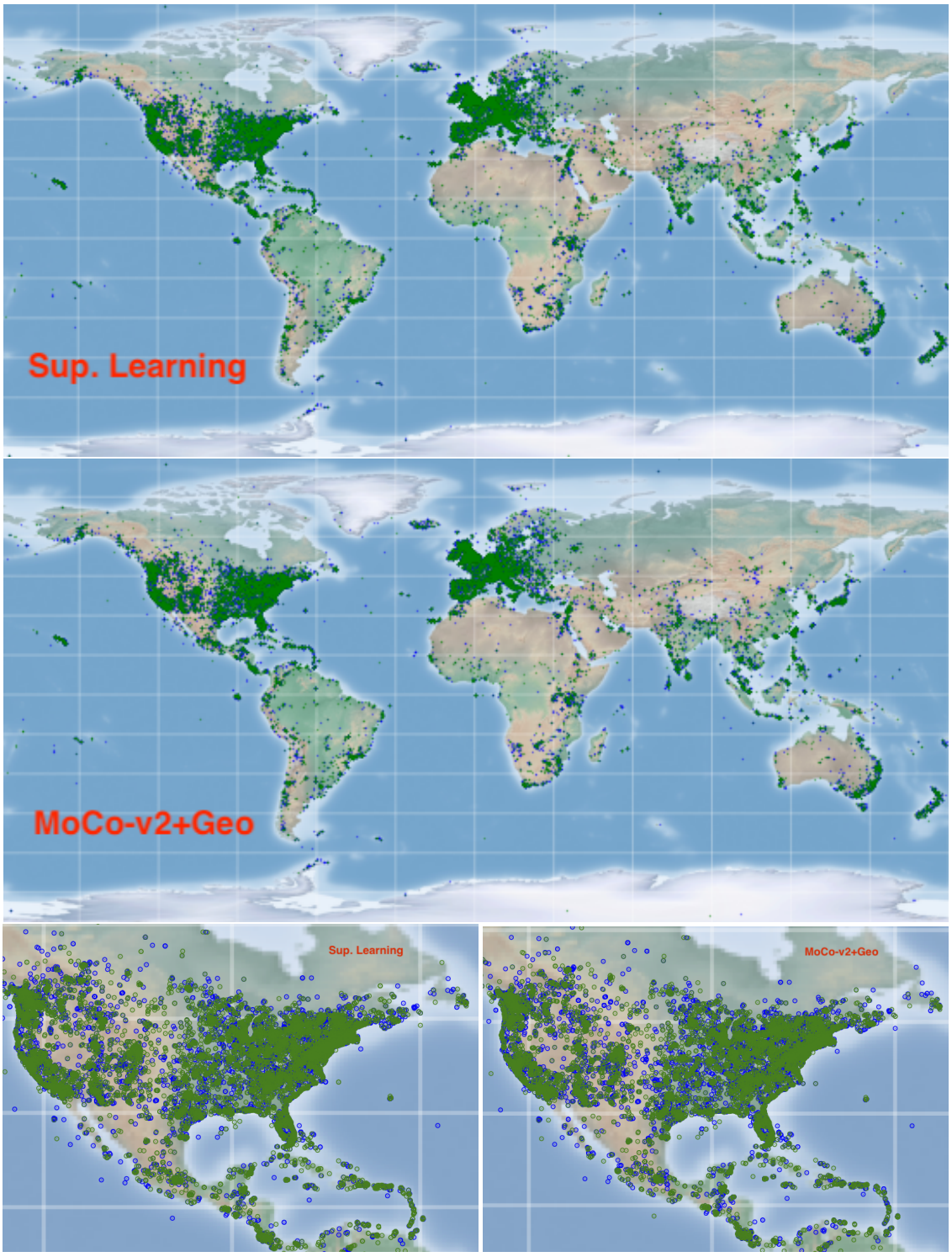


Figure 4.11: Distributions of the predictions by the Supervised learning model (**First Row**) and MoCo-v2+Geo model (**Second Row**) on GeoImageNet test dataset. Green and Blue represent the successfully predicted images and failures respectively. We can see that both model have similar distribution. We note that Supervised learning and MoCo-v2+Geo achieves 54.11% and 58.71% top-5 accuracies on the test set. We recommend the readers to zoom-in to see the differences between two models on Northern America in **Third Row**.

Chapter 5

Negative Data Augmentation

5.1 Introduction

Data augmentation strategies for synthesizing new data in a way that is consistent with an underlying task are extremely effective in both supervised and unsupervised learning [59, 31, 32, 121]. Because they operate at the level of samples, they can be combined with most learning algorithms. They allow for the incorporation of prior knowledge (inductive bias) about properties of typical samples from the underlying data distribution [122, 123], e.g., by leveraging invariances to produce additional "positive" examples of how a task should be solved. To enable users to specify an even wider range of inductive biases, we propose to leverage an alternative and complementary source of prior knowledge that specifies how a task should *not* be solved. We formalize this intuition by assuming access to a way of generating samples that are guaranteed to be out-of-support for the data distribution, which we call a *Negative Data Augmentation* (NDA). Intuitively, negative out-of-distribution (OOD) samples can be leveraged as a useful inductive bias because they provide information about the support of the data distribution to be learned by the model. For example, in a density estimation problem we can bias the model to avoid putting any probability mass in regions which we know a-priori should have zero probability. This can be an effective prior if the negative samples cover a sufficiently large area.

The best NDA candidates are ones that expose common pitfalls of existing models, such as prioritizing local structure over global structure [124]; this motivates us to consider known transformations from the literature that intentionally destroy the spatial coherence of an image [32, 125, 126], such as Jigsaw transforms.

Building on this intuition, we introduce a new GAN training objective where we use NDA as an additional source of fake data for the discriminator as shown in Fig. 5.1. Theoretically, we can show that if the NDA assumption is valid, optimizing this objective will still recover the data distribution in the limit of infinite data. However, in the finite data regime, there is a need to generalize beyond the empirical distribution [60]. By explicitly providing the discriminator with samples we want to avoid, we are able to bias the generator towards avoiding undesirable samples thus improving generation quality.

Furthermore, we propose a way of leveraging NDA for unsupervised representation learning. We propose a new contrastive predictive coding [3, 127] (CPC) objective that encourages the distribution of representations corresponding to in-support data to become disjoint from that of NDA data.

Empirically, we show that applying NDA with our proposed transformations (e.g., forcing the representation of normal and jigsaw images to be disjoint) improves performance in downstream tasks.

With appropriately chosen NDA strategies, we obtain superior empirical performance on a variety of tasks, with almost no cost in computation. For generative modeling, models trained with NDA achieve better image generation, image translation and anomaly detection performance compared with the same model trained without NDA. Similar gains are observed on representation learning for images (including satellite imagery) and videos over downstream tasks such as image classification, object detection and action recognition. These results suggest that NDA has much potential to improve a variety of self-supervised learning techniques.

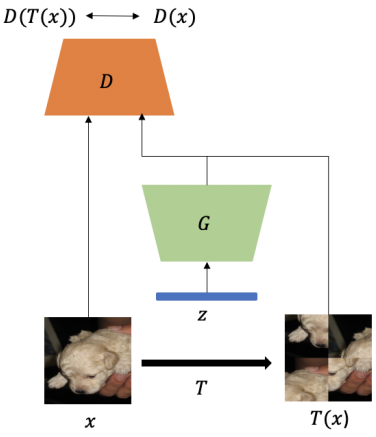


Figure 5.1: Negative Data Augmentation for GANs.

5.2 Negative Data Augmentation

The input to most learning algorithms is a dataset of samples from an underlying data distribution p_{data}. While p_{data} is unknown, learning algorithms always rely on prior knowledge about its properties (inductive biases [128]), e.g., by using specific functional forms such as neural networks. Similarly, data augmentation strategies exploit known invariances of p_{data}, such as the conditional label distribution being invariant to semantic-preserving transformations.

While typical data augmentation strategies exploit prior knowledge about what is in support of p_{data}, in this paper, we propose to exploit prior knowledge about what is *not* in the support of p_{data}. This information is often available for common data modalities (e.g., natural images and videos) and is under-exploited by existing approaches. Specifically, we assume: (1) there exists an alternative distribution $\overline{p}$ such that its support is disjoint from that of p_{data}; and (2) access to a procedure to efficiently sample from $\overline{p}$. We emphasize $\overline{p}$ need not be explicitly defined (e.g., through an explicit density) – it may be implicitly defined by a dataset or by a procedure that transforms samples from p_{data} into ones from $\overline{p}$ by suitably altering their structure.

Analogous to typical data augmentations, NDA strategies are by definition domain and task specific. In this paper, we focus on natural images and videos, and leave the application to other domains (such as natural language processing) as future work. How do we select a good NDA strategy? According to the manifold hypothesis [129], natural images lie on low-dimensional manifolds: p_{data} is supported on a low-dimensional manifold of the ambient (pixel) space. This suggests that many negative data augmentation strategies exist. Indeed, sampling random noise is in most cases a valid NDA. However, while this prior is generic, it is not

Figure 5.2: Negative augmentations produce out-of-distribution samples lacking the typical structure of natural images; these negative samples can be used to inform a model on what it should *not* learn.

very informative, and this NDA will likely be ineffective for most learning problems. Intuitively, NDA is informative if its support is close (in a suitable metric) to that of p_{data}, while being disjoint. These negative samples will provide information on the "boundary" of the support of p_{data}, which we will show is helpful in several learning problems.

In most of our tasks, the images are processed by convolutional neural networks (CNNs) that are good at processing local features but not necessarily global features [124]. Therefore, we may consider NDA examples to be ones that preserve local features ("informative") and break global features, so that it forces the CNNs to learn global features (by realizing NDAs are different from real data).

Leveraging this intuition, we show several image transformations from the literature that can be viewed as generic NDAs over natural images in Figure 5.2, that we will use for generative modeling and representation learning in the following sections. Details about these transformations can be found in Section 5.7.

5.3 NDA for Generative Adversarial Networks

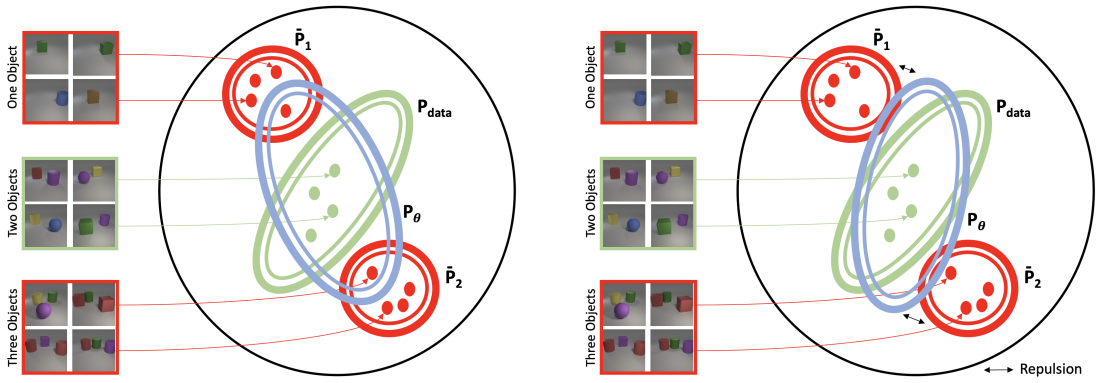


Figure 5.3: Schematic overview of our NDA framework. **Left**: In the absence of NDA, the support of a generative model P_θ (blue oval) learned from samples (green dots) may "over-generalize" and include samples from $\overline{P_1}$ or $\overline{P_2}$. **Right**: With NDA, the learned distribution P_θ becomes disjoint from NDA distributions $\overline{P_1}$ and $\overline{P_2}$, thus pushing P_θ closer to the true data distribution p_{data} (green oval). As long as the prior is consistent, i.e. the supports of $\overline{P_1}$ and $\overline{P_2}$ are truly disjoint from p_{data}, the best fit distribution in the infinite data regime does not change.

In GANs, we are interested in learning a generative model G_θ from samples drawn from some data distribution p_{data} [130]. GANs use a binary classifier, the so-called discriminator D_ϕ, to distinguish real data from generated (fake) samples. The generator G_θ is trained via the following mini-max objective that performs variational Jensen-Shannon divergence minimization:

$$\min_{G_\theta \in \mathcal{P}(\mathcal{X})} \max_{D_\phi} L_{\text{JS}}(G_\theta, D_\phi) \quad \text{where} \tag{5.1}$$

$$L_{\text{JS}}(G_\theta, D_\phi) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} [\log(D_\phi(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{x} \sim G_\theta} [\log(1 - D_\phi(\boldsymbol{x}))] \tag{5.2}$$

This is a special case to the more general variational f-divergence minimization objective [131]. The optimal D_ϕ for any G_θ is $(p_{\text{data}}/G_\theta)/(1 + p_{\text{data}}/G_\theta)$, so the discriminator can serve as a density ratio estimator between p_{data} and G_θ.

With sufficiently expressive models and infinite capacity, G_θ will match p_{data}. In practice, however, we have access to finite datasets and limited model capacity. This means that the generator needs to generalize beyond the empirical distribution, which is challenging because the number of possible discrete distributions scale *doubly exponentially* w.r.t. to the data dimension. Hence, as studied in [60], the role of the inductive bias is critical. For example, [60] report that when trained on images containing 2 objects only, GANs and other generative models can sometimes "generalize" by generating images with 1 or 3 objects (which were never seen in the training set). The generalization behavior – which may or may not be desirable – is determined by factors such as network architectures, hyperparameters, etc., and is difficult to characterize analytically.

Here we propose to bias the learning process by directly specifying what the generator should *not* generate through NDA. We consider an adversarial game based on the following objective:

$$\min_{G_\theta \in \mathcal{P}(\mathcal{X})} \max_{D_\phi} L_{\text{JS}}(\lambda G_\theta + (1 - \lambda)\overline{P}, D_\phi) \tag{5.3}$$

where the negative samples are generated from a mixture of G_θ (the generator distribution) and $\overline{P}$ (the NDA distribution); the mixture weights are controlled by the hyperparameter λ. Intuitively, this can help addresses the above "over-generalization" issue, as we can directly provide supervision on what should not be generated and thus guide the support of G_θ (see Figure 5.3) . For instance, in the object count example above, we can empirically prevent the model from generating images with an undesired number of objects (see Section 5.6 for experimental results on this task).

In addition, the introduction of NDA samples will not affect the solution of the original GAN objective in the limit. In the following theorem, we show that given infinite training data and infinite capacity discriminators and generators, using NDA will not affect the optimal solution to the generator, *i.e.* the generator will still recover the true data distribution.

theoremthmgan Let $\overline{P} \in \mathcal{P}(\mathcal{X})$ be any distribution over $\mathcal{X}$ with disjoint support than p_{data}, *i.e.*, such that $(p_{\text{data}}) \cap (\overline{P}) = \varnothing$. Let $D_\phi : \mathcal{X} \to R$ be the set of all discriminators over $\mathcal{X}$, $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be a convex, semi-continuous function such that $f(1) = 0$, $f^\star$ be the convex conjugate of f, f' its derivative, and G_θ be a

distribution with sample space $\mathcal{X}$. Then $\forall \lambda \in (0, 1]$, we have:

$$\operatorname*{argmin}_{G_\theta \in \mathcal{P}(\mathcal{X})} \max_{D_\phi : \mathcal{X} \to \mathbb{R}} L_f(G_\theta, D_\phi) = \operatorname*{argmin}_{G_\theta \in \mathcal{P}(\mathcal{X})} \max_{D_\phi : \mathcal{X} \to \mathbb{R}} L_f(\lambda G_\theta + (1-\lambda)\overline{P}, D_\phi) = p_{\text{data}} \tag{5.4}$$

where $L_f(Q, D_\phi) = E_{\boldsymbol{x} \sim p_{\text{data}}}[D_\phi(\boldsymbol{x})] - E_{\boldsymbol{x} \sim Q}[f^\star(D_\phi(\boldsymbol{x}))]$ is the objective for f-GAN [131]. However, the optimal discriminators are different for the two objectives:

$$\arg\max_{D_\phi : \mathcal{X} \to \mathbb{R}} L_f(G_\theta, D_\phi) = f'(p_{\text{data}}/G_\theta) \tag{5.5}$$

$$\arg\max_{D_\phi : \mathcal{X} \to \mathbb{R}} L_f(\lambda G_\theta + (1-\lambda)\overline{P}, D_\phi) = f'(p_{\text{data}}/(\lambda G_\theta + (1-\lambda)\overline{P})) \tag{5.6}$$

See Section 5.8.

The above theorem shows that in the limit of infinite data and computation, adding NDA changes the optimal discriminator solution but not the optimal generator. In practice, when dealing with finite data, existing regularization techniques such as weight decay and spectral normalization [132] allow potentially many solutions that achieve the same objective value.

The introduction of NDA samples allows us to filter out certain solutions by providing additional inductive bias through OOD samples. In fact, the optimal discriminator will reflect the density ratio between p_{data} and $\lambda G_\theta + (1-\lambda)\overline{P}$ (see Eq.(5.6)), and its values will be higher for samples from p_{data} compared to those from $\overline{P}$. As we will show in Section 5.5, a discriminator trained with this objective and suitable NDA performs better than relevant baselines for other downstream tasks such as anomaly detection.

5.4 NDA for Constrastive Representation Learning

Using a classifier to estimate a density ratio is useful not only for estimating f-divergences (as in the previous section) but also for estimating mutual information between two random variables. In representation learning, mutual information (MI) maximization is often employed to learn compact yet useful representations of the data, allowing one to perform downstream tasks efficiently [133, 134, 135, 59]. Here, we show that NDA samples are also beneficial for representation learning.

In contrastive representation learning (such as CPC [59]), the goal is to learn a mapping $h_\theta(\boldsymbol{x}) : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$ that maps a datapoint $\boldsymbol{x}$ to some distribution over the representation space $\mathcal{Z}$; once the network h_θ is learned, representations are obtained by sampling from $\boldsymbol{z} \sim h_\theta(\boldsymbol{x})$. CPC *maximizes* the following objective:

$$I_{\text{CPC}}(h_\theta, g_\phi) := \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x}), \boldsymbol{z} \sim h_\theta(\boldsymbol{x}), \widehat{\boldsymbol{z}}_i \sim p_\theta(\boldsymbol{z})} \left[\log \frac{n g_\phi(\boldsymbol{x}, \boldsymbol{z})}{g_\phi(\boldsymbol{x}, \boldsymbol{z}) + \sum_{j=1}^{n-1} g_\phi(\boldsymbol{x}, \widehat{\boldsymbol{z}}_j)} \right] \tag{5.7}$$

where $p_\theta(\boldsymbol{z}) = \int h_\theta(\boldsymbol{z}|\boldsymbol{x}) p_{\text{data}}(\boldsymbol{x}) \boldsymbol{x}$ is the marginal distribution of the representations associated with p_{data}. Intuitively, the CPC objective involves an n-class classification problem where g_ϕ attempts to identify

a matching pair (i.e. $(\boldsymbol{x}, \boldsymbol{z})$) sampled from the joint distribution from the $(n-1)$ non-matching pairs (i.e. $(\boldsymbol{x}, \widehat{\boldsymbol{z}}_j)$) sampled from the product of marginals distribution. Note that g_ϕ plays the role of a discriminator/-critic, and is implicitly estimating a density ratio. As $n \to \infty$, the optimal g_ϕ corresponds to an un-normalized density ratio between the joint distribution and the product of marginals, and the CPC objective matches its upper bound which is the mutual information between X and Z [136, 137].

However, this objective is no longer able to control the representations for data that are out of support of p_{data}, so there is a risk that the representations are similar between p_{data} samples and out-of-distribution ones.

To mitigate this issue, we propose to use NDA in the CPC objective, where we additionally introduce a batch of NDA samples, for each positive sample:

$$\overline{I_{\text{CPC}}}(h_\theta, g_\phi) := \mathbb{E}\left[\log \frac{(n+m)g_\phi(\boldsymbol{x}, \boldsymbol{z})}{g_\phi(\boldsymbol{x}, \boldsymbol{z}) + \sum_{j=1}^{n-1} g_\phi(\boldsymbol{x}, \widehat{\boldsymbol{z}}_j) + \sum_{k=1}^{m} g_\phi(\boldsymbol{x}, \overline{\boldsymbol{z}}_k)}\right] \tag{5.8}$$

where the expectation is taken over $\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x}), \boldsymbol{z} \sim h_\theta(\boldsymbol{x}), \widehat{\boldsymbol{z}}_i \sim p_\theta(\boldsymbol{z}), \overline{\boldsymbol{x}}_k \sim \overline{p}$ (NDA distribution), $\overline{\boldsymbol{z}}_k \sim h_\theta(\overline{\boldsymbol{x}}_k)$ for all $k \in [m]$. Here, the behavior of $h_\theta(\boldsymbol{x})$ when $\boldsymbol{x}$ is NDA is optimized explicitly, allowing us to impose additional constraints to the NDA representations. This corresponds to a more challenging classification problem (compared to basic CPC) that encourages learning more informative representations.

In the following theorem, we show that the proposed objective encourages the representations for NDA samples to become disjoint from the representations for p_{data} samples, *i.e.* NDA samples and p_{data} samples do not map to the same representation.

(Informal) The optimal solution to h_θ in the NDA-CPC objective maps the representations of data samples and NDA samples to disjoint regions. See Section 5.9 for a detailed statement and proof.

5.5 NDA-GAN Experiments

In this section we report experiments with different types of NDA for image generation. Additional details about the network architectures and hyperparameters can be found in Section 5.14.

Unconditional Image Generation. We conduct experiments on various datasets using the BigGAN architecture [138] for unconditional image generation[1]. We first explore various image transformations from the literature to evaluate which ones are effective as NDA. For each transformation, we evaluate its performance as NDA (training as in Eq. 5.3) and as a traditional data augmentation strategy, where we enlarge the training set by applying the transformation to real images (denoted PDA for positive data augmentation). Table 5.1 shows the FID scores for different types of transformations as PDA/NDA. The results suggest that transformations that spatially corrupt the image are strong NDA candidates. It can be seen that Random Horizontal Flip is not effective as an NDA; this is because flipping does not spatially corrupt the image but is rather a semantic preserving transformation, hence the NDA distribution $\overline{P}$ is not disjoint from p_{data}. On

[1] We feed a single label to all images to make the architecture suitable for unconditional generation.

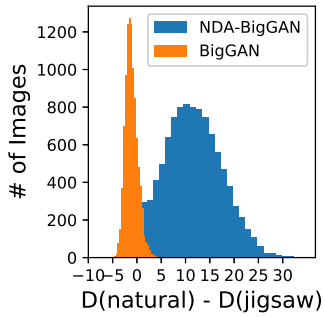


Figure 5.4: Histogram of difference in the discriminator output for a real image and it's Jigsaw version.

Table 5.1: FID scores over CIFAR-10 using different transformations as PDA and NDA in BigGAN. The results indicate that some transformations yield better results when used as NDA. The common feature of such transformations is they all spatially corrupt the images.

w/o Aug.	Jigsaw		Cutout		Stitch		Mixup		Cutmix		Random Crop		Random Flip		Gaussian	
	PDA	NDA	PDA	NDA	PDA	NDA	PDA	NDA	PDA	NDA	PDA	NDA	PDA	NDA	PDA	NDA
18.64	98.09	12.61	79.72	14.69	108.69	13.97	70.64	17.29	90.81	15.01	20.02	15.05	16.65	124.32	44.41	18.72

the contrary, it is reasonable to assume that if an image is likely under p_{data}, its flipped variant should also be likely. This is confirmed by the effectiveness of this strategy as PDA.

We believe spatially corrupted negatives perform well as NDA in that they push the discriminator to focus on global features instead of local ones (e.g., texture). We confirm this by plotting the histogram of differences in the discriminator output for a real image and it's Jigsaw version as shown in Fig. 5.4. We show that the difference is *(a)* centered close to zero for normal BigGAN (so *without NDA training, the discriminator cannot distinguish real and Jigsaw samples well*), and *(b)* centered at a positive number (logit 10) for our method (NDA-BigGAN). Following our findings, in our remaining experiments we use Jigsaw, Cutout, Stitch, Mixup and Cutmix as they achieve significant improvements when used as NDA for unconditional image generation on CIFAR-10.

Table 5.2 shows the FID scores for BigGAN when trained with five types of negative data augmentation on four different benchmarks. Almost all the NDA augmentations improve the baseline across datasets. For all the datasets except CIFAR-100, $\lambda = 0.25$, whereas for CIFAR-100 it is 0.5. We show the effect of λ on CIFAR-10 performance in Section 5.12. We additionally performed an experiment using a mixture of augmentation policy. The results (FID 16.24) were better than the baseline method (18.64) but not as good as using a single strategy.

Conditional Image Generation. We also investigate the benefits of NDA in conditional image generation using BigGAN. The results are shown in Table 5.3. In this setting as well, NDA gives a significant boost over the baseline model. We again use $\lambda = 0.25$ for CIFAR-10 and $\lambda = 0.5$ for CIFAR-100. For both unconditional and conditional setups we find the Jigsaw and Stitching augmentations to achieve a better FID

Table 5.2: Comparison of FID scores of different types of NDA for unconditional image generation on various datasets. The numbers in bracket represent the corresponding image resolution in pixels. Jigsaw consistently achieves the **best** or second best result.

	BigGAN	Jigsaw	Stitching	Mixup	Cutout	Cutmix	CR-BigGAN
CIFAR-10 (32)	18.64	**12.61**	13.97	17.29	14.69	15.01	14.56
CIFAR-100 (32)	22.19	**19.72**	20.99	22.21	22.08	20.78	–
CelebA (64)	38.14	37.24	**37.17**	37.51	37.39	37.46	–
STL10 (32)	26.80	**23.94**	26.08	24.45	24.91	25.34	–

score than the other augmentations.

Table 5.3: FID scores for conditional image generation using different NDAs.[2]

	BigGAN	Jigsaw	Stitching	Mixup	Cutout	Cutmix	CR-BigGAN
C-10	11.51	**9.42**	9.47	13.87	10.52	10.3	11.48
C-100	15.04	14.12	**13.90**	15.27	14.21	13.99	–

Image Translation. Next, we apply the NDA method to image translation. In particular, we use the Pix2Pix model [139] that can perform image-to-image translation using GANs provided paired training data. Here, the generator is conditioned on an image $\mathcal{I}$, and the discriminator takes as input the concatenation of generated/real image and $\mathcal{I}$. We use Pix2Pix for semantic segmentation on Cityscapes dataset [140] (i.e. photos $\rightarrow$ labels). Table 5.4 shows the quantitative gains obtained by using Jigsaw NDA[3] while Figure 5.5 highlights the qualitative improvements. The NDA-Pix2Pix model avoids noisy segmentation on objects including buildings and trees.

Table 5.4: Results on CityScapes, using per pixel accuracy (Pp.), per class accuracy (Pc.) and mean Intersection over Union (mIOU). We compare Pix2Pix and its NDA version.

Metric	Pp.	Pc.	mIOU
Pix2Pix (cGAN)	0.80	0.24	0.27
NDA (cGAN)	**0.84**	**0.34**	**0.28**
Pix2Pix (L1+cGAN)	0.72	0.23	0.18
NDA (L1+cGAN)	**0.75**	**0.28**	**0.22**

Anomaly Detection. As another added benefit of NDA for GANs, we utilize the output scores of the BigGAN discriminator for anomaly detection. We experiment with 2 different types of OOD datasets. The

[2]We use a PyTorch code for BigGAN. The number reported in brock2018large for C-10 is 14.73.
[3]We use the official PyTorch implementation and show the best results.

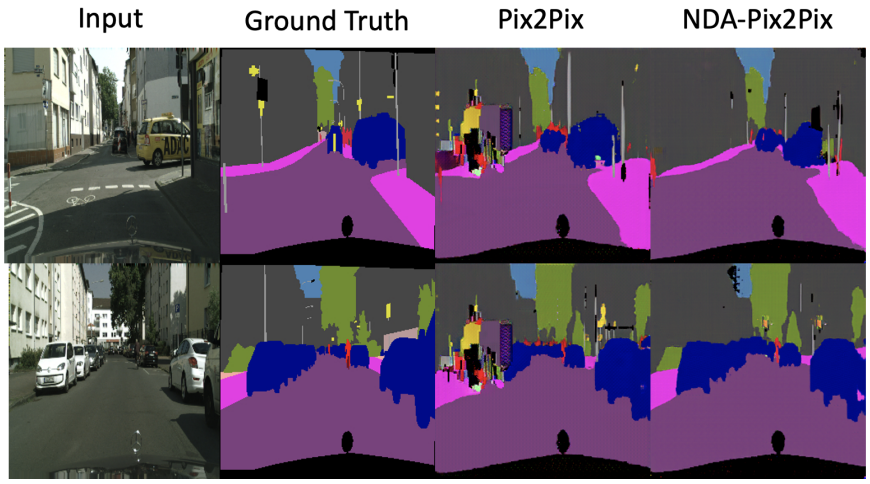


Figure 5.5: Qualitative results on Cityscapes.

Table 5.5: AUROC scores for different OOD datasets. OOD-1 contains different datasets, while OOD-2 contains the set of 19 different corruptions in CIFAR-10-C [1] (the average score is reported).

		BigGAN	Jigsaw	EBM
OOD-1	DTD	0.70	0.69	0.48
	SVHN	0.75	0.61	0.63
	Places-365	0.35	0.58	0.68
	TinyImageNet	0.40	0.62	0.67
	CIFAR-100	0.63	0.64	0.50
	Average	0.57	**0.63**	0.59
OOD-2	CIFAR-10-C	0.56	**0.63**	0.60

first set consists of SVHN [141], DTD [142], Places-365 [143], TinyImageNet, and CIFAR-100 as the OOD datapoints following the protocol in [144, 145]. We train BigGAN w/ and w/o Jigsaw NDA on the train set of CIFAR-10 and then use the output value of discriminator to classify the test set of CIFAR-10 (not anomalous) and different OOD datapoints (anomalous) as anomalous or not. We use the AUROC metric as proposed in [146] to evaluate the anomaly detection performance. Table 5.5 compares the performance of NDA with a likelihood based model (Energy Based Models (EBM [144]). Results show that Jigsaw NDA performs much better than baseline BigGAN and other generative models. We did not include other NDAs as Jigsaw achieved the best results.

We consider the extreme corruptions in CIFAR-10-C [1] as the second set of OOD datasets. It consists of 19 different corruptions, each having 5 different levels of severity. We only consider the corruption of highest severity for our experiment, as these constitute a significant shift from the true data distribution. Averaged over all the 19 different corruptions, the AUROC score for the normal BigGAN is **0.56**, whereas the BigGAN

trained with Jigsaw NDA achieves **0.63**. The histogram of difference in discriminator's output for clean and OOD samples are shown in Figure 5.8. High difference values imply that the Jigsaw NDA is better at distinguishing OOD samples than the normal BigGAN.

Unsupervised Learning on Images. In this section, we perform experiments on three benchmarks: (a) CIFAR10 (C10), (b) CIFAR100 (C100), (c) ImageNet-100 [78] and (d) fMoW [112] to show the benefits of NDA on representation learning with the contrastive loss function. In our experiments, we use the momentum contrast method [3], *MoCo-V2*, as it is currently the state-of-the-art model on unsupervised learning on ImageNet. For C10 and C100, we train the MoCo-V2 model for unsupervised learning (w/ and w/o NDA) for 1000 epochs. On the other hand, for ImageNet-100 and fMoW, we train the MoCo-V2 model (w/ and w/o NDA) for 200 epochs. Additional hyperparameter details can be found in a later section. To evaluate the representations, we train a linear classifier on the representations on the same dataset with labels.

Table 5.6 shows the top-1 accuracy of the classifier. We find that across all the three datasets, different NDA approaches outperform *MoCo-V2*. While Cutout NDA performs the best for C10, the best performing NDA for C100 and ImageNet-100 are Jigsaw and Mixup respectively. For fMoW, we show only results for baseline MoCo-V2, Jigsaw NDA and Stitching NDA and we observe that Jigsaw NDA gives the best performance followed by Stitching NDA. Figure 5.9 compares the cosine distance of the representations learned w/ and w/o NDA (jigsaw) and shows that jigsaw and normal images are projected far apart from each other when trained using NDA whereas with original MoCo-v2 they are projected close to each other.

Table 5.6: Top-1 accuracy results on image recognition w/ and w/o NDA on MoCo-V2.

	MoCo-V2	Jigsaw	Stitching	Cutout	Cutmix	Mixup
CIFAR-10	91.20	91.66	91.59	**92.26**	91.51	91.36
CIFAR-100	69.63	**70.17**	69.21	69.81	69.83	69.99
ImageNet-100	69.41	69.95	69.54	69.77	69.61	**70.01**
fMoW	60.69	**63.96**	63.53	-	-	-

Transfer Learning for Object Detection. We transfer the network pre-trained over ImageNet-100 for the task of Pascal-VOC object detection using a Faster R-CNN detector (C4 backbone) [147]. We fine-tune the network on Pascal VOC 2007+2012 trainval set and test it on the 2007 test set. The baseline MoCo achieves 38.47 AP, 65.99 AP50, 38.81 AP75 whereas the MoCo trained with mixup NDA gets 38.72 AP, 66.23 AP50, 39.16 AP75 (an improvement of ≈ 0.3).

Unsupervised Learning on Videos. In this section, we investigate the benefits of NDA in self-supervised learning of spatio-temporal embeddings from video, suitable for human action recognition. We apply NDA to Dense Predictive Coding [127], which is a single stream (RGB only) method for self-supervised representation learning on videos. For videos, we create NDA samples by performing the same transformation on all frames of the video (e.g. the same jigsaw permutation is applied to all the frames of a video). We evaluate the approach by first training the DPC model with NDA on a large-scale dataset (UCF101), and then evaluate the representations by training a supervised action classifier on UCF101 and HMDB51 datasets. As

shown in Table 5.7, Jigsaw and Cutmix NDA improve downstream task accuracy on UCF-101 and HMDB-51, achieving new state-of-the-art performance among single stream (RGB only) methods for self-supervised representation learning (when pre-trained using UCF-101).

Table 5.7: Top-1 accuracy results on action recognition in videos w/ and w/o NDA in DPC.

	DPC	Jigsaw	Stitching	Cutout	Cutmix	Mixup
UCF-101 (Pre-trained on UCF-101)	61.35	64.54	**66.07**	64.52	63.52	63.65
HMDB51 (Pre-trained on UCF-101)	45.31	**46.88**	45.31	45.31	**48.43**	43.75

5.6 Numerosity Containment

[60] systematically investigate generalization in deep generative models using two different datasets: (a) a toy dataset where there are k non-overlapping dots (with random color and location) in the image (see Figure 5.6a), and (b) the CLEVR dataset where ther are k objects (with random shape, color, location, and size) in the images (see Figure 5.6b). They train a GAN model (WGAN-GP [148]) with (either) dataset and observe that the learned distribution does not produce the same number of objects as in the dataset it was trained on. The distribution of the numerosity in the generated images is centered at the numerosity from the dataset, with a slight-bias towards over-estimation. For, example when trained on images with six dots, the generated images contain anywhere from two to eight dots (see Figure 5.7a). The observation is similar when trained on images with two CLEVR objects. The generated images contain anywhere from one to three dots (see Figure 5.7b).

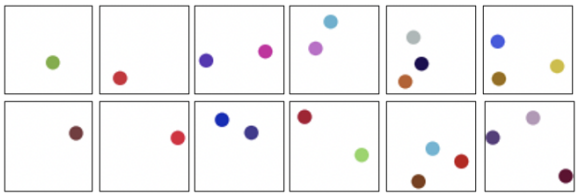
In order to remove samples with numerosity different from the train dataset, we use such samples as negative data during training. For example, while training on images with six dots we use images with four, five and seven dots as negative data for the GAN. The resulting distribution of the numerosity in the generated images is constrained to six. We observe similar behaviour when training a GAN with images containing two CLEVR objects as positive data and images with one or three objects as negative data.

5.7 Image Transformations

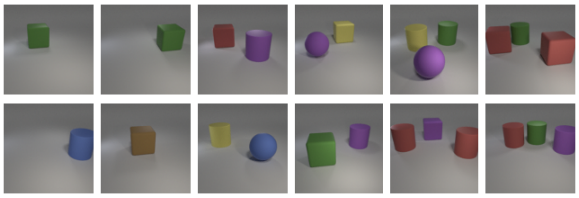
Given an image of size $H \times W$, the different image transformations that we used are described below.

Jigsaw-K [32] We partition the image into a grid of $K \times K$ patches of size $(H/K) \times (W/K)$, indexed by $[1, \ldots, K \times K]$. Then we shuffle the image patches according to a random permutation (different from the original order) to produce the NDA image. Empirically, we find $K = 2$ to work the best for Jigsaw-K NDA.

Stitching We stitch two equal-sized patches of two different images, either horizontally $(H/2 \times W)$ or vertically $(H \times W/2)$, chosen uniformly at random, to produce the NDA image.

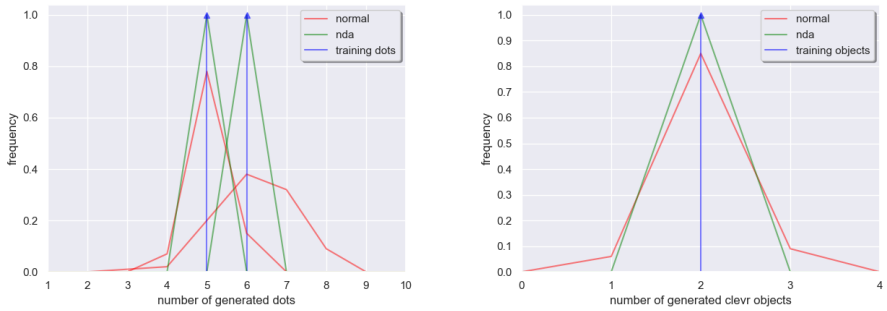


(a) Dots



(b) CLEVR

Figure 5.6: Toy Datasets used in Numerosity experiments.



(a) (b)

Figure 5.7: **Left**: Distribution over number of dots. The arrows are the number of dots the learning algorithm is trained on, and the solid line is the distribution over the number of dots the model generates. **Right**: Distribution over number of CLEVR objects the model generates. Generating CLEVR is harder so we explore only one, but the behaviour with NDA is similar to dots.

Cutout / Cutmix We select a random patch in the image with its height and width lying between one-third and one-half of the image height and width respectively. To construct NDA images, this patch is replaced with the mean pixel value of the patch (like cutout [125] with the only difference that they use zero-masking), or the pixel values of another image at the same location (cutmix [126]).

Mixup-α NDA image is constructed from a linear interpolation between two images $\boldsymbol{x}$ and $\boldsymbol{y}$ [149], $\gamma\boldsymbol{x} + (1-\gamma)\boldsymbol{y}$; $\gamma \sim \text{Beta}(\alpha, \alpha)$. α is chosen so that the distribution has high density at 0.5.

Other classes NDA images are sampled from other classes in the same dataset. See Section 5.6.

5.8 NDA for GANs

Let us use $p(x), \overline{p}(x), q(x)$ to denote the density functions of $p_{\text{data}}, \overline{P}$ and G_θ respectively (and P, $\overline{P}$, Q for the respective distributions). First, from Lemma 1 in [134], we have that

$$\max_{D_\phi:\mathcal{X}\to\mathbb{R}} L_f(G_\theta, D_\phi) = D_f(P\|G_\theta) \tag{5.9}$$

$$\max_{D_\phi:\mathcal{X}\to\mathbb{R}} L_f(\lambda G_\theta + (1-\lambda)\overline{P}, D_\phi) = D_f(P\|\lambda Q + (1-\lambda)\overline{P}) \tag{5.10}$$

where D_f refers to the f-divergence. Then, we have

$$\begin{aligned}
& D_f(P||\lambda Q + (1-\lambda)\overline{P}) \\
=& \int_{\mathcal{X}} (\lambda q(x) + (1-\lambda)\overline{p}(x))\, f\left(\frac{p(x)}{\lambda q(x) + (1-\lambda)\overline{p}(x)}\right) \\
=& \int_{\mathcal{X}} \lambda q(x) f\left(\frac{p(x)}{\lambda q(x) + (1-\lambda)\overline{p}(x)}\right) + (1-\lambda) f(0) \\
\geq& \lambda f\left(\int_{\mathcal{X}} q(x) \frac{p(x)}{\lambda q(x) + (1-\lambda)\overline{p}(x)}\right) + (1-\lambda) f(0) \qquad (5.11) \\
=& \lambda f\left(\frac{1}{\lambda}\int_{\mathcal{X}} \lambda q(x) \frac{p(x)}{\lambda q(x) + (1-\lambda)\overline{p}(x)}\right) + (1-\lambda) f(0) \\
=& \lambda f\left(\frac{1}{\lambda}\int_{\mathcal{X}} (\lambda q(x) + (1-\lambda)\overline{p}(x) - (1-\lambda)\overline{p}(x)) \frac{p(x)}{\lambda q(x) + (1-\lambda)\overline{p}(x)}\right) + (1-\lambda) f(0) \\
=& \lambda f\left(\frac{1}{\lambda} - \int_{\mathcal{X}} ((1-\lambda)\overline{p}(x)) \frac{p(x)}{\lambda q(x) + (1-\lambda)\overline{p}(x)}\right) + (1-\lambda) f(0) \\
=& \lambda f\left(\frac{1}{\lambda}\right) + (1-\lambda) f(0) \qquad (5.12)
\end{aligned}$$

where we use the fact that f is convex with Jensen's inequality in Eq.(5.11) and the fact that $p(x)\overline{p}(x) = 0, \forall x \in \mathcal{X}$ in Eq.(5.12) since P and $\overline{P}$ has disjoint support.

We also have

$$\begin{aligned}
D_f(P||\lambda P + (1-\lambda)\overline{P}) &= \int_{\mathcal{X}} (\lambda p(x) + (1-\lambda)\overline{p}(x))\, f\left(\frac{p(x)}{\lambda p(x) + (1-\lambda)\overline{p}(x)}\right) \\
&= \int_{\mathcal{X}} (\lambda p(x))\, f\left(\frac{p(x)}{\lambda p(x) + (1-\lambda)\overline{p}(x)}\right) + (1-\lambda) f(0) \\
&= \int_{\mathcal{X}} (\lambda p(x))\, f\left(\frac{p(x)}{\lambda p(x) + 0}\right) + (1-\lambda) f(0) \\
&= \lambda f\left(\frac{1}{\lambda}\right) + (1-\lambda) f(0)
\end{aligned}$$

Therefore, in order for the inequality in Equation 5.11 to be an equality, we must have that $q(\boldsymbol{x}) = p(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. Therefore, the generator distribution recovers the data distribution at the equlibrium posed by the

NDA-GAN objective, which is also the case for the original GAN objective.

Moreover, from Lemma 1 in [134], we have that:

$$\arg\max_{D_\phi} L_f(Q, D_\phi) = f'(p_{\text{data}}/Q) \tag{5.13}$$

Therefore, by replacing Q with G_θ and $(\lambda G_\theta + (1-\lambda)\overline{P})$, we have:

$$\arg\max_{D_\phi:\mathcal{X}\to\mathbb{R}} L_f(G_\theta, D_\phi) = f'(p_{\text{data}}/G_\theta) \tag{5.14}$$

$$\arg\max_{D_\phi:\mathcal{X}\to\mathbb{R}} L_f(\lambda G_\theta + (1-\lambda)\overline{P}, D_\phi) = f'(p_{\text{data}}/(\lambda G_\theta + (1-\lambda)\overline{P})) \tag{5.15}$$

which shows that the optimal discriminators are indeed different for the two objectives.

5.9 NDA for Contrastive Representation Learning

We describe the detailed statement of Theorem 2 and proof as follows. theoremthmcpc For some distribution $\overline{p}$ over $\mathcal{X}$ such that $(\overline{p}) \cap (p_{\text{data}}) = \varnothing$, and for any maximizer of the NDA-CPC objective

$$\hat{h} \in \arg\max_{h_\theta} \max_{g_\phi} \overline{I_{\text{CPC}}}(h_\theta, g_\phi)$$

the representations of negative samples are disjoint from that of positive samples for $\hat{h}$; *i.e.*, $\forall \boldsymbol{x} \in (p_{\text{data}}), \bar{\boldsymbol{x}} \in (\overline{p})$,

$$(\hat{h}(\bar{\boldsymbol{x}})) \cap (\hat{h}(\boldsymbol{x})) = \varnothing$$

We use a contradiction argument to establish the proof. For any representation mapping that maximizes the NDA-CPC objective,

$$\hat{h} \in \arg\max_{h_\theta} \max_{g_\phi} \overline{I_{\text{CPC}}}(h_\theta, g_\phi)$$

suppose that the positive and NDA samples share some support, i.e., $\exists \boldsymbol{x} \in (p_{\text{data}}), \bar{\boldsymbol{x}} \in (\overline{p})$,

$$(\hat{h}(\bar{\boldsymbol{x}})) \cap (\hat{h}(\boldsymbol{x})) \neq \varnothing$$

We can always construct $\hat{h}'$ that shares the same representation with $\hat{h}$ for p_{data} but have disjoint representations for NDA samples; i.e., $\forall \boldsymbol{x} \in (p_{\text{data}}), \bar{\boldsymbol{x}} \in (\overline{p})$, the following two statements are true:

1. $\hat{h}(\boldsymbol{x}) = \hat{h}'(\boldsymbol{x})$;
2. $(\hat{h}'(\bar{\boldsymbol{x}})) \cap (\hat{h}'(\boldsymbol{x})) = \varnothing$.

Our goal is to prove that:

$$\max_{g_\phi} \overline{I_{\mathrm{CPC}}}(\hat{h}', g_\phi) > \max_{g_\phi} \overline{I_{\mathrm{CPC}}}(\hat{h}, g_\phi) \tag{5.16}$$

which shows a contradiction.

For ease of exposition, let us allow zero values for the output of g, and define $0/0 = 0$ (in this case, if g assigns zero to positive values, then the CPC objective becomes $-\infty$, so it cannot be a maximizer to the objective).

Let $\hat{g} \in \arg\max \overline{I_{\mathrm{CPC}}}(\hat{h}, g_\phi)$ be an optimal critic to the representation model $\hat{h_\theta}$. We then define a following critic function:

$$\hat{g}'(\boldsymbol{x}, \boldsymbol{z}) = \begin{cases} \hat{g}(\boldsymbol{x}, \boldsymbol{z}) & \text{if } \exists \boldsymbol{x} \in (p_{\mathrm{data}}) \quad s.t. \quad \boldsymbol{z} \in (\hat{h}'(\boldsymbol{x})) \\ 0 & \text{otherwise} \end{cases} \tag{5.17}$$

In other words, the critic assigns the same value for data-representation pairs over the support of p_{data} and zero otherwise. From the assumption over $\hat{h}$, $\exists \boldsymbol{x} \in (p_{\mathrm{data}}), \bar{\boldsymbol{x}} \in (\overline{p})$, and $\overline{\boldsymbol{z}} \in (\hat{h}(\bar{\boldsymbol{x}}))$,

$$\overline{\boldsymbol{z}} \in (\hat{h}(\boldsymbol{x}))$$

so $(\boldsymbol{x}, \overline{\boldsymbol{z}})$ can be sampled as a positive pair and $\hat{g}(\boldsymbol{x}, \overline{\boldsymbol{z}}) > 0$.

Therefore,

$$\begin{aligned}
&\max_{g_\phi} \overline{I_{\mathrm{CPC}}}(\hat{h}', g_\phi) \geq \overline{I_{\mathrm{CPC}}}(\hat{h}', \hat{g}') && (5.18)\\
&= \mathbb{E}\left[\log \frac{(n+m)\hat{g}'(\boldsymbol{x}, \boldsymbol{z})}{\hat{g}'(\boldsymbol{x}, \boldsymbol{z}) + \sum_{j=1}^{n-1} \hat{g}'(\boldsymbol{x}, \widehat{\boldsymbol{z}_j}) + \sum_{k=1}^{m} \underbrace{\hat{g}'(\boldsymbol{x}, \overline{\boldsymbol{z}_k})}_{=0}} \right] && \text{(plug in definition for NDA-CPC)}\\
&\geq \mathbb{E}\left[\log \frac{(n+m)\hat{g}(\boldsymbol{x}, \boldsymbol{z})}{\hat{g}(\boldsymbol{x}, \boldsymbol{z}) + \sum_{j=1}^{n-1} \hat{g}(\boldsymbol{x}, \widehat{\boldsymbol{z}_j}) + \sum_{k=1}^{m} \hat{g}(\boldsymbol{x}, \overline{\boldsymbol{z}_k})} \right] && \text{(existence of some} \hat{g}(\boldsymbol{x}, \overline{\boldsymbol{z}}) > 0 \text{)}\\
&= \max_{g_\phi} \overline{I_{\mathrm{CPC}}}(\hat{h}, g_\phi) && \text{(Assumption that } \hat{g} \text{ is optimal critic)}
\end{aligned}$$

which proves the theorem via contradiction.

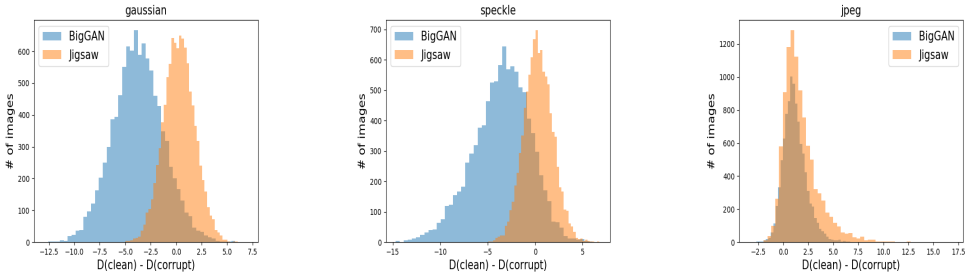
5.10 What does the theory over GANs entail?

Our goal is to show that NDA GAN objectives are principled in the sense that with infinite computation, data, and modeling capacity, NDA GAN will recover the same optimal generator as a regular GAN. In other words, under these assumptions, NDA will not bias the solution in an undesirable way. We note that the NDA GAN objective is as stable as regular GAN in practice since both methods estimate a lower bound to the divergence

with the discriminator, and then minimize that lower bound w.r.t. the generator. The estimated divergences are slightly different, but they have the same minimizer (which is the ground truth data distribution). Intuitively, while GAN and NDA GAN will give the same solution asymptotically, NDA GAN might get there faster (with less data) because it leverages a stronger prior over what the support should (not) be.

5.11 Anomaly Detection

Here, we show the histogram of difference in discriminator's output for clean and OOD samples in Figure 5.8. High difference values imply that the Jigsaw NDA is better at distinguishing OOD samples than the normal BigGAN.



(a) Gaussian Noise (b) Speckle Noise (c) JPEG Compression

Figure 5.8: Histogram of D(clean) - D(corrupt) for 3 different corruptions.

5.12 Effect of hyperparameter on Unconditional Image generation

Here, we show the effect of λ for unconditional image generation on CIFAR-10 dataset.

Table 5.8: Effect of λ on the FID score for unconditional image generation on CIFAR-10 using Jigsaw as NDA.

λ	1.0	0.75	0.5	0.25	0.15
FID	18.64	16.61	14.95	**12.61**	13.01

5.13 Dataset Preparation for FID evaluation

For dataset preparation, we follow the the following procedures: (a) CIFAR-10 contains 60K 32×32 images with 10 labels, out of which 50K are used for training and 10K are used for testing, (b) CIFAR-100 contains 60K 32 × 32 images with 100 labels, out of which 50K are used for training and 10K are used for testing, (c) CelebA contains 162,770 train images and 19,962 test images (we resize the images to 64×64px), (d) STL-10

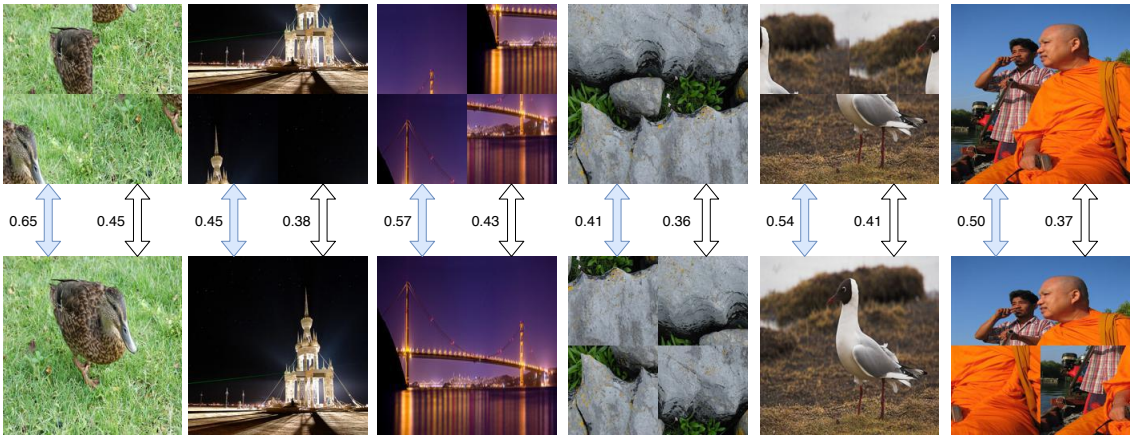


Figure 5.9: Comparing the cosine distance of the representations learned with Jigsaw NDA and Moco-V2 (**shaded blue**), and original Moco-V2 (**white**). With NDA, we project normal and its jigsaw image representations further away from each other than the one without NDA.

contains 100K (unlabeled) train images and 8K (labeled) test images (we resize the images to 32×32px). In our experiments the FID is calculated on the test dataset. In particular, we use 10K generated images vs. 10K test images for CIFAR-10, 10K vs. 10K for CIFAR-100, 19,962 vs. 19,962 for CelebA, and 8K vs 8K for STL-10.

5.14 Hyperparameters and Network Architecture

Generative Modeling. We use the same network architecture in BigGAN [138] for our experiments. The code used for our experiments is based over the author's PyTorch code. For CIFAR-10, CIFAR-100, and CelebA we train for 500 epochs whereas for STL-10 we train for 300 epochs. For all the datasets we use the following hyperparameters: batch-size = 64, generator learning rate = 2e-4, discriminator learning rate = 2e-4, discriminator update steps per generator update step = 4. The best model was selected on the basis of FID scores on the test set (as explained above).

Momentum Contrastive Learning. We use the official PyTorch implementation for our experiments. For CIFAR-10 and CIFAR-100, we perform unsupervised pre-training for 1000 epochs and supervised training (linear classifier) for 100 epochs. For Imagenet-100, we perform unsupervised pre-training for 200 epochs and supervised training (linear classifier) for 100 epochs. For CIFAR-10 and CIFAR-100, we use the following hyperparameters during pre-training: batch-size = 256, learning-date = 0.3, temperature = 0.07, feature dimensionality = 2048. For ImageNet-100 and fMoW pre-training we have the following: batch-size = 128, learning-date = 0.015, temperature = 0.2, feature dimensionality = 128. During linear classification we use a batch size of 256 for all the datasets and learning rate of 10 for CIFAR-10, CIFAR-100, whereas for ImageNet-100 and fMoW we use learning rate of 30 and 1 respectively.

Dense Predictive Coding. We use the same network architecture and hyper-parameters in DPC [127] for our experiments and use the official PyTorch implementation. We perform self-supervised training on UCF-101 for 200 epochs and supervised training (action classifier) for 200 epochs on both UCF-101 and HMDB51 datasets.

5.15 Implementation Details

For our experiment over GAN, we augment the batch of real samples with a negative augmentation of the same batch, and we treat the augmented images as fake images for the discriminator. Similarly, for the contrastive learning experiments, we consider negative augmentation of the query image batch as negatives for that batch.

For all our experiments we used existing open-source models. For experiments over GAN, we use the open-source implementations of BigGAN and Pix2Pix models, and for contrastive learning, we use the open-source implementation of the MoCo-v2 model and Dense Predictive Coding. Hence, we did not explain in detail each of the models. Implementing NDA is quite simple as we only need to generate NDA samples from the images in a mini-batch which only takes several lines of code.

5.16 Does the gain of NDA for representation learning come from the fact that more negative samples are used?

We perform the experiments over MoCo-v2 which maintains a queue of negative samples. The number of negatives is around 65,536. With our approach, we use the augmented versions of images in the same batch as negative. We transform both the key and query images to create NDA samples. Thus, the number of negatives for our approach is 65,536 + 2 (one NDA sample created using query image and other using key image), only 0.00003051664 times more than the original number of negatives samples in MoCo-v2. Thus our experiments are comparable to the baseline MoCo-v2. In terms of computation, we need an additional forward pass in each batch to get the representations of the NDA samples. The normal MoCo-v2 requires 1.09 secs for entire forward computation, which includes forward pass through the network, momentum update of the key encoder and dot product between the positive and negative samples. With NDA, 1 forward computation requires 1.36 secs.

5.17 What happens when negative data augmentations are noisy?

Regarding the performance of negative data augmentation, we perform 2 different experiments:

a) When the noise is low - When using jigsaw as our NDA strategy with a 2 x 2 grid, one out of the 24 permutations will be the original image. We find that when this special permutation is not removed, or there

is 4% "noise", the FID score is 12.61, but when it is removed the FID score is 12.59. So, we find that when the noise is low, the performance of our approach is not greatly affected and is robust in such scenarios.

b) When the noise is large - We use random vertical flipping as our NDA strategy, where with 50% probability the image is vertically flipped during NDA. In this case, the "noise" is large, as 50% of the time, the negative sample is actually the original image. We contrast this with the "noise-free" NDA strategy where the NDA image is always vertically flipped. We find that for the random vertical flipping NDA, the FID score of BigGAN is 15.84, whereas, with vertical flipping NDA, the FID score of BigGAN is 14.74. So performance degrades with larger amounts of noise.

5.18 Related work

In several machine learning settings, negative samples are produced from a statistical generative model. [54] aim to generate negative data using GANs for semi-supervised learning and novelty detection while we are concerned with efficiently creating negative data to improve generative models and self-supervised representation learning. [55] also propose an alternative theoretical framework that relies on access to an oracle which classifies a sample as valid or not, but do not provide any practical implementation. [56] use adversarial training to generate hard negatives that fool the discriminator for NLP tasks whereas we obtain NDA data from positive data to improve image generation and representation learning. [57] use a GAN to learn the negative data distribution with the aim of classifying positive-unlabeled (PU) data whereas we do not have access to a mixture data but rather generate negatives by transforming the positive data.

In contrastive unsupervised learning, common negative examples are ones that are assumed to be further than the positive samples semantically. Word2Vec [58] considers negative samples to be ones from a different context and CPC-based methods [59] such as momentum contrast [3], the negative samples are data augmentations from a different image. Our work considers a new aspect of "negative samples" that are neither generated from some model, nor samples from the data distribution. Instead, by applying negative data augmentation (NDA) to existing samples, we are able to incorporate useful inductive biases that might be difficult to capture otherwise [60].

5.19 Conclusion

We proposed negative data augmentation as a method to incorporate prior knowledge through out-of-distribution (OOD) samples. NDAs are complementary to traditional data augmentation strategies, which are typically focused on in-distribution samples. Using the NDA framework, we interpret existing image transformations (e.g., jigsaw) as producing OOD samples and develop new learning algorithms to leverage them. Owing to rigorous mathematical characterization of the NDA assumption, we are able to theoretically analyze their properties. As an example, we bias the generator of a GAN to avoid the support of negative samples, improving results on conditional/unconditional image generation tasks. Finally, we leverage NDA for unsupervised

representation learning in images and videos. By integrating NDA into MoCo-v2 and DPC, we improve results on image and action recognition on CIFAR10, CIFAR100, ImageNet-100, fMoW, UCF-101, and HMDB-51 datasets. Results on satellite imagery dataset, fMoW, show that our method is appropriately positioned for sustainability related applications which have large amount unlabeled satellite imagery at their dispense for various tasks like poverty estimation, infrastructure management, etc. Future work include exploring other augmentation strategies as well as NDAs for other modalities like text, speech, etc.

Chapter 6

Efficient Conditional Pre-training for Transfer Learning

6.1 Introduction

Recent success of many modern computer vision methods relies heavily on large-scale labeled datasets, which are often costly and time-consuming to collect [110, 3, 5]. Alternatives to large-scale labelled data include pre-training a network on the publicly available ImageNet dataset with labels [78]. It has been shown that ImageNet features can transfer well to many different target tasks [61, 62, 63, 64, 65]. Another alternative, unsupervised learning, has received tremendous attention recently with the availability of extremely large-scale data with no labels, as such data is costly to obtain [110]. It has been shown that recent unsupervised learning methods, e.g. contrastive learning, can perform on par with their supervised learning counterparts [3, 2, 111, 150, 33, 5] and even better in certain settings [3, 2, 5].

The explosion of data quantity and improvement of unsupervised learning portends that the standard approach in future tasks will be to (1) learn weights a on a very large-scale dataset with unsupervised learning and (2) fine-tune the weights on a small-scale target dataset. A major problem with this approach is the large amount of computational resources required to train a network on a very large scale dataset [110]. For example, a recent contrastive learning method, MoCo-v2 [2, 3], uses 8 Nvidia-V100 GPUs to train on ImageNet-1k for 53 hours, which can cost thousands of dollars. Extrapolating, this forebodes pre-training costs on the order of millions of dollars when considering much larger-scale datasets. Those without access to such resources will require selecting relevant subsets of those datasets. However, other studies that perform conditional filtering, such as [66, 67, 68, 69], do not consider efficiency.

Cognizant of these pressing issues, we propose novel methods to efficiently *filter* a user defined number of pre-training images conditioned on a target dataset as well as a novel sequential pre-training method for our methods to work efficiently in practical settings with several target tasks. We also investigate the use

of low resolution images for pre-training, which we find provides a great cost to performance trade-off. Our approach consistently outperforms other methods by 2-9% and are both flexible, translating to both supervised and unsupervised settings, and adaptable, translating to a wide range of target tasks including image recognition, object detection and semantic segmentation. Due to our focus on filtering based on image features, not labels, our methods perform especially well in the more relevant unsupervised setting, where pre-training on a 12% subset of data can achieve within 1-4% of full pre-training target task performance. Additionally, we use our methods to tune ImageNet pre-trained models and filter from larger scale data to improve on standard ImageNet pre-training by 1-3% on downstream tasks. Given these results and the exponentially growing scale of unlabeled data, our methods can replace the standard ImageNet pre-training with a target task specific efficient conditional pre-training.

Our method is appropriately positioned for sustainability related applications which have large amount unlabeled satellite imagery at their dispense for various tasks like poverty estimation, infrastructure management, etc. Our methods can replace pre-training on full labeled or unlabeled dataset with a target task specific efficient conditional pre-training without sacrificing performance.

6.2 Related Work

Active Learning Active Learning fits a function by selectively querying labels for samples where the function is currently uncertain. In a basic greedy setup, the samples with the highest entropies are chosen for annotation [151, 152, 153, 154]. The model is iteratively updated with these samples and accordingly selects new samples. Active learning typically assumes similar data distributions for candidate samples, whereas our data distributions can potentially have large shifts. Furthermore, active learning, due to its iterative nature, can be quite costly, hard to tune, and can require prior distributions [155].

Unconditional Transfer Learning The success of deep learning on datasets with increased sample complexity has brought transfer learning to the attention of the research community. Pre-training networks on ImageNet-1k has been shown to be a very effective way of initializing weights for a target task with small sample size [61, 62, 63, 64, 65]. However, all these studies use unconditional pre-training as they employ the weights pre-trained on the full source dataset, which can be computationally infeasible for future large scale datasets.

Conditional Transfer Learning [66, 67, 68], on the other hand, filter the pre-training dataset conditioned on target tasks. [67, 69] use greedy class-specific clustering based and learn image representations with an encoder trained on the massive JFT-300M dataset [70], which dramatically increases cost. [66] trains a number of expert models on many subsets of the pre-training dataset and uses their performance to weight source images, however this method is naturally quite computationally expensive. Our methods differ from the past works as we take into account pre-training dataset filtering efficiency, adaptability to different tasks and settings, and target task performance.

6.3 Problem Definition and Setup

We assume a target task dataset represented as $\mathcal{D}_t = (\mathcal{X}_t, \mathcal{Y}_t)$ where $\mathcal{X}_t = \{x_t^1, x_t^2, \ldots, x_t^M\}$ represents a set of M images with their ground truth labels $\mathcal{Y}_t$. Our goal is to train a function f_t parameterized by θ_t on the dataset $\mathcal{D}_t$ to learn $f_t : x_t^i \mapsto y_t^i$. To transfer learn, we first pre-train θ_t on a large-scale source dataset $\mathcal{D}_s$ and fine-tune θ_t on $\mathcal{D}_t$. This strategy reduces the amount of labeled samples needed in $\mathcal{D}_t$ and boosts the accuracy in comparison to the randomly initialized weights [110, 44]. For the pre-training dataset, we can have either labelled or unlabelled setups: (1) $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s)$ and (2) $\mathcal{D}_s = (\mathcal{X}_s)$ where $\mathcal{X}_s = \{x_s^1, x_s^2, \ldots, x_s^N\}$. The most common example of the labelled setup is the ImageNet dataset [78]. However, it is tough to label vast amounts of publicly available images, and with the increasing popularity of unsupervised learning methods [5, 156, 33, 3, 2], it is easy to see that unsupervised pre-training on very large $\mathcal{D}_s$ with no ground-truth labels will be the standard and preferred practice in the future.

A major problem with learning θ_t on a very large-scale dataset $\mathcal{D}_s$ is the computational cost, and using the whole dataset may be impossible for most. One way to reduce costs is to filter out images deemed less relevant for $\mathcal{D}_t$ to create a dataset $\mathcal{D}_s^{'} \in \mathcal{D}_s$ where $\mathcal{X}_s = \{x_s^1, x_s^2, \ldots, x_s^{N^{'}}\}$ represents a filtered version of $\mathcal{D}_s$ with $N^{'} \ll N$. Our approach conditions the filtering step on the target dataset $\mathcal{D}_s$. In this study, we propose flexible and adaptable methods to perform *efficient conditional pre-training*, which reduces the computational costs of pre-training and maintains high performance on the target task.

6.4 Methods

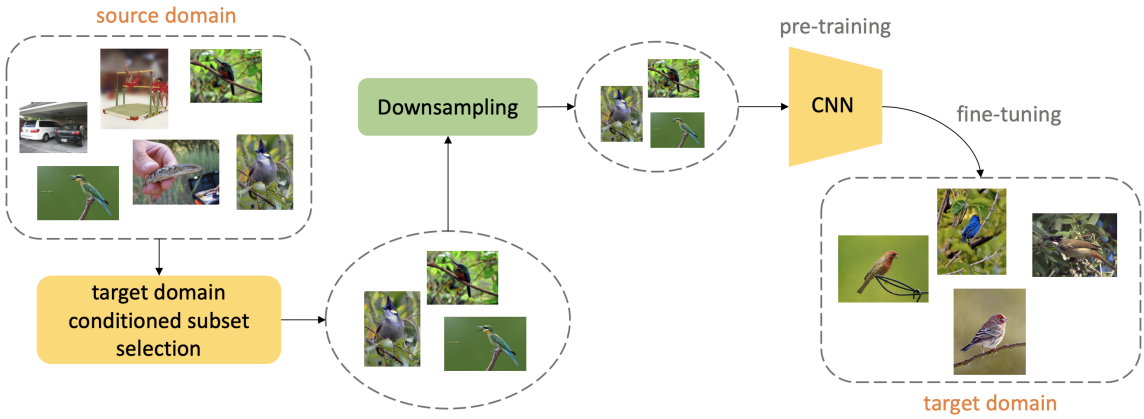


Figure 6.1: Schematic overview of our approach. We first perform a conditional filtering method on the source dataset and downsample image resolution on this filtered subset. Finally, we perform pre-training on the subset and finetuning on the target task.

We investigate a variety of methods to perform efficient pre-training while maintaining high performance on the target dataset. We visualize our overall procedure in Figure 6.1 and explain our techniques below.

6.4.1 Conditional Data Filtering

We propose novel methods to perform conditional filtering efficiently. Our methods score every image in the source domain and select the best scoring images according to a pre-specified data budget N'. Our methods are fast, requiring at most one forward pass through $\mathcal{D}_s$ to get the filtered dataset $\mathcal{D}_s^{'}$ and can work on both $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s)$ and $\mathcal{D}_s = (\mathcal{X}_s)$. The fact that we consider *data features not labels* perfectly lends our methods to the more relevant unsupervised setting. This is in contrast to previous work such as [67, 69, 68] which do not consider efficiency and are designed primarily for the supervised setting and thus will be more difficult for most to apply to large scale datasets.

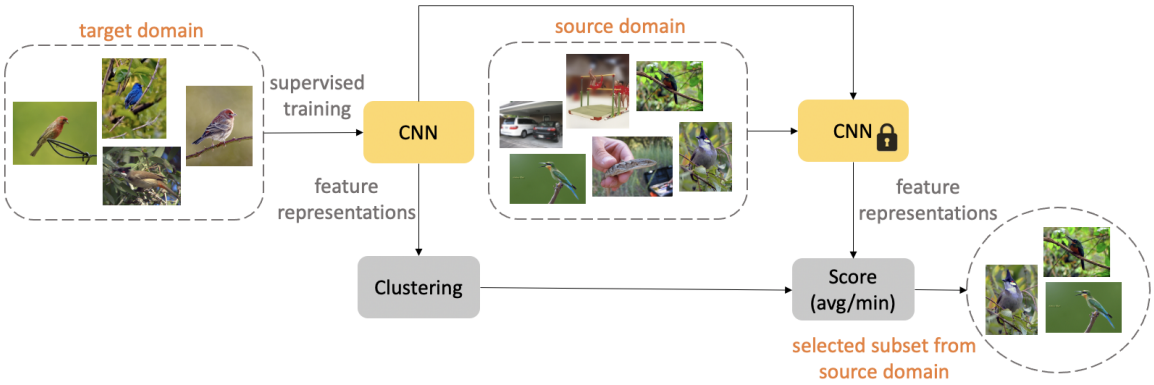


Figure 6.2: Schematic overview of clustering based filtering. We first train a model on the target domain to extract representations, which we use to cluster the target domain. We score source images with either average or min distance to cluster centers and then filter.

Algorithm 3 Clustering Based Filtering

procedure CLUSTERFILTER($\mathcal{D}_s, \mathcal{D}_t, N^{'}, K, AggOp$)
  $f_h \leftarrow TRAIN(\mathcal{D}_t)$ ▷ Train Feature Extractor
  $\mathcal{Z}_t \leftarrow \{f_h(x_t^i)\}_{i=1}^{M}$ ▷ Target Representations
  $\{\hat{z}\}_{k=1}^{K} \leftarrow K\text{-}Means(\mathcal{Z}_t, K)$ ▷ Cluster Target
  $d_k^i \leftarrow ||f_h(x_s^i) - \hat{z}_k||_2$ ▷ Source Distances
  $c_s \leftarrow \{AggOp(\{d_k^i\}_{k=1}^{K})\}_{i=1}^{N}$ ▷ Score Source
  $\mathcal{D}_s^{'} \leftarrow BOTTOM(\mathcal{D}_s, N^{'}, c_s)$ ▷ Filter Source
  return $\mathcal{D}_s^{'}$ ▷ Return the Filtered Subset

Conditional Filtering by Clustering

Selecting an appropriate subset $\mathcal{D}_s^{'}$ of pre-training data $\mathcal{D}_s$ can be viewed as selecting a set of data that minimizes some distance metric between $\mathcal{D}_s^{'}$ and the target dataset $\mathcal{D}_t$, as explored in [67, 69]. This is accomplished by taking feature representations $\mathcal{Z}_s$ of the set of images $\mathcal{X}_s$ and selecting pre-training image classes which are close (by some distance metric) to the representations of the target dataset classes. Building

on this, we make several significant modifications to account for our goals of efficiency and application to unsupervised settings.

Training Only with Target Data. We do not train a network f_h on a large scale dataset, i.e. JFT-300M [67], as this defeats the entire goal of pre-training efficiency. Therefore, we first train a model f_h with parameters θ_h using the target dataset $\mathcal{D}_t = (\mathcal{X}_t, \mathcal{Y}_t)$ and use the learned θ_h to filter the source dataset $\mathcal{D}_s$.

Consider Source Images Individually. Selecting entire classes of pre-training data can be suboptimal when limited to selecting a small subset of the data. For example, if limited to 6% of ImageNet, (a reasonable budget for massive datasets), we can only select 75 of the 1000 classes, which may prohibit the model from having the breadth of data needed to learn transferrable features. Instead, we treat each image x_s^i from $\mathcal{D}_s$ separately to flexibly over-represent relevant classes while not being forced to select entire classes. Additionally, very large scale datasets may not have class labels $\mathcal{Y}_s$. For this reason, we want to develop methods that work with unsupervised learning, and treating source images independently accomplishes this.

Scoring and Filtering. Finally, we choose to perform K-Means clustering on the representations $\mathcal{Z}_t$ learned by f_h to get K cluster centers $\{\hat{z}\}_{k=1}^K$. We then compute the distances between $\mathcal{X}_s$ and $\{\hat{z}\}_{k=1}^K$ as

$$d_k^i(x_s^i, k) = ||f_h(x_s^i; \theta_h) - \hat{z}_k||_p \tag{6.1}$$

where p is typically 1 or 2 (L1 or L2 distance). We can score x_s^i by considering an *Aggregation Operator*(AggOp) of either average distance to the cluster centers

$$c_s^i = \frac{1}{K} \sum_{k=1}^{K} d_k^i \tag{6.2}$$

or minimum distance

$$c_s^i = \min(\{d_k^i\}_{k=1}^K). \tag{6.3}$$

To filter, we sort by c_s^i in ascending order and select N' images to create $\mathcal{D}_s^{'} \in \mathcal{D}_s$ and pre-train θ_t on it.

Advantages of our Method Performing unsupervised clustering ensures that our method is not fundamentally limited to image recognition target tasks and also does not assume that source dataset images in the same class should be grouped together. Furthermore, our method requires only a relatively cheap single forward pass through the pre-training dataset. It attains our goals of efficiency and flexibility, in contrast to prior work such as [69, 67]. We outline the algorithm step-by-step in Algorithm 3 and lay out the method visually in Figure 6.2.

Conditional Filtering with Domain Classifier

In this section, we propose a novel domain classifier to filter $\mathcal{D}_s$ with several desirable attributes. We outline the algorithm step-by-step in Algorithm 4 and provide a depiction in Figure 6.3.

Training. In this method, we propose to learn θ_h to ascertain whether an image belongs to $\mathcal{D}_s$ or $\mathcal{D}_t$. θ_h is learned on a third dataset $\mathcal{D}_h = (\mathcal{X}_h, \mathcal{Y}_h)$ where $\mathcal{X}_h = \{\{x_s^i\}_{i=1}^M, \{x_t^i\}_{i=1}^M\}, M = |\mathcal{D}_t|$, consisting of full

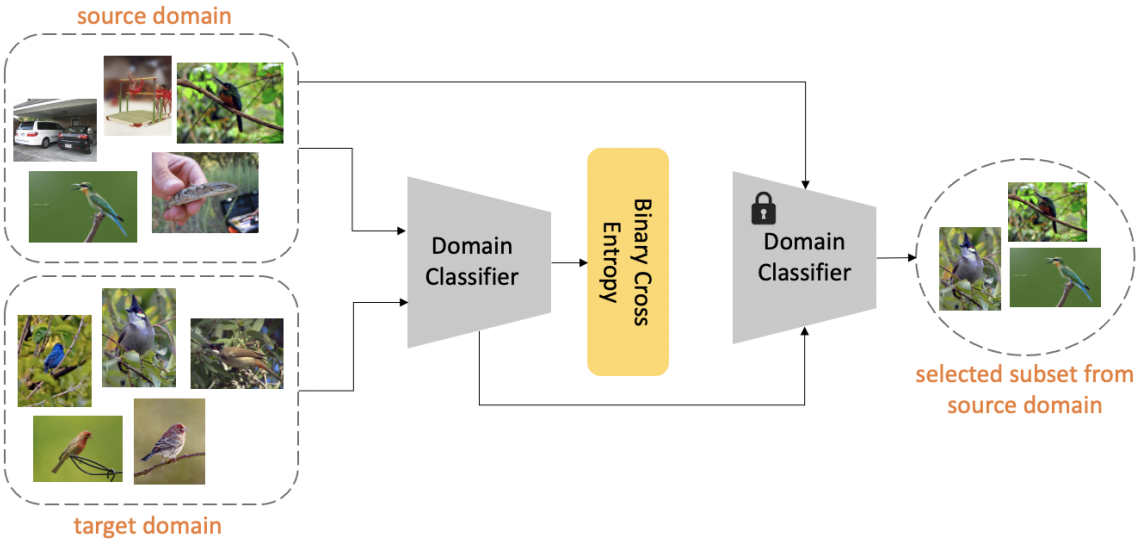


Figure 6.3: Depiction of the Domain Classifier. We train a simple binary classifier to discriminate between source and target domain and then use the output probabilities on source images to filter.

Algorithm 4 Domain Classifier Filtering

procedure DOMAINCLSFILTER($\mathcal{D}_s, \mathcal{D}_t, N^{'}$)
  SAMPLE $\{x_s^i\}_{i=1}^{M} \in \mathcal{D}_s$
  $\mathcal{X}_h \leftarrow \{\{x_s^i\}_{i=1}^{M}, \{x_t^i\}_{i=1}^{M}\}$
  $\mathcal{Y}_h \leftarrow \{\{0\}_{i=1}^{M}, \{1\}_{i=1}^{M}\}$ ▷ Domain Labels
  $\mathcal{D}_h \leftarrow (\mathcal{X}_h, \mathcal{Y}_h)$ ▷ Training Data
  $f_h(x;\theta_h) \leftarrow \text{argmin}_{\theta_h}\, CELoss(\mathcal{D}_h)$ ▷ Fit Model
  $c_s \leftarrow \{f_h(x_s^i;\theta_h)\}_{i=1}^{N}$ ▷ Score
  $\mathcal{D}_s^{'} \leftarrow TOP(\mathcal{D}_s, N^{'}, c_s)$ ▷ Filter Source
  return $\mathcal{D}_s^{'}$ ▷ Return the Filtered Subset

set of $\mathcal{D}_t$ and a small random subset of $\mathcal{D}_s$. Each source image $x_s^i \in \mathcal{X}_s^{'}$ receives a negative label and each target image $x_t^i \in \mathcal{X}_t^{'}$ receives a positive label giving us the label set $\mathcal{Y}_h = \{\{0\}_{i=1}^{M}, \{1\}_{i=1}^{M}\}$. We then learn θ_h on $\mathcal{D}_h$ using cross entropy loss as

$$\underset{\theta_h}{\text{argmin}} \sum_{i=1}^{2M} y_h^i log(f_h(x_h^i;\theta_h)) + (1-y_h^i) log(1-f_h(x_h^i;\theta_h)). \tag{6.4}$$

Scoring and Filtering. Once we learn θ_h we obtain the confidence score $p(y_h = 1|x_s^i;\theta_h)$ for each image $x_s^i \in \mathcal{X}_s$. We then sort the source images $\mathcal{X}_s$ in descending order based on $p(y_h = 1|x_s^i;\theta_h)$ and choose the top N' images to create the subset $\mathcal{D}_s^{'} \in \mathcal{D}_s$.

Interpretation. Our method can be interpreted as selecting images from the pre-training domain with high probability of belonging to the target domain. It can be shown [157] that the Bayes Optimal binary classifier $\hat{f}_h$ assigns probability

$$p(y_h = 1|x_s^i; \theta_h) = \frac{p_t(x_s^i)}{p_s(x_s^i) + p_t(x_s^i)} \tag{6.5}$$

for an image $x_s^i \in \mathcal{X}_s$ to belong to the target domain, where p_t and p_s are the true data probability distributions for the target and source domains respectively.

6.4.2 Sequential Pre-training

The methods we present work efficiently for a single target task. However, in practice, we may be interested in many different target tasks, and performing separate pre-training from scratch for each one may be prohibitively inefficient. As a result, we propose performing sequential pre-training, where we leverage previously trained models to more quickly learn better transfer learning representations.

Formally, we assume that we have a large scale source dataset S (which can potentially grow over time) and want to perform tasks on N target datasets, which we receive sequentially over time as $((S, D_1, t_1), (S, D_2, t_2), \ldots, (S, D_N, t_N))$. We receive our first task with dataset D_1 at time t_1, and we conditionally filter S into S_1' based on our data budget. Then, we pre-train a model, f_1, from scratch on S_1', and perform task one. Generally, when we receive D_i at time t_i, we filter S conditioned on D_i to obtain S_i'. Then, we take our last pre-trained model f_{i-1} and update its weights on S_i' to obtain f_i, which we separately use to perform the task on D_i. Subsequent tasks require smaller and smaller amounts of additional pre-training, thus drastically reducing the total epochs required for multiple tasks and making our methods feasible in practical settings.

6.4.3 Adjusting Pre-training Spatial Resolution

To augment our methods, we propose changing spatial resolution of images $\mathcal{X}_s$ in the source dataset $\mathcal{D}_s$ while pre-training. We assume that an image is represented as $x_s^i \in \mathbb{R}^{W_s \times H_s}$ or $x_t^i \in \mathbb{R}^{W_t \times H_t}$ where W_s, W_t, where H_s and H_t represent image width and height. Traditionally, after augmentations, we use $W_s, W_t = 224$ and $H_s, H_t = 224$. Here, we consider decreasing W_s and H_s on the pre-training task while maintaining $W_t, H_t = 224$ on the target task. Reducing image resolution while pre-training can provide significant speedups by decreasing FLOPs required by convolution operations, and our experiments show that downsizing image resolution by half $W_s, H_s = 112$ almost halves the pre-training time.

Training on downsized images and testing on higher resolution images due to geometric camera effects on standard augmentations has previously been explored [158]. Our setting is not as amenable to the same analysis, as we have separate data distributions $\mathcal{D}_s$ and $\mathcal{D}_t$ captured under different settings. Nevertheless, we show low resolution training is still an effective method in the transfer learning setting.

6.5 Experiments

In our experiments, we report finetuning performance for combinations of resolution, pre-training budget, and filtering method as well as performance with full pre-training and no pre-training for reference. We refer

the reader to a later section for specific details on experimental setup.

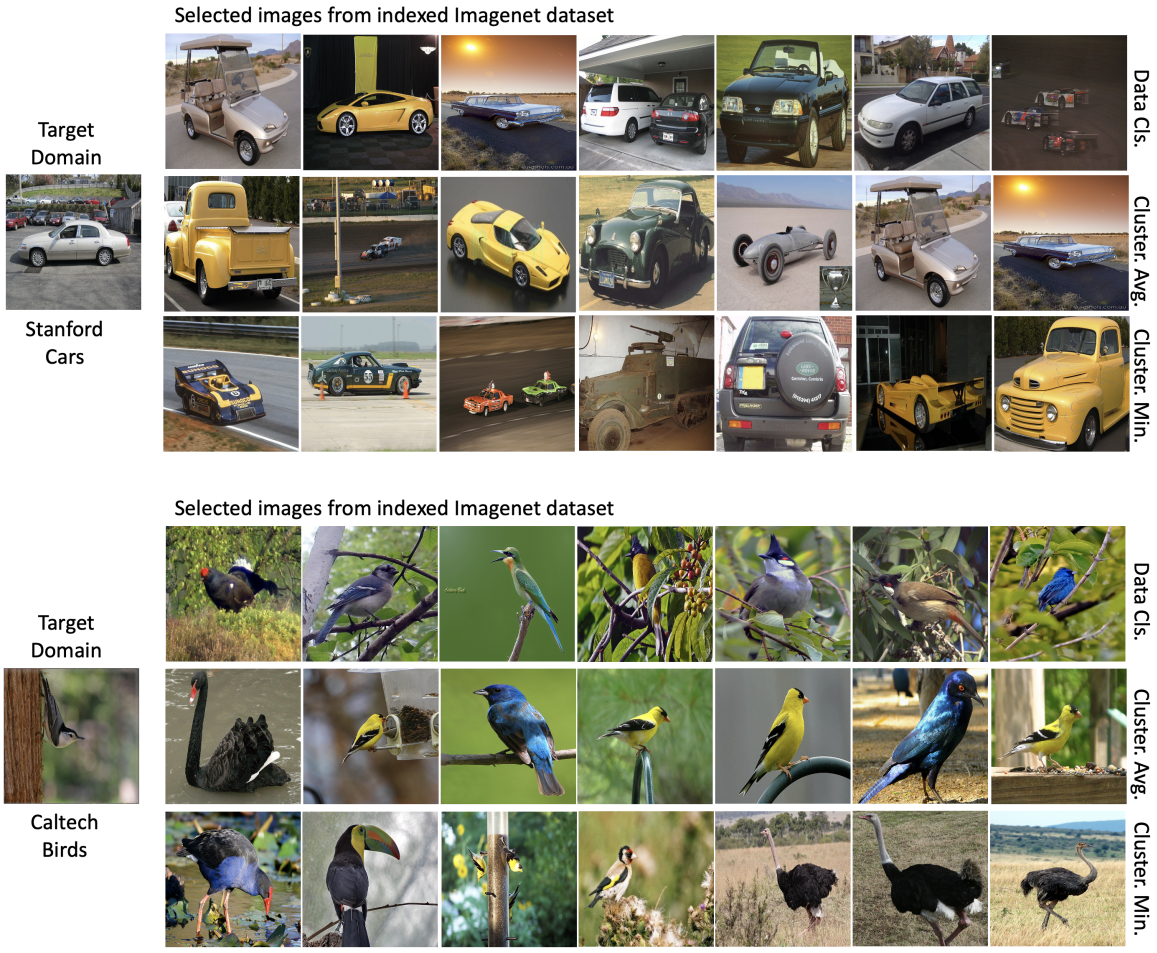


Figure 6.4: High scoring ImageNet samples selected by all our conditional filtering methods for target datasets Stanford Cars and Caltech Birds.

6.5.1 Datasets

Source Dataset

For our primary source dataset, we utilize ImageNet-2012 [78], with ∼1.28M images over 1000 classes. We experiment under two data budgets, limiting filtered subsets to 75K (∼6%) and 150K (∼12%) ImageNet images. This is an appropriate proportion when dealing with pre-training datasets on the scale of tens of millions or more images.

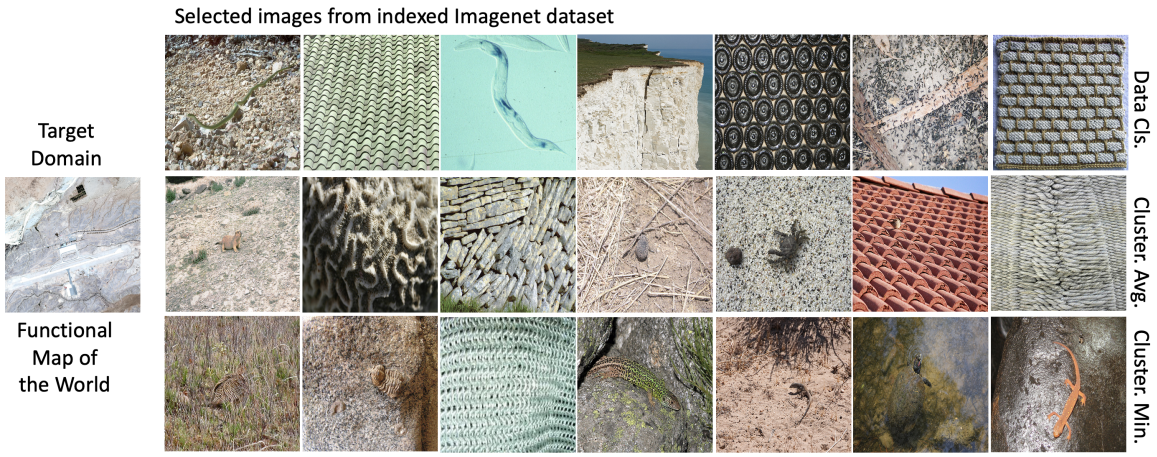


Figure 6.5: High scoring ImageNet samples selected by all our conditional filtering methods for fMoW.

Target Datasets

Image Recognition As target datasets, we utilize the Stanford Cars [159] dataset, the Caltech Birds [160] dataset, and a subset of the Functional Map of the World [112] (fMoW) dataset. We provide basic details about these datasets in a later section. These datasets have different degrees of variation and images per class and lend important diversity to validate the flexibility of our methods. Cars has a fairly small distribution shift from ImageNet, and pre-training on ImageNet performs well on it, but Birds contains a larger shift and datasets emphasizing natural settings such as iNat perform better [67, 161]. Finally, fMoW, consisting of overhead satellite images, contains images very dissimilar to ImageNet. Additionally, Birds and Cars are fine grained tasks, discriminating between different species of birds or models of cars, respectively. In contrast, fMoW is much more general, describing buildings or landmarks.

Supervised Pre-train.		Target Dataset			Cost
224 x 224		Small Shift		Large Shift	(hrs)
Pre-train. Sel.	Method	Cars	Birds	fMow	
0%	**Random Init.**	52.89	42.17	43.35	0
100%	**Entire Dataset**	82.63	74.87	59.05	160-180
6%	**Random**	72.2	57.87	50.25	30-35
	Domain Cls.	**74.37**	**59.73**	**51.17**	35-40
	Clustering (Avg)	73.64	56.33	**51.14**	40-45
	Clustering (Min)	**74.23**	57.67	50.27	40-45
12%	**Random**	76.12	62.73	**53.28**	45-50
	Domain Cls.	76.18	**64**	**53.41**	50-55
	Clustering (Avg)	**77.12**	61.73	53.12	55-60
	Clustering (Min)	75.81	**64.07**	52.91	55-60

Supervised Pre-train.		Target Dataset			Cost
112 x 112		Small Shift		Large Shift	(hrs)
Pre-train. Sel.	Method	Cars	Birds	fMow	
0%	**Random Init**	52.89	42.17	43.35	0
100%	**Entire Dataset**	83.78	73.47	57.39	90-110
6%	**Random**	72.76	57.4	49.73	15-20
	Domain Cls.	73.66	**58.73**	50.66	20-25
	Clustering (Avg)	**74.53**	56.97	**51.32**	25-30
	Clustering (Min)	71.72	**58.73**	49.06	25-30
12%	**Random**	75.4	62.63	52.59	30-35
	Domain Cls.	76.36	**63.5**	**53.37**	35-40
	Clustering (Avg)	**77.53**	61.23	52.67	40-45
	Clustering (Min)	76.36	63.13	51.6	40-45

Table 6.1: Target task accuracy and approximate filtering and pre-training cost(time in hrs on 1 GPU) on 3 visual categorization datasets obtained by pre-training on different subsets of the source dataset (ImageNet) with different filtering methods at different resolutions.

Detection and Segmentation [3, 2] show that unsupervised ImageNet pre-training is most effective when paired with more challenging low level downstream tasks. Therefore, we also perform experiments in the object detection and semantic segmentation setting to validate the flexibility and adaptability of our methods. To this end, we utilize the Pascal VOC [117] dataset with unsupervised ImageNet pre-training of the backbone.

6.5.2 Analyzing Filtering Methods

Here, we make some important points about our filtering methods and refer the reader to a later section for specific implementation details.

Domain Classifier Accuracy We typically train the domain classifier to 92-95% accuracy. We empirically find this is the "sweet spot" as classifiers with 88-90% accuracy, perhaps due to not learning relevant features, and 98+% accuracy, perhaps due to over-discriminating minor differences between domains such as noise or color/contrast, do not perform as well.

Efficiency and Adaptability Comparison. The domain classifier trains a simple binary classifier and bypasses full representation learning on a target dataset, computing distances, or clustering. However, this difference in efficiency is small compared to pre-training cost. More importantly, when the target task is not image level classification, the representation learning step for clustering based filtering must be modified in a non-trivial manner. This can involve a global pool over spatial feature maps while performing object detection or an entirely different setup like unsupervised learning. The domain classifier is more adaptable than clustering as it does not require modification for any type of target task.

Qualitative Analysis. In Figures 6.4 and 6.5, we visualize some of the highest scoring filtered images for all our methods on classification tasks and verify that our filtering methods do select images with *relevant features* to the target task. Unsurprisingly, more interpretable images are selected for Birds and Cars, as there are no satellite images in ImageNet. Nevertheless, we see that the selected images for fMoW stil contain *relevant features* such as color, texture, and shapes.

6.5.3 Transfer Learning for Image Recognition

We first apply our methods to the task of image classification with both supervised and unsupervised pre-training. We detail our results below.

Supervised Pre-training Results

We present target task accuracy for all our methods on Cars, Birds, and fMoW along with approximate pre-training and filtering time in Table 6.1.

Effect of Image Resolution. We see that downsizing pre-training resolution produces gains of up to .5% in classification accuracy on Cars and less than 1% drop in accuracy on Birds and fMoW, while being 30-50% faster than full pre-training. These trends suggest that training on lower resolution images can help the model

MoCo-v2 [2]		Target Dataset			Cost (hrs)
224 x 224		Small Shift		Large Shift	
Pre-train. Sel. Method		Cars	Birds	fMow	
0%	**Random Init.**	52.89	42.17	43.35	0
100%	**Entire Dataset**	83.52	67.49	56.11	210-220
6%	**Random**	75.70	56.82	52.53	20-25
	Domain Cls.	78.67	**61.55**	52.96	23-28
	Clustering (Avg)	78.66	60.88	53.19	25-30
	Clustering (Min)	**79.45**	59.36	**53.5**	25-30
12%	**Random**	75.66	61.70	53.56	30-35
	Domain Cls.	78.68	63.08	54.01	33-38
	Clustering (Avg)	78.68	62.53	**54.4**	35-40
	Clustering (Min)	**79.55**	**63.6**	**54.26**	35-40

MoCo-v2 [2]		Target Dataset			Cost (hrs)
112 x 112		Small Shift		Large Shift	
Pre-train. Sel. Method		Cars	Birds	fMow	
0%	**Random Init**	52.89	42.17	43.35	0
100%	**Entire Dataset**	84.09	66.57	56.83	110-120
6%	**Random**	75.38	56.63	52.59	10-15
	Domain Cls.	76.84	57.93	53.3	13-18
	Clustering (Avg)	76.86	**58.4**	**53.75**	15-20
	Clustering (Min)	**77.53**	57.1	**53.83**	15-20
12%	**Random**	78.35	61.50	54.28	15-20
	Domain Cls.	**80.38**	**63.93**	54.53	18-23
	Clustering (Avg)	80.21	63.50	**55.06**	20-25
	Clustering (Min)	79.63	62.77	**55.03**	20-25

Table 6.2: Target task accuracy and approximate filtering and pre-training cost(time in hrs on 4 GPUs) on 3 visual categorization datasets obtained by pre-training on different subsets of the source dataset (ImageNet) with different filtering methods at different resolutions.

learn more generalizeable features for similar source and target distributions. This effect erodes slightly as we move out of distribution, however pre-training on lower resolution images offers an attractive trade-off between efficiency and accuracy in all settings.

Impact of Filtering. We find that our filtering techniques consistently provide up to a 2.5% performance increase over random selection, with a relatively small increase in cost. Unsurprisingly, filtering provides the most gains on Cars and Birds where the target dataset has a smaller shift. On fMoW, it is very hard to detect *similar* images to ImageNet, as the two distributions have very little overlap. Nevertheless, in this setting, our filtering methods can still select enough relevant features to provide a 1-2$\%$ boost.

Comparison of Filtering Methods. While all our methods perform well, applying a finer lens, we see that the domain classifier is less variable than clustering and always outperforms random selection. On the other hand, average clustering performs well on Cars or fMoW, but does worse than random on Birds and vice versa for min clustering. These methods rely on computing high dimensional vector distances to assign a measure of similarity, which may explain their volatility since such high dimensional distances are not considered in supervised pre-training.

Unsupervised Pre-training Results

We observe promising results in the supervised setting, but as explained, a more realistic and useful setting is the unsupervised setting due to the difficulties inherent in collecting labels for large-scale data. Thus, we use MoCo-v2 [2], a state-of-the-art unsupervised learning method, to pre-train on ImageNet and present results for Cars, Birds, and fMoW in Table 6.2.

Effect of Image Resolution. We find that in the unsupervised setting, with 150K pre-training images, lower resolution pre-training largely maintains or even improves performance as the target distribution shifts. Unsupervised pre-training relies more on high level features and thus may be better suited than supervised methods for lower resolution pre-training, since higher resolution images may be needed to infer fine grained

label boundaries.

Increased Consistency of Clustering. Relative to the supervised setting, clustering based filtering provides more consistent performance boosts across the different settings and datasets. It is possible that clustering based filtering may be well suited for unsupervised contrastive learning techniques, which also rely on high dimensional feature distances.

Impact of Filtering. Our filtering techniques aim to separate the image distributions based on the true image distributions and feature similarity, not label distribution (which may not be observable). Unsupervised learning naturally takes advantage of our filtering methods, and we see gains of up to 5% over random filtering in the 75K setting and up to 4% in the 150K setting, a larger boost than during supervised pre-training. This leads to performance that is within 1-4% of full unsupervised pre-training but close to 10 times faster, due to using a 12% subset. These results are notable, because, as mentioned, we anticipate that unsupervised learning will be the default method for large-scale pre-training and our methods can approach full pre-training while significantly reducing cost.

Sequential Pre-training

Cognizant of the inefficiencies of performing independent pre-training with many target tasks, we assume a practical scenario where we receive three tasks, D_1, D_2, D_3 representing Cars/Birds/fMoW respectively, with S being ImageNet. We use the domain classifier to filter 150K images, obtain S'_1, S'_2, S'_3, and sequentially pre-train for 100, 40, and 20 epochs respectively with MoCo-v2.

We present results in Figure 6.6. Naturally, for Cars the results do not change, but since learned features are leveraged, not discarded, for subsequent tasks, we observe gains of up to 1% on Birds and 2% on fMoW over Table 6.2 while using 160 total pre-training epochs vs 300 for independent pre-training. Our sequential pre-training method augments the effectiveness of our filtering methods in settings with many target tasks over time and drastically reduces the number of epochs required. We leave the application of this technique for object detection and segmentation as future work.

Detection		224x224			112x112		
Pre-train.	Sel. Method	AP	AP50	AP75	AP	AP50	AP75
0%	**Random Init.**	14.51	31.00	11.62	14.51	31.00	11.62
100%	**Entire Dataset**	43.94	73.05	45.96	43.62	72.56	45.52
6%	**Random**	29.01	54.02	27.26	28.10	52.82	26.39
	Domain Cls.	**30.47**	**56.58**	29.04	**31.19**	**56.90**	**30.43**
	Clustering (Avg)	**30.61**	55.65	28.75	30.13	55.01	29.47
	Clustering (Min)	**30.44**	56.11	**29.46**	30.39	55.89	28.18
12%	**Random**	30.84	52.07	29.15	30.56	56.1	29.04
	Domain Cls.	**34.41**	**61.85**	**33.36**	**34.98**	**61.83**	**35.02**
	Clustering (Avg)	32.34	56.24	31.28	32.01	57.16	33.48
	Clustering (Min)	32.58	57.77	31.16	32.96	58.25	33.64

Segmentation		224x224			112x112		
Pre-train.	Sel. Method	mIOU	mAcc	allAcc	mIOU	mAcc	allAcc
0%	**Random Init.**	0.45	0.55	0.82	0.45	0.55	0.82
100%	**Entire Dataset**	0.65	0.74	0.89	0.63	0.72	0.88
6%	**Random**	0.55	0.65	0.85	0.58	0.68	0.87
	Domain Cls.	**0.62**	**0.70**	**0.88**	**0.62**	**0.70**	**0.88**
	Clustering (Avg)	0.61	**0.70**	**0.88**	0.59	0.69	0.87
	Clustering (Min)	0.61	**0.70**	**0.88**	0.61	**0.70**	**0.88**
12%	**Random**	0.56	0.65	0.86	0.59	0.69	0.87
	Domain Cls.	**0.65**	**0.74**	**0.89**	**0.62**	**0.71**	**0.89**
	Clustering (Avg)	0.64	0.73	**0.89**	0.59	0.68	0.87
	Clustering (Min)	0.61	0.70	0.88	0.61	0.70	0.88

Table 6.3: Comparison of different filtering methods and resolutions on transfer learning on Pascal-VOC detection and segmentation. For object detection and semantic segmentation, we use unsupervised pre-training method MoCo-v2 [2] .

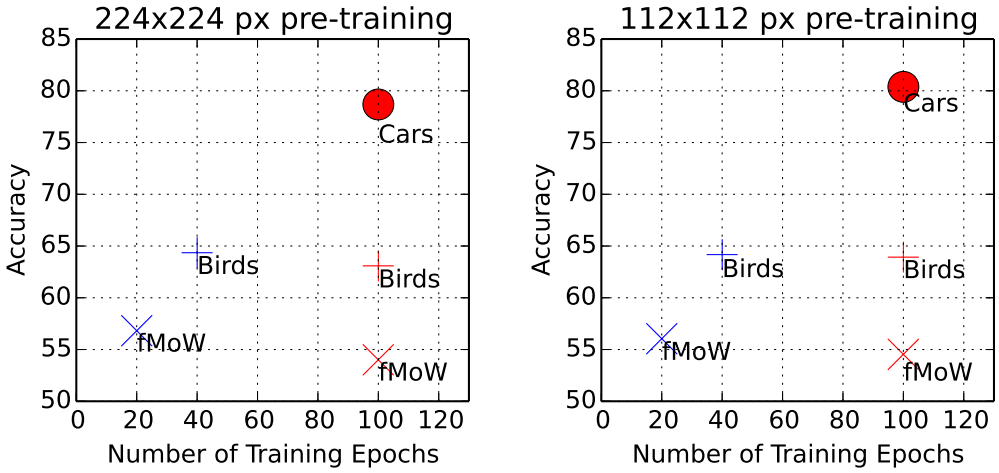


Figure 6.6: Results for sequential pre-training (blue) vs independent pre-training (red). Our sequential method requires fewer epochs over time and performs better than independent pre-training.

6.5.4 Transfer Learning for Low Level Tasks

Previously we explored image level classification target tasks for conditional pre-training. In this section, we perform experiments on transfer learning for object detection and semantic segmentation on the Pascal VOC dataset.

We present results in Table 6.3. For filtering, we use the domain classifier with no modifications and for clustering, we use MoCo-v2 on Pascal VOC to learn representations. We refer the reader to a later section for more experimental details.

Effect of Image Resolution. Overall, we see pre-training on low resolution images produces no overall decrease in performance, with the usual corresponding 30-50% reduction in training time, confirming the adaptability of pre-training on lower resolution images for more challenging lower level tasks.

Adaptability Comparison Relative to prior work [67, 66], our clustering method is more adaptable and can efficiently be used for detection/segmentation as well as image classification. However, the representation learning step for clustering must be changed for such target tasks, which can hinder downstream performance as a representation learning technique like MoCo-v2 may be more challenging on smaller scale datasets like Pascal VOC. The domain classifier, on the other hand, avoids these challenges and does not have to change when the target task is changed.

Performance Comparison We observe that all of our proposed filtering techniques yield consistent gains of up to 9% over random filtering, confirming their applicability to lower level tasks. In the segmentation setting, pre-training on a 12 % subset can match full pre-training performance. Clustering produces meaningful gains, but the domain classifier outperforms it in almost every object detection scenario and the majority of segmentation metrics. This is especially pronounced with a larger pre-training subset, showing the domain

classifier can effectively filter more relevant images.

ImageNet+	224x224			112x112		
Pre-train. Sel. Method	**Cars**	**Birds**	**fMow**	**Cars**	**Birds**	**fMow**
ImageNet	83.52	67.49	56.11	84.09	66.57	56.83
ImageNet+Domain Cls.	**84.33**	**69.78**	**57.95**	**84.56**	**69.88**	**58.04**

Table 6.4: Classification results for ImageNet+. By fine-tuning ImageNet weights on our ImageNet filtered subset, we can improve ImageNet pre-training performance on downstream classification tasks.

6.5.5 Improving on Full ImageNet Pre-training

Thus far, we have used ImageNet as a proxy for a very large scale dataset where full pre-training would be infeasible, and we show the promise of our methods in pre-training on subsets of ImageNet. We note that pre-trained models on ImageNet (1.28M images) are readily available, so we motivate practical use of our method by showing how they can outperform full ImageNet pre-training.

ImageNet+ Here, we take a model pre-trained on ImageNet and help it focus on specific examples with relevant features by tuning its weights for a small additional number of epochs on our conditionally filtered subsets before transfer learning. We find this is effective in the unsupervised setting due to its focus on image features without label distributions, as mentioned previously. Thus, we apply this method to Cars/Birds/fMoW and tune pre-trained ImageNet weights with MoCo-v2 for 20 additional epochs on 150K domain classifier filtered ImageNet subsets. We present results in Table 6.4 and report improvements by up to 1-3% over full ImageNet pre-training, a strong performance increase for minimal extra cost.

Large Scale Filtering Here, we improve on full ImageNet by filtering a similar number of images from a larger scale dataset. To this end, we assemble a large scale dataset consisting of 6.71M images from the Places, OpenImages, ImageNet, and COCO datasets [120, 143, 162] and filter 1.28M images using the domain classifier conditioned on the Cars dataset. We pre-train using MoCo-v2 and present our accuracy on the Cars dataset in Table 6.5. Our filtering methods improve on the current default of 224 resolution ImageNet pre-training by 1-1.5% with good cost tradeoffs. Interestingly, a random subset of the large scale dataset performs worse than ImageNet, showing that our filtering method is crucial to select relevant examples. We also note that previously, for classification, our filtering methods saw larger gains with 6% than 12% subsets,

	Random@224	**LargeScale@112**	**LargeScale@224**	**ImageNet@224**
Accuracy	82.96	84.29	**84.51**	83.52
Cost (hrs)	210-220	130-140	230-240	210-220

Table 6.5: Results on large scale experiments. Filtering a large scale dataset with the domain classifier improves accuracy on the Stanford Cars dataset over a random subset and ImageNet with about 10% more cost at 224 pixels resolution. and 35% savings at 112 pixels resolution.

but here we use a 19% subset, so access to even larger scale data could further improve results. This shows promise that in the future, our methods can leverage exponentially growing data scale to replace full ImageNet pre-training for a new pre-training method.

6.6 Additional Methods

6.6.1 Active Learning

Active learning is a research field concentrating on understanding which samples in a pool of samples should be given priority for annotation. One of the most common and simple active learning method relies on training a model on a labeled dataset and finding the entropy of the unseen samples by running them through the trained model. Next, top N unseen samples w.r.t their entropy (assigned by the current model) are listed in descending order. Usually, there is a single data distribution for labelled and unlabelled data, however, for our task we consider two data distributions: pre-training and target, which can be similar or completely different. For this reason, we apply two variations of active learning to conditional pre-training. First, we train a network f_t on the target dataset $\mathcal{D}_t$ and run images x_s^i in source dataset through the network f_t to get the entropy of the predictions. Next, we list the images x_s^i by ascending or descending entropy and choose the top $\mathcal{N}^{'}$ images. Choosing high entropy samples can be interpreted as standard active learning, and we call the method that chooses low entropy images *Inverse Active Learning*.

Dataset	**#classes**	**#train**	**#test**
Stanford Cars [159]	196	8143	8041
Caltech Birds [160]	200	6000	2788
Functional Map of the World [112]	62	18180	10609

Table 6.6: We use three challenging visual categorization datasets to evaluate the proposed pre-training strategies on target classification tasks.

6.6.2 Experimental Setup

For classification tasks, we train the linear classification layer from scratch and finetune the pre-trained backbone weights. We give basic details about our classification datasets in Table 6.6.

Methods We experiment with clustering based filtering, using $K = 200$ clusters and both average and min distance to cluster centers, as well as our domain classifier method, using ResNet-18 [87] as our classifier. Furthermore, we combine our filtering methods with downsizing pre-training image resolution from 224x224 to 112x112 using bilinear interpolation. We always perform filtering on 224 resolution source images, but use it to pre-training at both resolutions to assess flexibility, as we want robust methods that do not need to be specifically adjusted to the pre-training setup.

Supervised Pre-training. For supervised pre-training, in all experiments, we utilize the ResNet-34 model [87] on 1 Nvidia-TITAN X GPU. We perform standard cropping/flipping transforms for ImageNet and the target data. For pre-training, we pretrain on the given subset of ImageNet for 90 epochs, utilizing SGD with momentum .9, weight decay of 1e-4, and learning rate .01 with a decay of 0.1 every 30 epochs. We finetune for 90 epochs with a learning rate decay of 0.1 every 30 epochs for all datasets. For Cars and Birds, we utilize SGD with momentum .9 [163], learning rate 0.1, and weight decay of 1e-4. For fMoW, we utilize the Adam optimizer [164] with learning rate 1e-4.

Unsupervised Pre-training. For unsupervised pre-training, we utilize the state of the art MoCo-v2 [2] technique using a ResNet-50 model [87] in all experiments. We train on 4 Nvidia GPUs. MoCo [3, 2] is a self-supervised learning method that utilizes contrastive learning, where the goal is to maximize agreement between different views of the same image (positive pairs) and to minimize agreement between different images (negative pairs). Our choice to use MoCo is driven by (1) performance, and (2) computational cost. Compared to other self-supervised frameworks, such as SimCLR [33], which require a batch size of 4096, MoCo uses a momentum updated queue of previously seen samples and achieves comparable performance with a batch size of just 256 [3].

We keep the same data augmentations and hyperparameters used in [2]. We finetune the MoCo pre-trained backbone on our target tasks for 100 epochs using a learning rate of 0.001, batch size of 64, SGD optimizer for Cars and Birds, and Adam optimizer for fMoW.

6.6.3 Low Level Tasks

Object Detection. We use a standard setup for object detection with a Faster R-CNN detector with a R50-C4 backbone as in [3, 118, 119]. We pre-train the backbone with MoCo-v2 on the full or filtered subset of ImageNet. We finetune the final layers for 24k iterations ($\sim$ 23 epochs) on trainval2007 ($\sim$ 5k images). We evaluate on the VOC test2007 set with the default metric AP50 and the more stringent metrics of COCO-style [120] AP and AP75. For filtering, we use the domain classifier with no modifications and for clustering we use MoCo-v2 on Pascal VOC to learn representations.

Semantic Segmentation. We use PSAnet [116] network with ResNet-50 backbone to perform semantic segmentation. We train PSAnet network with a batch size of 16 and a learning rate of 0.01 for 100 epochs and use SGD optimizer. Similar to object detection, we pre-train the backbone with MoCo-v2 on the full or filtered subset of ImageNet and then we finetune the network using VOC train2012. We evaluate on the VOC test2012 set with the following three metrics: (a) **mIOU**: standard segmentation metric, (b) **mAcc**: mean classwise pixel accuracy, (c) **allAcc**: total pixel accuracy. For filtering, we use the domain classifier with no modifications and for clustering we use MoCo-v2 on Pascal VOC to learn representations.

Supervised Pre-train.		Target Dataset			Cost (hrs)
224 x 224		Small Shift		Large Shift	
Pretrain. Sel.	Method	Cars	Birds	fMow	
0%	**Random Init.**	52.89	42.17	43.35	0
100%	**Entire Dataset**	82.63	74.87	59.05	160-180
6%	**Random**	72.2	57.87	50.25	30-35
	Inv. Active Learning	72.19	58.17	49.7	40-45
	Active Learning	73.17	57.77	**50.91**	40-45
	Domain Cls.	**74.37**	**59.73**	**51.17**	35-40
	Clustering (Avg)	73.64	56.33	**51.14**	40-45
	Clustering (Min)	**74.23**	57.67	50.27	40-45
12%	**Random**	76.12	62.73	53.28	45-50
	Inv. Active Learning	76.1	62.7	**53.43**	55-60
	Active Learning	76.43	63.7	**53.63**	55-60
	Domain Cls.	76.18	**64**	**53.41**	50-55
	Clustering (Avg)	**77.12**	61.73	53.12	55-60
	Clustering (Min)	75.81	**64.07**	52.91	55-60

Supervised Pre-train.		Target Dataset			Cost (hrs)
112 x 112		Small Shift		Large Shift	
Pretrain. Sel.	Method	Cars	Birds	fMow	
0%	**Random Init**	52.89	42.17	43.35	0
100%	**Entire Dataset**	83.78	73.47	57.39	90-110
6%	**Random**	72.76	57.4	49.73	15-20
	Inv. Active Learning	71.05	58.43	49.56	25-30
	Active Learning	72.95	56.3	48.94	25-30
	Domain Cls.	73.66	**58.73**	50.66	20-25
	Clustering (Avg)	**74.53**	56.97	**51.32**	25-30
	Clustering (Min)	71.72	**58.73**	49.06	25-30
12%	**Random**	75.4	62.63	52.59	30-35
	Inv. Active Learning	75.3	62.4	52.45	40-45
	Active Learning	76.26	61.9	52.04	40-45
	Domain Cls.	76.36	**63.5**	**53.37**	35-40
	Clustering (Avg)	**77.53**	61.23	52.67	40-45
	Clustering (Min)	76.36	63.13	51.6	40-45

Table 6.7: Results on supervised pre-training and classification tasks, including Active Learning.

6.7 Additional Results

6.7.1 Active Learning

We utilize our Active Learning based methods using the same supervised pre-training and finetuning setup described previously. We present our results updated with Active Learning in Table 6.7.

Least vs Most Confident Samples We see that at 224×224 pixels resolution pre-training, standard active learning seems to be applicable to the transfer learning setting as selecting samples with high entropy generally does better than the inverse. However, at lower resolution (112×112 pixels) pre-training, active learning does worse than inverse active learning and random in most cases, suggesting a lack of robustness for the active learning method since filtering is performed at 224×224 pixels resolution.

Performance Comparison As alluded to, active learning methods perform noticeably worse in the lower resolution setting for all datasets, suggesting that filtering and pre-training conditions must be similar to maintain good performance, unlike domain classifier and clustering. In general, we see that for Cars and Birds, even at 224×224 pixels resolution pre-training, active learning performance lags behind our clustering and domain classifier methods and struggles to improve over the simple random baseline in several settings. In contrast, for an out of distribution dataset like fMoW, active learning does well in the 224×224 pixels resolution pre-training setting. Since active learning directly considers label distribution when filtering, it may be more prone to overfitting compared to the other methods. This can degrade its performance when relevant features are shared between the pre-training and target datasets and thus focusing only on features, not labels, when filtering may be more effective. However, in fMoW, there is very little overlap in relevant features with ImageNet, bridging the gap between active learning, and domain classifier and clustering.

Adaptability Comparison Active learning is noticeably less flexible than other methods as it relies on a notion of confidence that can be hard to construct and quantify for target tasks besides classification. As said, it is also much more sensitive to filtering and pre-training resolution, making it a less robust method. In

general, we see that our clustering and domain classification methods can outperform a non-trivial baseline like active learning in flexibility, adaptability, and performance.

6.8 Conclusion

In this chapter, we proposed filtering methods to efficiently pre-train on large scale datasets conditioned on the transfer learning task. To further improve pre-training efficiency, we proposed decreased image resolution for pre-training and found this shortens pre-training cost by 30-50% with similar transfer learning accuracy. Additionally, we introduced sequential pre-training to improve the efficiency of conditional pre-training with multiple target tasks. Finally, we demonstrated how our methods can improve the standard ImageNet pre-training by focusing models pre-trained on ImageNet on relevant examples and filtering an ImageNet-sized dataset from a larger scale dataset. Our method is appropriately positioned for sustainability related applications which have large amount unlabeled satellite imagery at their dispense for various tasks like poverty estimation, infrastructure management, etc. Our methods can replace pre-training on full labeled or unlabeled dataset with a target task specific efficient conditional pre-training without sacrificing performance.

Chapter 7

Conclusions and Future Work

7.1 Summary of Contributions

The motivating theme for this dissertation was to combine machine learning and satellite imagery that can effectively be used for many sustainability-related tasks including poverty prediction, infrastructure measurement, and forest monitoring. We developed accurate and interpretable ML systems for tasks like poverty prediction. However, we identified supervision (and the large amount of cost associated with data acquisition and annotation) as the key bottleneck in developing ML systems that can draw inferences and make decisions in such tasks. To offset the high cost requirements of such systems, we discussed agents that can efficiently use high-resolution satellite imagery. Additionally, to mitigate the high supervision requirements of ML models like object detection, segmentation, classification, etc., used as part of the whole system, we discussed the theory and practice of unsupervised/self-supervised learning, data augmentation, and efficient conditional pre-training. Next, we summarize the contributions within each of these parts.

In **Chapter 2** [4], we discuss that accurate local-level poverty measurement is an essential task for governments and humanitarian organizations to track the progress towards improving livelihoods and distribute scarce resources. We highlight that recent computer vision advances in using satellite imagery to predict poverty have shown increasing accuracy, but they do not generate features that are interpretable to policymakers, inhibiting adoption by practitioners. Then we demonstrate an interpretable computational framework to accurately predict poverty at a local level by applying object detectors to high resolution (30cm) satellite images. Using the weighted counts of objects as features, we achieve 0.539 Pearson's r^2 in predicting village level poverty in Uganda, a 31% improvement over existing (and less interpretable) benchmarks. Feature importance and ablation analysis revealed intuitive relationships between object counts and poverty predictions. Our results suggested that interpretability does not have to come at the cost of performance, at least in this important domain.

Primary Contributions in Chapter 2 Performed a literary survey of various object recognition models, regression and interpretability methods and performed all the experiments and analyses.

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. In **Chapter 3** [17], we turned our attention to such challenges and explored opportunities to make these systems cost-effective. 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 again apply this approach to the task of poverty prediction in Uganda, building on our earlier approach [4] that used object detection to count objects and use these counts to predict poverty. Our approach exceeds our previous performance benchmarks on this task while using 80% fewer high-resolution images, and could be useful in many domains that require high-resolution imagery.

Primary Contributions in Chapter 3 Jointly proposed the idea of extending a previous work to reduce acquisition cost of high-resolution satellite imagery for poverty prediction. Performed all the experiments and analyses including data scraping of satellite imagery from multiple seasons.

Image classification, object detection, and semantic segmentation models are important components of an effective computational framework for various sustainability related tasks. The performance of these models are highly dependent on the appropriate pre-training of their backbone networks. A main purpose of such pre-training is to learn good representations (i.e., features) that can be transferred to these downstream tasks of detection, segmentation, classification, etc., by fine-tuning on limited training data. In the next three chapters, we explore and propose novel methods to improve unsupervised/self-supervised (for pre-training) learning that can effectively boost performance of various downstream tasks.

In **Chapter 4** [18], we discuss that contrastive learning methods have significantly narrowed the gap between supervised and unsupervised learning on computer vision tasks. We explore their application to geo-located datasets, e.g. remote sensing, where unlabeled data is often abundant but labeled data is scarce. We first show that due to their different characteristics, a non-trivial gap persists between contrastive and supervised learning on standard benchmarks. To close the gap, we propose novel training methods that exploit the spatio-temporal structure of remote sensing data. We leverage spatially aligned images over time to construct temporal positive pairs in contrastive learning and geo-location to design pre-text tasks. Our experiments showed that our proposed method closes the gap between contrastive and supervised learning on image classification, object detection and semantic segmentation for remote sensing. Moreover, we demonstrate that the proposed method can also be applied to geo-tagged ImageNet images, improving down-stream performance on various tasks.

Primary Contributions in Chapter 4 Proposed the idea of using temporal information from satellite imagery for self-supervised representation learning. Performed all the experiments corresponding to the satellite dataset.

In **Chapter 5** [19], we propose a new form of Data Augmentation. Data augmentation is often used to enlarge datasets with synthetic samples generated in accordance with the underlying data distribution. To

enable a wider range of augmentations, we explore *negative* data augmentation strategies (NDA) that intentionally create out-of-distribution samples. We show that such negative out-of-distribution samples provide information on the support of the data distribution, and can be leveraged for generative modeling and representation learning. We introduce a new GAN training objective where we use NDA as an additional source of synthetic data for the discriminator. We prove that under suitable conditions, optimizing the resulting objective still recovers the true data distribution but can directly bias the generator towards avoiding samples that lack the desired structure. Empirically, models trained with our method achieve improved conditional/unconditional image generation along with improved anomaly detection capabilities. Further, we incorporate the same negative data augmentation strategy in a contrastive learning framework for self-supervised representation learning on images (*including satellite imagery*) and videos, achieving improved performance on downstream image classification, object detection, and action recognition tasks. These results suggest that prior knowledge on what does not constitute valid data is an effective form of weak supervision across a range of unsupervised learning tasks.

Primary Contributions in Chapter 5 Jointly proposed the idea of NDA. Primarily focused on experiments pertaining to contrastive learning on videos and images along with some experiments on GANs.

Finally in **Chapter 6** [20], we discuss that almost all the state-of-the-art neural networks for computer vision tasks are trained by (1) pre-training on a large-scale dataset and (2) finetuning on the target dataset. This strategy helps reduce dependence on the target dataset and improves convergence rate and generalization on the target task. Although pre-training on large-scale datasets is very useful, its foremost disadvantage is high training cost. To address this, we propose efficient filtering methods to select relevant subsets from the pre-training dataset. Additionally, we discover that lowering image resolutions in the pre-training step offers a great trade-off between cost and performance. We validate our techniques by pre-training on ImageNet in both the unsupervised and supervised settings and finetuning on a diverse collection of target datasets (including *satellite imagery*) and tasks. Our proposed methods drastically reduce pre-training cost and provide strong performance boosts. Finally, we improve standard ImageNet pre-training by 1-3$\%$ by tuning available models on our subsets and pre-training on a dataset filtered from a larger scale dataset.

Primary Contributions in Chapter 6 Proposed extension of the idea to unsupervised settings and performed experiments corresponding to that as well as transfer learning experiments (detection and segmentation).

7.2 Future Work

There is enormous potential for using AI and satellite imagery to address core questions in sustainability, including poverty mitigation, and food security. For such applications, ML models can potentially aid the design of experiments, and guide decisions for planning over long horizons. In practice however, we are limited by budget constraints for acquiring data as well as the high complexity and noise in real world systems. This dissertation presented advancements in machine learning to tackle some of these challenges. The

contributions of this dissertation suggest new opportunities and challenges for future work, some of which we highlight below.

Predicting Livelihood Indicators from Street-Level Imagery [165] present a novel approach to make predictions on poverty, population, and women's body-mass index from street-level imagery. We believe that our proposed representation learning and pre-training techniques can provided richer features in order to bolster the performance in predicting such indicators.

Scaling to the Whole World In order to scale ML systems for sustainability related tasks to the whole world, it is important to tackle the challenge of distribution shift caused due to geographical diversity. [113] analyze the accuracy of publicly available object-recognition systems on a geographically diverse dataset and find that the systems perform relatively poorly on household items that commonly occur in countries with a low household income. Their qualitative analyses suggest the drop in performance is primarily due to appearance differences within an object class (e.g., dish soap) and due to items appearing in a different context (e.g., toothbrushes appearing outside of bathrooms). Their study suggests that further work is needed to make object-recognition systems work equally well for people across different countries and income levels. The same has been observed for satellite imagery from different continents and countries, where an ML system trained on one country/continent performs poorly on other countries/continents. This advocates future research in this area to make ML systems robust to geographical changes.

Application to other Sustainability related tasks In this thesis, we primarily focused on poverty estimation as one of the main sustainable development tasks. Our future work includes application of our methods to other sustainability-related tasks like food security, infrastructure management, forest monitoring, etc., using high-resolution images at large scale.

In summary, we believe advancements in computation and supervision-constrained machine learning will have a profound impact in accelerating scientific research and addressing key challenges in the road to sustainable development. We hope that our methods can be employed as a cheap but effective alternative to traditional surveying methods for organizations to measure the well-being of developing regions.

Bibliography

[1] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018.

[2] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[3] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[4] 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.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] David E Sahn and David C Stifel. Poverty comparisons over time and across countries in africa. *World development*, 28(12):2123–2155, 2000.

[7] Kathleen Kahn, Stephen M Tollman, Mark A Collinson, Samuel J Clark, Rhian Twine, Benjamin D Clark, Mildred Shabangu, Francesc Xavier Gomez-Olive, Obed Mokoena, and Michel L Garenne. Research into health, population and social transitions in rural south africa: Data and methods of the agincourt health and demographic surveillance system1. *Scandinavian journal of public health*, 35(69_suppl):8–20, 2007.

[8] Philip Antwi-Agyei, Evan DG Fraser, Andrew J Dougill, Lindsay C Stringer, and Elisabeth Simelton. Mapping the vulnerability of crop production to drought in ghana using rainfall, yield and socioeconomic data. *Applied Geography*, 32(2):324–334, 2012.

[9] Alessio Signorini, Alberto Maria Segre, and Philip M Polgreen. The use of twitter to track levels of disease activity and public concern in the us during the influenza a h1n1 pandemic. *PloS one*, 6(5):e19467, 2011.

[10] Herman Anthony Carneiro and Eleftherios Mylonakis. Google trends: a web-based tool for real-time surveillance of disease outbreaks. *Clinical infectious diseases*, 49(10):1557–1564, 2009.

[11] UN Global Pulse. Mining indonesian tweets to understand food price crises. *Jakarta: UN Global Pulse*, 2014.

[12] Azadeh Nikfarjam, Abeed Sarker, Karen O'connor, Rachel Ginn, and Graciela Gonzalez. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22(3):671–681, 2015.

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

[14] Frank F Xu, Bill Y Lin, Qi Lu, Yifei Huang, and Kenny Q Zhu. Cross-region traffic prediction for china on openstreetmap. In *Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pages 37–42, 2016.

[15] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive computing*, 7(4):12–18, 2008.

[16] 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.

[17] Kumar Ayush, Burak Uzkent, Kumar Tanmay, Marshall Burke, David Lobell, and Stefano Ermon. Efficient poverty mapping using deep reinforcement learning. *arXiv preprint arXiv:2006.04224*, 2020.

[18] Kumar Ayush, Burak Uzkent, Chenlin Meng, Kumar Tanmay, Marshall Burke, David Lobell, and Stefano Ermon. Geography-aware self-supervised learning. *arXiv preprint arXiv:2011.09980*, 2020.

[19] Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano Ermon. Negative data augmentation. *arXiv preprint arXiv:2102.05113*, 2021.

[20] Shuvam Chakraborty, Burak Uzkent, Kumar Ayush, Kumar Tanmay, Evan Sheehan, and Stefano Ermon. Efficient conditional pre-training for transfer learning. *arXiv preprint arXiv:2011.10231*, 2020.

[21] Abhishek Sinha, Kumar Ayush, Jiaming Song, Kelly He, and Stefano Ermon. Flexible distribution shift and outlier detection with self-supervised kernels. *Under review ICCV*, 2021.

[22] Anthony Perez, Christopher Yeh, George Azzari, Marshall Burke, David Lobell, and Stefano Ermon. Poverty prediction with public landsat 7 satellite imagery and machine learning. *arXiv preprint arXiv:1711.03654*, 2017.

[23] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, and Li Fei-Fei. Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the united states. *Proceedings of the National Academy of Sciences*, 114(50):13108–13113, 2017.

[24] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[25] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[27] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017.

[28] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015.

[29] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2701–2710, 2017.

[30] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[31] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[32] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[33] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[34] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[35] Anil M Cheriyadat. Unsupervised feature learning for aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):439–451, 2013.

[36] Yansheng Li, Chao Tao, Yihua Tan, Ke Shang, and Jinwen Tian. Unsupervised multilayer feature learning for satellite image scene classification. *IEEE Geoscience and Remote Sensing Letters*, 13(2):157–161, 2016.

[37] Adriana Romero, Carlo Gatta, and Gustau Camps-Valls. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1349–1362, 2015.

[38] Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3967–3974, 2019.

[39] Xiaoqiang Lu, Xiangtao Zheng, and Yuan Yuan. Remote sensing scene classification by unsupervised representation learning. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9):5148–5157, 2017.

[40] T Nathan Mundhenk, Goran Konjevod, Wesam A Sakla, and Kofi Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *European Conference on Computer Vision*, pages 785–800. Springer, 2016.

[41] Lichao Mou, Pedram Ghamisi, and Xiao Xiang Zhu. Unsupervised spectral–spatial feature learning via deep residual conv–deconv network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(1):391–406, 2017.

[42] Lefei Zhang, Liangpei Zhang, Bo Du, Jane You, and Dacheng Tao. Hyperspectral image unsupervised classification by robust manifold matrix factorization. *Information Sciences*, 485:154–169, 2019.

[43] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010.

[44] Burak Uzkent, Evan Sheehan, Chenlin Meng, Zhongyi Tang, Marshall Burke, David B Lobell, and Stefano Ermon. Learning to interpret satellite images using wikipedia. In *IJCAI*, pages 3620–3626, 2019.

[45] Kevin Tang, Manohar Paluri, Li Fei-Fei, Rob Fergus, and Lubomir Bourdev. Improving image classification with location context. In *Proceedings of the IEEE international conference on computer vision*, pages 1008–1016, 2015.

[46] Hayate Iso, Shoko Wakamiya, and Eiji Aramaki. Density estimation for geolocation via convolutional mixture density network. *arXiv preprint arXiv:1705.02750*, 2017.

[47] Eric Muller-Budack, Kader Pustu-Iren, and Ralph Ewerth. Geolocation estimation of photos using a hierarchical model and scene classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018.

[48] Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-only geographical priors for fine-grained image classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9596–9606, 2019.

[49] Grace Chu, Brian Potetz, Weijun Wang, Andrew Howard, Yang Song, Fernando Brucher, Thomas Leung, and Hartwig Adam. Geo-aware networks for fine-grained recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[50] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016.

[51] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *2008 ieee conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[52] James Hays and Alexei A Efros. Large-scale image geolocalization. In *Multimodal location estimation of videos and images*, pages 41–62. Springer, 2015.

[53] Nam Vo, Nathan Jacobs, and James Hays. Revisiting im2gps in the deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2621–2630, 2017.

[54] Yi Lin Sung, Sung-Hsien Hsieh, Soo-Chang Pei, and Chun-Shien Lu. Difference-seeking generative adversarial network–unseen sample generation. In *International Conference on Learning Representations*, 2019.

[55] Steve Hanneke, Adam Tauman Kalai, Gautam Kamath, and Christos Tzamos. Actively avoiding nonsense in generative models. In *Conference On Learning Theory*, pages 209–227, 2018.

[56] Avishek Joey Bose, Huan Ling, and Yanshuai Cao. Adversarial contrastive estimation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1021–1032, 2018.

[57] Ming Hou, Brahim Chaib-Draa, Chao Li, and Qibin Zhao. Generative adversarial positive-unlabeled learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2255–2261, 2018.

[58] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[59] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[60] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. In *Advances in Neural Information Processing Systems*, pages 10792–10801, 2018.

[61] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

[62] Michael Xie, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Transfer learning from deep features for remote sensing and poverty mapping. *arXiv preprint arXiv:1510.00098*, 2015.

[63] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[64] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. *arXiv preprint arXiv:1901.09960*, 2019.

[65] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2661–2671, 2019.

[66] Xi Yan, David Acuna, and Sanja Fidler. Neural data server: A large-scale search engine for transfer learning data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[67] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4109–4118, 2018.

[68] Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018.

[69] Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1086–1095, 2017.

[70] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[71] General Assembly. Sustainable development goals. *SDGs Transform Our World*, 2030, 2015.

[72] Morten Jerven. How much will a data revolution in development cost? In *Forum for Development Studies*, volume 44, pages 31–50. Taylor & Francis, 2017.

[73] 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.

[74] Deon Filmer and Lant H Pritchett. Estimating wealth effects without expenditure data—or tears: an application to educational enrollments in states of india. *Demography*, 38(1):115–132, 2001.

[75] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[76] 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.

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

[78] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[79] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE, 2011.

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

[81] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[82] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[83] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[84] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[85] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*, 2016.

[86] Ryan Engstrom, Jonathan Hersh, and David Newhouse. Poverty from space: using high-resolution satellite imagery for estimating economic well-being, 2017.

[87] 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.

[88] Dave Donaldson and Adam Storeygard. The view from above: Applications of satellite data in economics. *Journal of Economic Perspectives*, 30(4):171–98, 2016.

[89] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[90] 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.

[91] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[92] 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.

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

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] 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.

[102] 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.

[103] Burak Uzkent and Stefano Ermon. Learning when and where to zoom with deep reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12345–12354, 2020.

[104] 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.

[105] 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.

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

[107] 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.

[108] Gloria Rosenthal and James A Rosenthal. *Statistics and data interpretation for social work*. Springer publishing company, 2011.

[109] 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.

[110] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[111] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[112] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6172–6180, 2018.

[113] Terrance de Vries, Ishan Misra, Changhan Wang, and Laurens van der Maaten. Does object recognition work for everyone? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 52–59, 2019.

[114] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[115] Adam Van Etten, Dave Lindenbaum, and Todd M Bacastow. Spacenet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232*, 2018.

[116] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018.

[117] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[118] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[119] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[120] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[121] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. *arXiv preprint arXiv:1904.13132*, 2019.

[122] Ayush Jaiswal, Rex Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Unsupervised adversarial invariance. In *Advances in Neural Information Processing Systems*, pages 5092–5102, 2018.

[123] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[124] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[125] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[126] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.

[127] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[128] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[129] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[130] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[131] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016.

[132] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[133] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *arXiv preprint arXiv:1503.02406*, March 2015.

[134] Xuanlong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *arXiv preprint arXiv:0809.0853*, (11):5847–5861, September 2008.

[135] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. *arXiv preprint arXiv:1905.06922*, May 2019.

[136] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. *arXiv preprint arXiv:1905.06922*, 2019.

[137] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*, October 2019.

[138] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[139] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[140] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[141] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[142] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[143] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[144] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

[145] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

[146] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[147] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[148] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[149] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[150] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33, 2020.

[151] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016.

[152] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017.

[153] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

[154] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[155] Mijung Park and Jonathan Pillow. Bayesian active learning with localized priors for fast receptive field characterization. *Advances in neural information processing systems*, 25:2348–2356, 2012.

[156] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[157] Aditya Grover, Jiaming Song, Ashish Kapoor, Kenneth Tran, Alekh Agarwal, Eric J Horvitz, and Stefano Ermon. Bias correction of learned generative models using likelihood-free importance weighting. In *Advances in Neural Information Processing Systems*, pages 11058–11070, 2019.

[158] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. In *Advances in Neural Information Processing Systems*, pages 8252–8262, 2019.

[159] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[160] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[161] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[162] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, and et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, Mar 2020.

[163] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[164] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[165] Jihyeon Lee, Dylan Grosz, Burak Uzkent, Sicheng Zeng, Marshall Burke, David Lobell, and Stefano Ermon. Predicting livelihood indicators from community-generated street-level imagery. 2021.