

Automated Tree Censusing from Aerial Imagery with Noisy Supervision

Sandeep Mukherjee
Ken Goldberg, Ed.
Alexei (Alyosha) Efros, Ed.



Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-81

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-81.html>

May 10, 2024

Copyright © 2024, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Automated Tree Censusing from Aerial Imagery with Noisy Supervision

by

Sandeep Mukherjee

A thesis submitted in partial satisfaction of the
requirements for the degree of

Master of Science

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair
Professor Alexei Efros

Spring 2024

Automated Tree Censusing from Aerial Imagery with Noisy Supervision

Copyright 2024
by
Sandeep Mukherjee

Automated Tree Censusing from Aerial Imagery with Noisy Supervision

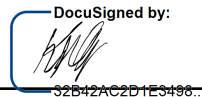
by Sandeep Mukherjee

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II.**

Approval for the Report and Comprehensive Examination:

Committee:

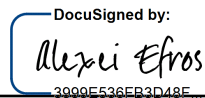
DocuSigned by:

32B42AC2D1E3496...

Professor Ken Goldberg
Research Advisor

5/10/2024

(Date)

* * * * *

DocuSigned by:

3099E536EB3D48E...

Professor Alexei Efros
Second Reader

5/6/2024

(Date)

Abstract

Automated Tree Censusing from Aerial Imagery with Noisy Supervision

by

Sandeep Mukherjee

Master of Science in Computer Science

University of California, Berkeley

Professor Ken Goldberg, Chair

Classifying trees from GPS-registered aerial imagery poses several challenges: data are low-signal and noisy and there is a long-tailed, fine-grained class distribution. In this setting, supervised classification heads trained on DINO features under-perform simple fully-supervised classifiers. On the other hand, noise-resistant feature extraction excels but can be overly class-indicative. We find that the Graph Attention Transformer (GAT), is well-specified to model geo-spatial correlations present in aerial imagery, but tail class sensitivity suffers when trained on overly class-indicative features (section 6.2). We present **S**oftening **H**ead **I**mbalances with **E**ffective **L**earning and **D**ebiasing for **G**raph **N**eural **N**etworks (SHIELD-GNN), a classification method that uses temperature-softened teacher prediction penalties and test-time debiasing for graph-aware predictions, surpassing baselines by up to 7% accuracy and 3% average recall.

In this thesis, we also provide a justification for why SHIELD-GNN is able to prevent tail class smoothing with empirical evidence. Finally, we provide early results for ongoing work in tree segmentation.

To my parents

Contents

Contents	ii
List of Figures	iv
List of Tables	vii
1 Introduction	1
1.1 Motivation	1
1.2 Core Challenges	1
1.3 Contributions	2
2 Related Work	5
2.1 Aerial Image Classification	5
2.2 Graph Neural Networks (GNNs)	5
2.3 Object Detection	6
2.4 Feature Extraction for GNNs	6
2.5 Learning with Noisy Labels	6
2.6 Fine-grained Classification	7
2.7 Test Time Adaptation and Debiasing	7
2.8 Tree Canopy Coverage Estimation	7
3 Problem Statement	8
4 Method	10
4.1 Preliminaries	10
4.2 SHIELD-GNN	11
5 Experimental Design and Results	14
5.1 Experimental Setup	14
5.2 Results	16
5.3 Ablation Studies	18
6 Discussion	22

6.1	Distribution Shift in Auto Arborist	22
6.2	“Class-Indicative” Features	22
6.3	Motivating SHIELD-GNN	23
7	Limitations	34
8	Extension: Unsupervised Object Discovery	35
8.1	Motivation	35
8.2	Evaluation Protocol Proposal	35
9	Conclusion and Future Work	38
9.1	Conclusion	38
9.2	Future Work	39
	Bibliography	40
A	Implementation Details	47
A.1	Noise Robust Feature Extraction	47
A.2	SHIELD-GNN	47

List of Figures

1.1	Real and Induced Noise. Auto Arborist is a naturally noisy dataset due to its data collection process. Here we demonstrate two kinds of noise: multi-object noise (left) and null noise (no object, center left). In xView, we show induced noise on the container crane (center right) and airplane hangar (right) categories. On the right image, we see that a collection of planes, which is likely correlated with airplane hangars.	2
1.2	Examples of clean imagery. Here we visualize examples of clean imagery from Auto Arborist: there is typically one prominent tree, the image is well lit, and the tree typically has features that are identifiable. For example Phoenix trees (bottom right) are distinctly palm trees.	3
3.1	Graph Construction. To capture the surrounding context for images, we construct a radius graph using GPS coordinates on Auto Arborist. To visualize the full context of examples, we use Google Earth: the red box is the target tree and the black boxes are its identified neighbors. This also shows the bias of the Auto Arborist dataset, which only catalogs trees on public land. We also see homophily in the graph as similar trees are often found on the same street.	8
4.1	The SHIELD-GNN pipeline begins by constructing a radius graph. It then leverages DivideMix for feature extraction and base prediction generation. These elements are used to train a GATv2 model, which outputs a prediction set. Finally, this set is debiased to arrive at the final predictions.	11
4.2	Categories of noisy examples from Auto Arborist. From these examples, we can see the wide variety of noise in the dataset, which helps motivate the need for a noise-robust training loop. Although DivideMix was originally created for structured noise (ie label perturbations), it can handle less structured noise (like multiple trees) through pseudolabels.	12

5.1	SHIELD-GNN performance visualized spatially on Los Angeles (top), Washington DC (Middle), and Denver (Bottom). We show ℓ_1 label distribution distance of test set chunks from the train set (left) and the change in chunk accuracy from DivideMix to SHIELD-GNN (right). We often see the largest increases in accuracy in sections where the distribution distance is high, suggesting that SHIELD-GNN helps improve robustness to small geospatial shifts. We use OpenStreetMap [50] for visualization.	21
6.1	ℓ_1 distribution distance between regions (left) and cities (right). We reproduce the figure from Beery <i>et al.</i> [6] showing that these large label shifts occur on the public subset, but only represents one axis of distribution shift present in the data.	23
6.2	Visual Distribution Shift. We randomly sample 10 aerial images from Los Angeles (top), San Francisco (middle) and New York (bottom) to show to qualitative differences between sets of images from each city.	28
6.3	Within class distribution shift is common in Auto Arborist due to the relatively broad genus level labels. Platanus (Planes tree, Sycamore) trees are an example of this as the genus encompasses are many visually distinct species [10].	29
6.4	Class-wise LA validation accuracy. Highly "aerially distinctive" trees like Phoenix and Washingtonia (both palms) have better validation accuracy than other genera. This indicates that Beery <i>et al.</i> were correct and there are some visual differences amongst genera which aid in classification.	30
6.5	t-SNE visualization of the induced distribution shift between the training and validation sets on 10 randomly selected classes from Auto Arborist LA. . .	31
6.6	Negative entropy penalty and average recall. We show class average recall as a function of the penalty weight (left) in log-scale. Initially, as the penalty increases and we get less class-indicative features improve AR, but then as the penalty starts damaging the representation, we see a corresponding drop off in performance. Class indication scores ($\tau = 1$, right) show that the negative entropy classifier properly regulates class-indicative features.	31
6.7	Class Conditional Homophily. Here we show homophily (the node-average proportion of neighbors which share the same label as the target node) as a function of log class size. It's clear that there is significant variance in homophily amongst classes which is correlated with class size. In the case of xView with extreme imbalance, the top two classes are considerably more homophilic than other classes.	32

- 6.8 **Desirable and Undesirable Behavior of SHIELD-GNN.** Here we show examples of desirable (top three) and undesirable (bottom three) behaviors of SHIELD-GNN from the validation set. For each example, we show the average neighbor prediction (the softmax of the average of neighbor logits), the base DivideMix softmax prediction, the base GNN softmax prediction, and finally the SHIELD-GNN prediction. The correct genus is highlighted in yellow. In each of the top three examples, SHIELD-GNN produces a prediction which is significantly closer to DivideMix resulting in a more accurate classification. We then display two failure modes in the bottom three images, the first image shows a situation where we correctly predict the genus even though there is clearly no tree. The second failure mode is when SHIELD-GNN does not overcome the neighbor sway and makes a confidently incorrect prediction in line with the neighbors, not the base prediction. 33
- 8.1 **Failure Modes of Existing Tree Detection.** Here we visualize the failure modes of DeepForest and SAM for tree detection. **(a)** We show that DeepForest often misses trees and is relative sensitive to image quality, whereas SAM can generate more masks, but sometimes groups crowns together. **(b)** However, SAM is also unable to recognize some trees (particularly deciduous trees) and single objects. MaskCut is able to address this, but doesn't produce as robust segmentations as SAM. 36

List of Tables

5.1	Dataset statistics of Auto Arborist (AA) splits and xView. We use the definition of node homophily defined by Pei <i>et al.</i> [53] and imbalance (imb.) ratio: the ratio of the largest class size to $\max(\text{smallest class size}, 10)$. We also include statistics on xView with subgraphs of only the top 2 (head-only) and bottom 58 classes (tail-only).	15
5.2	DINO trained on Auto Arborist LA and DINO V2 without finetuning on the top 100 classes of Auto Arborist LA (validation set).	16
5.3	Comparison of methods on the top 100 genera from Auto Arborist LA. We also include the Top-5 Classifier (section 5.1).	17
5.4	Comparison of methods on robustness to distribution shifts in Auto Arborist. The diagonal blocks correspond to validation accuracies (<i>i.e.</i> West-trained and West-tested).	17
5.5	Comparison of noisy-data methods on xView with noise perturbation. We include the baseline Top-5 Fixed classifier (section 5.1) to emphasize that accuracy is a relatively uninformative metric when class imbalance is so extreme.	18
5.6	Comparison of architectures on Auto Arborist LA using the cross-entropy loss. We adopt the EfficientNet V2-S as our backbone because of it’s small size and relatively strong performance.	19
5.7	We ablate each component of the SHIELD-GNN pipeline on Auto Arborist LA. Baseline refers to the predictions produced by DivideMix, GATv2 refers to a GATv2 trained with the cross-entropy loss, +Debiasing refers a GATv2 with test-time debiasing only, +Teacher refers to a GATv2 with the teacher penalized loss only, and +Both refers to the full SHIELD-GNN pipeline.	19
5.8	Class-average recall on the Auto Arborist LA test set (Santa Monica), computed over a grid of possible debias λ and temperature T	20
5.9	Accuracy on the Auto Arborist LA test set (Santa Monica), computed over a grid of possible debias λ and temperature T	20
6.1	We sweep various regularizers (Dropout, Label Smoothing, and Weight Decay) over and report validation and test average recall and accuracy. Note that while the regularizers are able to improve accuracy (by about 1% over not using them), they are unable to meaningfully improve AR, which harms performance out of distribution, on the test set.	25

6.2	We try three potential remedies for noisy labels impacting the GAT’s training: first fitting to DivideMix pseudolabels, next weighting loss by the DivideMix probability of the labe, and finally using the ELR loss.	26
A.1	Auto Arborist SHIELD-GNN implementation details.	49
A.2	xView SHIELD-GNN implementation details.	49

Acknowledgments

It was the all people who have been around me all this time who have helped mold me into the person and researcher I am today.

I owe a debt of gratitude to Professor Goldberg for welcoming me into the lab, mentoring me, and supporting my development since freshman year. Being in the lab has been an awesome opportunity to be around and conduct interesting research—I will forever cherish that. From this, I’ve learned how to “think like a scientist,” a challenging, rewarding, and ongoing process.

Next, I’d like to express my appreciation for my mentors: Simeon Adebola, Jonathan Huang, and Sara Beery. Simeon, you were able to offer good advice and help me find clarity in the project. When experiments were yielding confusing results, your advice and help working through them was invaluable in clarify the underlying story. Jonathan and Sara, your expertise, guidance, and positivity were incredibly helpful in my development this year. Your collective technical advice, from engaging with typos in loss functions to help work through strange-seeming results was invaluable. However, more than the technical advice, I greatly appreciate your help in honing my internal barometer for what makes research interesting and fun. Your advice has truly helped me take a few gradient steps towards being a good researcher.

I’d also like to thank my collaborators: Abby O’Neill, Ethan Qiu, Rishi Parikh, and Shrey Aeron. Your countless hours in the trenches with me—writing papers, debugging code, and talking about all sorts of things—are some of my favorite memories from the past years.

Finally, thank you to my family: my mother, father, and sister, who have supported and inspired me for my whole life. Your constant support and advice has been invaluable especially when you listen to me ramble about the latest thing that had broken or not gone according to plan.

Chapter 1

Introduction

1.1 Motivation

Environmental monitoring and Earth observation from aerial imagery have the potential to enable policymakers to make data-informed decisions to facilitate societal adaptation to a changing climate [8, 58]. However, aerial data repositories from satellite and low-flying aircraft are currently in the petabyte scale and growing, making extracting useful and relevant information to support policy intractable without automation. Aerial image classification has potential impact in humanitarian aid and disaster relief, wilderness forests, agriculture, and urban mapping with uses in city planning, resource management, and environmental monitoring [79, 1, 48, 31]. For example, urban ecologists need to know the location and type of trees in cities so that they can target replanting to improve climate adaptation. Collecting this information from ground-level tree censuses is both time consuming and expensive, thus automated tree genus classification from Global Positioning System (GPS)-registered aerial imagery is increasingly of interest. Datasets like Auto Arborist [6], enable the computer vision community to investigate automated methods for tree genus classification from aerial imagery at scale, containing images and genus labels for over 1M individual trees [6].

1.2 Core Challenges

Automated tree classification in aerial imagery is inherently difficult and is associated with many fundamental challenges not present in typical clean, academic datasets. Among the core challenges are the following:

1. **Noisy labels.** Images are commonly mislabeled: genus classification is difficult and requires specialized expertise, GPS localization from the ground can be in error, there are often multiple trees within a single image with only a single label, and temporal inconsistencies can occur as trees are not imaged and labeled at the same time, leading, *e.g.*, to some locations being imaged after the trees have died. We visualize sources of

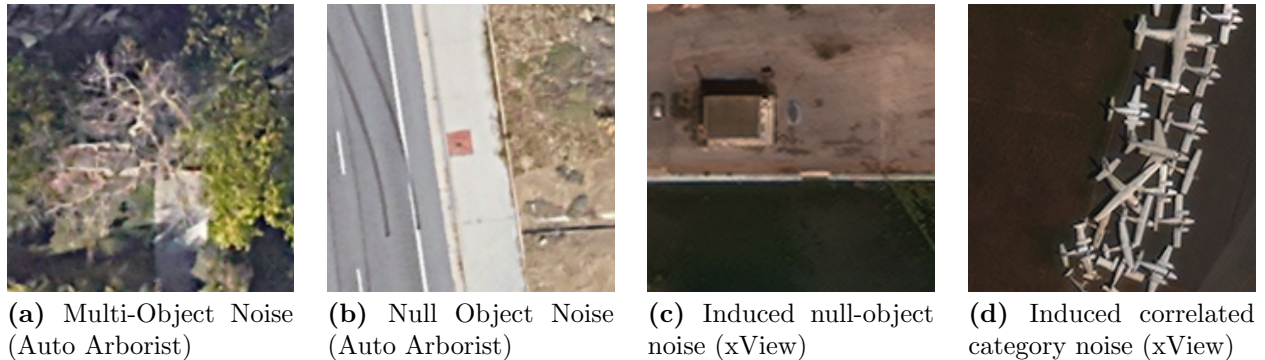


Figure 1.1: Real and Induced Noise. Auto Arborist is a naturally noisy dataset due to its data collection process. Here we demonstrate two kinds of noise: multi-object noise (left) and null noise (no object, center left). In xView, we show induced noise on the container crane (center right) and airplane hangar (right) categories. On the right image, we see that a collection of planes, which is likely correlated with airplane hangars.

noise in Auto Arborist in fig. 1.1 (with xView) and fig. 4.2. We also visualize clean images in fig. 1.2.

2. **Non-IID data.** Geospatial data also breaks the typical deep learning assumption that data will be independent and identically distributed (IID) as spatially close examples often contain correlations. For example, trees are often planted in groups (*e.g.* a row of cherry trees along the same street). We visualize the geo-spatial distribution of samples in fig. 3.1.
3. **Fine-grained and long-tailed class distribution.** Tree classification is fine-grained, with only subtle differences between many genera, and the distribution of trees is long-tailed, a known challenge in machine learning [80, 15, 39]. These characteristics tend to skew classification models towards predicting predominant classes.
4. **Geospatial generalization under distribution shift.** Finally, to be maximally impactful, a model must generalize geospatially as accurately categorizing trees in areas where censuses are already available (and thus used for training data) is of less value than generating a census in an area that has not been able to afford a ground-level census. We visualize performance under these distribution shifts in fig. 5.1.

1.3 Contributions

Based on the intuition that geospatial similarity is inherent to many aerial object classification problems, we choose to focus on graph neural networks (GNNs) [5] to capture and make use of this structure by efficiently providing shared spatial context across nearby objects.



Figure 1.2: Examples of clean imagery. Here we visualize examples of clean imagery from Auto Arborist: there is typically one prominent tree, the image is well lit, and the tree typically has features that are identifiable. For example Phoenix trees (bottom right) are distinctly palm trees.

Prior applications of GNNs to aerial data either formulate large aerial image tiles as graphs (*i.e.* pixels are nodes) or adopt super-pixel-based image decompositions to create nodes in the graph [17, 4, 34]. Instead, we define cropped regions of larger image tiles corresponding to specific objects within a geographic region as nodes in our graph, and add edges between all pairs of cropped images within a defined Euclidean distance radius of each other (in latitude/longitude coordinates). Note that this approach requires GPS location metadata to be available for each image crop. Liang *et al.* [40] propose a similar object graph where they jointly optimize GNN predictions on the object graph and convolutional neural network predictions on larger image tiles. In contrast, we extract per-crop features (explained in detail below), and use a Graph Attention Transformer (GAT) across these features to selectively aggregate context between relevant neighbors [9].

To address learning from noisy labels, we extract features using the DivideMix [36] framework, which frames learning with noisy labels as semi-supervised learning, exploiting the phenomenon that “neural networks learn patterns first, then memorize” [2, 24]. While Di-

videMix is very effective in many noisy data settings, when combined naïvely with GATs, we show that the combined approach can lead to undesirable tail class smoothing. More specifically, we observe that features learned via the DivideMix framework end up being highly “class-indicative,” meaning that they lead to strong predictions with a simple (*e.g.* linear) model (see section 6.2 for a more formal discussion) on the train set that causes undesirable tail-class smoothing. This is problematic in the graph setting because of homophily (the tendency to have neighbors of the same class): if the model can usually get a strong signal on a target node’s class from its neighbors, it will learn to trust neighbors at the expense of direct information on the target provided by the target features. Therefore, we propose a framework for training GNNs on highly class-indicative features using a temperature-softened teacher penalty and test-time debiasing, achieving better performance than baselines. We make the following contributions:

1. A novel approach to learning graph-aware predictions on overly class-indicative features, which helps prevent GNNs from under-predicting tail classes on long-tailed classification problems.
2. Results and analysis on the relatively understudied technique of supervised feature extraction for GNNs on two datasets: Auto Arborist and xView [35].
3. A rationale for understanding tail class smoothing using DivideMix features (and potentially other supervised feature extractors), and how their overconfidence can harm GNN performance on tail classes with empirical evidence.

Chapter 2

Related Work

2.1 Aerial Image Classification

Neural networks have been used for Aerial Imagery and Remote Sensing as early as 1997, when Atkinson *et al.*, proposed the usage of multi-layer perceptrons (MLPs) to process remote sensing images. [3]. While earlier works studied rougher shapes such as the types of clouds, later works extend to smaller objects such as buildings [43] and vehicles [82]. This is a harder problem [39] not only due to the increased amount of noise from the smaller targets of interest, but also due to greater appearance variance of the targets (the same types of clouds may appear more homogeneous in aerial images, while buildings and vehicles may vary based on the lighting conditions, orientation, and location).

2.2 Graph Neural Networks (GNNs)

GNNs are well suited to processing data that have some inherent relationships which can be represented in a graph structure [78]. In particular, we are interested in modeling local correlations between tree genera in urban forests.

GNNs have demonstrated competence in remote sensing for their ability to use "spatio-topological relationships" [38]. Li et. al designed a framework where a GNN was used as a pooling operation on information gathered by trained CNNs. This framework was evaluated on the UCM multi-label dataset and AID multi-label dataset [30], which contain aerial remote sensing images of various objects including airplanes, buildings, cars, etc. Experiments demonstrate that this framework outperforms baseline CNNs [38]. We adopt a modified Graph Structure in our network that we discuss more in section 4.1.

2.3 Object Detection

GeoGraph [47] is a framework that uses a GNN and an end-to-end learning pipeline for multi-view object detection, re-identification, and location using only street-level images. Notably, one of two datasets the paper uses for validating GeoGraph is the Pasadena Multi-View ReID dataset [46], a multi-view dataset of 6,020 trees from Pasadena, California.

2.4 Feature Extraction for GNNs

Common feature extraction approaches for graph-valued data often adopt unsupervised feature extraction with large graph-agnostic models to learn a useful representation of nodes. For example, state-of-the-art models on the Open Graph Benchmark (OGB) node-property prediction tasks [29], use language models or other forms of unsupervised learning to extract relevant features from raw data [19, 81, 83]. Others have proposed jointly optimizing a representation and the graph predictor *e.g.* Shi *et al.* [59]. However, incorporating graph information into node feature extraction may impact noisy sample separation. Instead, we use a two-staged approach: first features are extracted, then a predictor GNN is trained on the learned representation. Hu et al explored combining GNN and CNN for classification of hyperspectral image [17], images that contain information beyond the visible spectrum. These images usually have many channels to produce more detailed readings. To combine the information of each superpixel with the relative locations of superpixels, a 2D convolution was used with a 1D convolution depth wise. The features are processed by a multi-scale CNN and a GNN that treats the image as a graph to further emphasize information passing. The two networks finally aggregate their data to produce a pixel-wise segmentation for hyperspectral image classification.

2.5 Learning with Noisy Labels

With a dataset such as Auto Arborist, it is not sufficient to naively train a supervised classifier on the existing labels, which may be incorrect or misleading. To address this problem, Li et al [36] proposed DivideMix, which uses a Gaussian mixture model on a network’s per-sample loss to isolate noisy samples. This method was shown to be a ”substantial improvement” over all baselines over noise/method ratios ranging from 20% to 90%. DivideMix also achieved 75.7% accuracy on the real-world Clothing1M dataset, an increase of over 1% compared to the next best method. Wei et al presented RoLT: Robust Long-Tailed Learning [74], which improved the long-tailed performance of DivideMix. Many prior works in the space consider the IID case, where samples are exchangeable. This assumption doesn’t hold in our setting due to local correlations between samples, however, these approaches are still strong base learners.

2.6 Fine-grained Classification

Fine-grained classification is the problem of categorizing examples in classes that are sub-categories of certain supercategories [75]. Prior approaches typically require supervision to achieve strong results, though unsupervised pretraining can be beneficial [26]. The iNaturalist dataset [28] provides a standard large-scale benchmark for fine-grained classification. Top approaches on this dataset require supervised finetuning to achieve strong performance, as shown by Srivastava and Sharma [61]. We adopt ImageNet [16] pretraining for each of our models.

2.7 Test Time Adaptation and Debiasing

One way of dealing with distribution shift is by test time adaptation which is the process of adjusting a model "from the source domain to unlabeled data in the target domain" [41, 63]. Types of test-time adaptation include source-free domain adaptation, test-time batch adaptation, and online test-time adaptation [41]. In test-time domain adaptation, pseudo-labels can be used to adapt the trained model to the new domain [55, 13, 62]. Wang *et al.* [70] show that bias and class imbalance can appear in psuedo-labeling. Thus, Wang *et al.* present DebiasPL, which uses counterfactual reasoning and adaptive margins to deal with the imbalance in self-supervised and zero-shot learning. We combine these ideas by debiasing predictions using test-time statistics.

2.8 Tree Canopy Coverage Estimation

Tolan *et al.* [66] recently released a high resolution map of canopy coverage across the world using self-supervised DINO-style pretraining on aerial imagery followed by Lidar supervised height prediction. Their approach highlights another important area of Earth observation and less granular tree cataloging. It also shares resemblance to our approach in it's use of metadata like latitude and longitude to inform predictions. Canopy coverage in urban areas also represents an interesting axis of distribution shift in the Auto Arborist dataset. In Los Angeles, for examples, canopy coverage varies significantly geographically [14] which is correlated with health outcomes and socioeconomics. Visually, north LA (training set) has denser tree coverage than south LA (validation set), however, because Auto Arborist only catalogs public trees, we see the inverse *i.e.* that examples in the training set have a lower average degree than the validation set. This is likely due to smaller and more regular-shaped blocks in south LA—an artifact of how we construct the graph between trees as described in chapter 3.

Chapter 3

Problem Statement

We consider classification into C possible genera on image chips: square tiles, typically only meters in length/width, representing crops from larger aerial image tiles associated with individual objects of interest. Specifically, we predict a class label $y_i \in \{0, \dots, C - 1\}$ of a chip v_i .

We construct a radius graph (fig. 3.1) *i.e.* connecting images that are within a certain ℓ_2 distance of each other in latitude/longitude space. Thus, we have a graph-valued problem

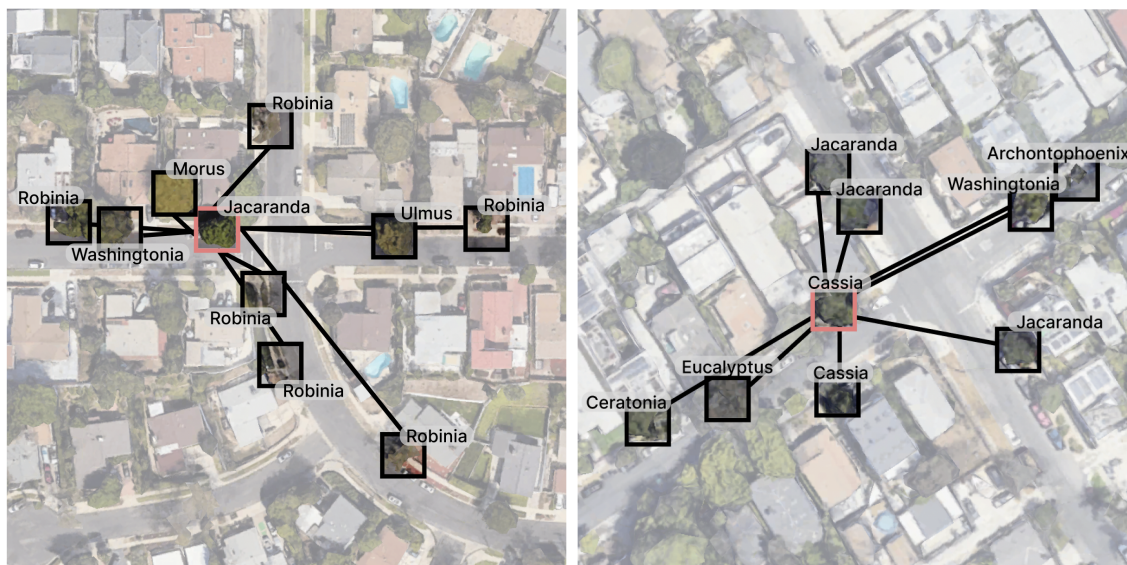


Figure 3.1: Graph Construction. To capture the surrounding context for images, we construct a radius graph using GPS coordinates on Auto Arborist. To visualize the full context of examples, we use Google Earth: the red box is the target tree and the black boxes are its identified neighbors. This also shows the bias of the Auto Arborist dataset, which only catalogs trees on public land. We also see homophily in the graph as similar trees are often found on the same street.

$G = (V, E)$ with V as the set of images (chips) in the dataset and E as the links connecting these images. We additionally expect our dataset to exhibit the core challenges in section 1.2, making the problem more challenging than standard node classification on a graph.

In addition, our dataset often contains missing classes in various training and testing splits (*i.e.* a class exists in the train set, but not the test set or vice versa). We handle this by having a classifier produce predictions on the set of all possible classes.

Chapter 4

Method

Our method, **S**oftening **H**ead **I**mbalances with **E**ffective **L**earning and **D**ebiasing for **G**raph **N**eural **N**etworks (SHIELD-GNN) includes a noise-robust feature extraction stage followed by a GNN fitting stage. A GNN is fit on the learned representation, regularized by penalizing prediction dissimilarity to softened predictions generated during feature extraction. SHIELD-GNN then applies a test-time debiasing module to remove head-class bias from the final prediction set.

4.1 Preliminaries

Learning on graphs with information-rich node features (*e.g.* image or text data) is often in two stages: node feature extraction, then GNN classification [19, 81, 83].

DivideMix

DivideMix is a popular and effective algorithm to improve classification performance under noise, but assumes that data is exchangeable and IID. SHIELD-GNN uses DivideMix to learn a strong set of node features for classification. DivideMix trains two networks jointly, first warming up the networks using the cross-entropy loss on all samples then uses a two-component Gaussian Mixture Model (GMM) to split samples based on their loss. Samples that are clustered in the Gaussian with a lower mean retain their label, while samples in the high mean Gaussian have their labels removed. We modify this step to ensure entire tail classes are not removed from the labeled data by having class-conditional GMMs split samples. The networks then trade splits and are optimized using the semi-supervised Mix-Match [7] framework. The predictions produced by DivideMix alone cannot reason about the surrounding neighborhood of trees because of the framework’s implicit IID assumption and therefore cannot make context-aware predictions, limiting classification performance. We include more specific implementation details in appendix A.1.

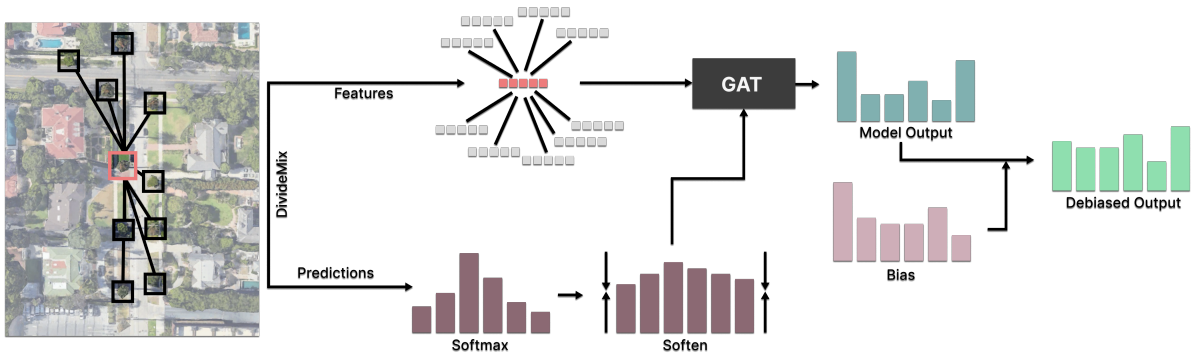


Figure 4.1: The SHIELD-GNN pipeline begins by constructing a radius graph. It then leverages DivideMix for feature extraction and base prediction generation. These elements are used to train a GATv2 model, which outputs a prediction set. Finally, this set is debiased to arrive at the final predictions.

Graph Neural Networks

To exploit the local correlations in this class of problems, we probe node features using a Graph Attention Transformer V2 (GATv2) [9]. GATv2s use an expressive version of attention as their message passing function to aggregate features from nearby nodes. Brody *et al.* find that GATv2s achieve strong performance on standard graph learning benchmarks and are robust to noise, which make the architecture well-suited to our task [9]. We use a single layer GATv2 to aggregate nearby context in the generated radius graphs. We use a single layer network to prevent feature over-smoothing which occurs in shallow networks, but is even more prevalent in deeper GNNs [57].

4.2 SHIELD-GNN

SHIELD-GNN combines feature extraction and GNN prediction to produce strong context-aware predictions. A naive combination of these two components leads to undesirable tail-class oversmoothing as confident neighbors swamp tail predictions. We use the prediction set from the feature extraction to regularize GNN predictions. SHIELD-GNN regulates the confidence of the non-graph predictions using temperature softening, as overconfident base predictions can degrade performance, which we show in the supplementary material via an ablation study. The debiasing module removes globally averaged logits from each prediction weighted by a hyperparameter, λ_D . An overview of the method is shown in fig. 4.1. We include more specific implementation details in appendix A.2.

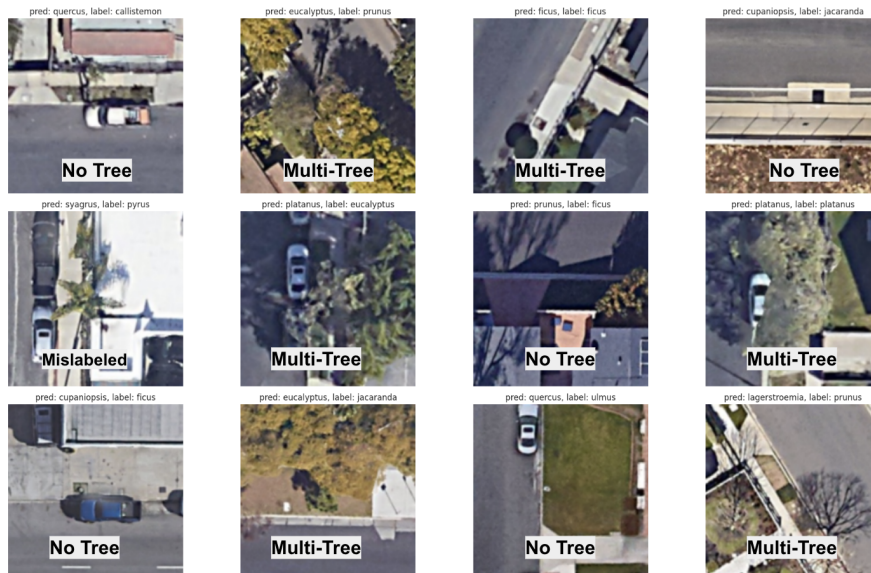


Figure 4.2: Categories of noisy examples from Auto Arborist. From these examples, we can see the wide variety of noise in the dataset, which helps motivate the need for a noise-robust training loop. Although DivideMix was originally created for structured noise (ie label perturbations), it can handle less structured noise (like multiple trees) through pseudolabels.

Feature Extraction

DINO [11] has shown promise in other remote sensing representation learning tasks [72, 71]. However, in our setting, DINO feature extraction is ineffective as it does not extract relevant information to make fine-grained distinctions between genera. We demonstrate this using DINO trained on Auto Arborist and pre-trained DINOv2 without finetuning in section 5.2 (DINOv2 has shown strong generalization ability and was trained on a considerably larger dataset than DINO v1 [51]). We, therefore, use a supervised classifier, DivideMix [36], for feature extraction. While DivideMix achieves strong performance on noisy data, it is prone to producing highly class-indicative features on the train set, which can harm GNN performance.

GNN Prediction

After feature extraction, we train a GNN classification head using a modified Early-Learning Regularization (ELR) loss [42], penalizing the GNN predictions’ dissimilarity to original DivideMix predictions (which we refer to as teacher predictions) rather than the original temporal ensemble of predictions. We find that without the penalty, GNN performance on tail classes suffers. To reduce the impact of DivideMix overconfidence on GNN predictions, we *temperature soften* predictions from the teacher [7].

Let \hat{y}_b be the original predictions from feature extraction and $T \leq 1$ be the temperature softening factor. We define softened predictions as

$$\hat{y}_{b;soft} = \frac{\hat{y}_b^{1/T}}{\sum \hat{y}_b^{1/T}}. \quad (4.1)$$

Next, let $\langle \cdot, \cdot \rangle$ be the standard euclidean inner product, \hat{y}_g be the softmax GAT prediction, y be the original label. Our regularized loss is

$$\mathcal{L}_{SHIELD-GNN}(\hat{y}_{b;soft}, \hat{y}_g, y) = \mathcal{L}_{CE}(\hat{y}_g, y) + \log(1 - \langle \hat{y}_{b;soft}, \hat{y}_g \rangle). \quad (4.2)$$

Debiasing Prediction Sets

We find that GATs systematically overpredict some classes, so we require increased logit imbalance to predict those classes on an inference set.

Our final prediction set $\hat{y}_{g;f}$ on the inference graph $G_{test} = (V_{test}, E_{test})$ is regulated by debiasing strength hyperparameter λ_D . Debiasing is performed on output logits, rather than softmax probabilities.

$$\hat{y}_{g;f} = \hat{y}_g - \frac{\lambda_D}{|V_{test}|} \sum_i^{|V_{test}|} \hat{y}_{g;i}. \quad (4.3)$$

Chapter 5

Experimental Design and Results

5.1 Experimental Setup

We benchmark SHIELD-GNN on two datasets: Auto Arborist [6], and xView [35]. We modify xView to be a classification dataset, similar to Singhal *et al.* [60].

Auto Arborist

The Auto Arborist dataset, is a multi-view, fine-grained visual tree categorization dataset containing images of over 1 million public zone trees from 300 genus-level categories across 23 major cities in the US and Canada (We note that the dataset represents only a portion of the total tree population). From Auto Arborist [6], we focus on identifying tree genera using aerial imagery. Specifically, each tree record in the dataset is associated with a $15\text{m} \times 15\text{m}$, 300×300 pixel RGB aerial image, including latitude and longitude. The method for labeling the tree genus class in the dataset varies across cities, utilizing a combination of volunteer citizen scientists and expert labelers.

The aerial imagery component of the dataset contains many noise sources, such as inaccuracies in labeling due to human error, potentially outdated tree records, inconsistent image quality across different cities, and occlusion of targets. We conduct two sets of experiments to assess our methodology’s performance on a larger benchmark and its robustness to large distribution shifts.

We first run experiments on the top 100 genera from the canonical Los Angeles (LA) training set to demonstrate the effectiveness of each component of our method and to benchmark the method’s effectiveness against relevant baselines. We use the canonical LA test set for validation and all of the Santa Monica for testing, both on the same 100 classes. We run larger scale experiments on the East-Central-West splits defined by Beery *et al.* [6]. We enumerate the constituent cities in the supplementary material. The goal of these experiments is to show that our method doesn’t simply overfit to structural noise in the training split but still outperforms baselines under large distribution shifts. We report statistics on the intersection of genera from the regions.

Dataset	#Examples	#Classes	Imb. Ratio	Homophily	Avg. Degree
AA - LA 100	167k	100	102	.44	15.2
AA - West	573k	261	2.7k	.34	23.0
AA - East	382k	93	4.3k	.34	19.0
AA - Central	179k	82	2.2k	.37	18.7
xView - all	602k	60	17.6k	.69	45.7
xView - head only	528k	2	1.5k	.79	43.7
xView - tail only	74k	58	1.2k	.57	17.8

Table 5.1: Dataset statistics of Auto Arborist (AA) splits and xView. We use the definition of node homophily defined by Pei *et al.* [53] and imbalance (imb.) ratio: the ratio of the largest class size to max(smallest class size, 10). We also include statistics on xView with subgraphs of only the top 2 (head-only) and bottom 58 classes (tail-only).

xView

The xView dataset [35], released in 2018, represents a collection of aerial images (we focus on the RGB bands), encompassing more than 1 million object instances across 60 distinct classes, spanning over 1,400 square kilometers. This dataset was curated from WorldView-3 satellite images, captured with a ground sample distance of 0.3 meters. The classes of objects within the xView dataset includes different vehicle types, structure names, and locations significant to vehicle activity. The data annotation process for the xView dataset involves a manual examination of satellite images to accurately identify, label, and delineate each object with bounding boxes of varying sizes. We augmented the dataset with latitude and longitude features, representing the centroids of each object’s bounding box. We divided the dataset into training, validation, and test sets (approx. 70-15-15) by assigning image tiles to each set.

We use xView to assess the impact of increased noise on the efficacy of our method and show its generality. We perturb the dataset with 30% and 60% noise. To generate object image chips, we begin by inflating bounding boxes by 20 pixels and increasing the minor axis to make the bounding box a square. Then if a sample is noisy (which we determine with IID draws of a uniform random variable), we shift the chip’s image frame in a random direction such that it omits the entire bounding box. This enables us simulate real-world examples of noise: inflating the bounding boxes naturally captures surrounding context and other potentially in-vocabulary objects, shifting the bounding box can mislabel examples and also create null examples.

Baseline: Fixed Top-5 Classifier

As a baseline, we use the Fixed Top-5 classifier, which predicts the top-5 classes from the train set with 100% accuracy and precision and all other classes incorrectly.

Method	Acc	AR	AP	Parameters
DINO	34.0	13.2	13.8	22.2M
DINO + GATv2	39.2	16.9	15.8	22.2M
DINO V2	22.9	7.9	7.9	86M
DINO V2 + GATv2	23.7	9.4	8.7	86M
Cross-Entropy	45.3	18.8	19.8	20.3M

Table 5.2: DINO trained on Auto Arborist LA and DINO V2 without finetuning on the top 100 classes of Auto Arborist LA (validation set).

Metrics

We assess model performance through three main metrics: mean accuracy (Acc.), class-average recall (AR), class-average precision (AP). Given the significant class imbalance in both Auto Arborist and xView, relying solely on accuracy can misleadingly favor models that overpredict the majority classes. We interpret AR and AP jointly: an increase in both corresponds to a stronger prediction rule, but an increase in AP can be easily achieved at the expense of AR by not predicting tail classes unless the model is extremely confident and vice-versa with AR. Because inflated AP is easier to achieve, we primarily look to AR to assess long-tailed performance following [6]. We also report top-5 class accuracy for the Auto Arborist distribution shifts which allows us to assess model performance more clearly in a challenging setting.

Specifically for xView, we emphasize average recall (AR), recognizing that models maintaining reasonably high accuracy while boosting average recall are better equipped to navigate class imbalances and distributional variations. In xView, a model predicting only the two head classes (neither of which are fine-grained categories: small car, building) can easily achieve high accuracy ($\sim 88\%$), which says little about the performance of the model on fine-grained classes. We include these head classes, as a model should be able to handle the extreme but realistic class imbalances exhibited in both datasets.

5.2 Results

In all experiments, we use the same hyperparameters: $r = .001$ (the radius for connecting nodes in latitude/longitude coordinates), $T = .5$, and $\lambda_D = .5$. For xView, we use a two-thirds inverse class-frequency weighted sampler.

Auto Arborist Los Angeles: Top 100 Classes

We conduct two primary sets of experiments on Auto Arborist LA. The first tests unsupervised feature extraction to evaluate if supervised feature extraction is required in our setting.

Category	Method	Val			Test			Parameters
		Acc	AR	AP	Acc	AR	AP	
Baseline	Fixed Top-5	8.2	5.0	5.0	3.4	5.0	5.0	-
	Cross-Entropy	47.7	20.5	21.4	48.7	16.0	20.6	40.6M
Noisy Label (NL)	DivideMix	49.9	23.4	20.8	55.8	18.6	21.9	40.6M
	RoLT	42.2	21.9	20.0	44.8	17.4	20.4	40.6M
Graph-Aware	DMix + GATv2	55.6	22.3	26.1	58.7	16.3	25.7	40.8M
	DMix + SHIELD	55.0	23.8	22.5	62.7	19.4	24.6	40.8M

Table 5.3: Comparison of methods on the top 100 genera from Auto Arborist LA. We also include the Top-5 Classifier (section 5.1).

Method (Test Set)	West-trained				East-trained				Central-trained			
	Acc	AP	AR	Top-5	Acc	AP	AR	Top-5	Acc	AP	AR	Top-5
DivideMix (West)	47.5	17.3	20.6	62.5	15.7	6.4	6.7	32.0	13.5	4.2	4.8	30.2
+SHIELD-GNN (West)	51.7	19.9	20.6	69.0	17.8	7.9	7.3	39.2	14.4	5.4	5.1	35.3
DivideMix (East)	13.9	3.1	2.2	30.1	38.0	16.3	16.4	58.6	17.9	5.9	5.9	37.8
+SHIELD-GNN (East)	16.1	3.4	2.4	35.0	40.4	19.8	15.8	64.8	18.6	8.3	6.4	42.3
DivideMix (Central)	9.7	2.8	2.0	21.7	19.8	7.1	6.5	40.3	46.3	18.0	13.9	65.5
+SHIELD-GNN (Central)	10.1	2.8	2.1	26.3	20.7	8.4	6.5	47.1	47.1	20.8	14.1	71.5

Table 5.4: Comparison of methods on robustness to distribution shifts in Auto Arborist. The diagonal blocks correspond to validation accuracies (*i.e.* West-trained and West-tested).

This experiment uses DINO [11] finetuned on Auto Arborist and DINOv2 [51] without finetuning. Results are shown in section 5.2. From this experiment, it’s clear the DINO features are ill-suited for this problem, as with GNN probing they don’t match naive cross-entropy performance.

In the second set of experiments (section 5.2), we explore methods to handle the long-tailed and noisy nature of the dataset to find the most effective base model, and add graph network prediction on extracted features to compare naive probing with SHIELD-GNN. We outperform baselines and can generalize under smaller geospatial distribution shifts present in the canonical train-test splits from Auto Arborist. We visualize validation performance spatially in fig. 5.1.

Auto Arborist: Full Dataset

We conduct two types of experiments on the entire Auto Arborist dataset: (1) in-distribution evaluations of SHIELD-GNN and the baseline method it was trained on, (2) we also test the model on a different region’s evaluation sets. The goal of assessing model robustness under

Method	30%			60%		
	Acc	AR	AP	Acc	AR	AP
Fixed Top-5	90.7	8.3	8.3	90.7	8.3	8.3
Cross Entropy	89.7	30.9	37.5	88.5	29.1	37.9
DivideMix	84.7	38.0	32.4	82.7	34.2	27.8
DMix + SHIELD-GNN	85.5	39.0	33.9	83.5	34.9	27.2

Table 5.5: Comparison of noisy-data methods on xView with noise perturbation. We include the baseline Top-5 Fixed classifier (section 5.1) to emphasize that accuracy is a relatively uninformative metric when class imbalance is so extreme.

distribution shift is two-fold: first to assess robustness, but also to verify that SHIELD-GNN has not simply overfit to structural noise. For example, a GNN may learn structural patterns in a neighborhood enabling it to better predict noisy samples (which are present in test sets). This performance gain is undesirable, but by subjecting the model to large distribution shifts, we can show that the model is likely learning a general signal. In section 5.2 we show that SHIELD-GNN both achieves strong performance on the canonical test distribution and exceeds the baseline out of distribution, suggesting that SHIELD-GNN is not overfitting to a region-specific type of noise.

xView

On xView, we present experiments with 30% and 60% injected training noise (section 5.2). In both experiments, SHIELD-GNN is able to achieve higher accuracy and AR than naive DivideMix, though in the 60% noise setting AP is degraded. This is a failure mode of the debiasing module. Increasing the number of tail predictions often affects precision negatively as the classifier’s debiased predictions are less likely to be accurate than very confident prior tail predictions.

The cross-entropy classifier performs quite well in terms of accuracy and AP on both dataset versions. This is an artifact of xView’s extreme imbalance, where two classes make up roughly 88% of the dataset and so a classifier can achieve high accuracy and AP by selectively predicting tail classes, as can be seen from their significantly lower AR. We show this using the Top-5 guessing classifier.

5.3 Ablation Studies

We present a study on the backbone architecture we used for SHIELD-GNN and an ablation study on each component of the pipeline.

In section 5.3 we test backbone architectures on Auto Arborist - LA. From this study, we see that although scale does help, these improvements are generally smaller than gains from

Method	Val			Test			Params
	Acc	AR	AP	Acc	AR	AP	
ViT-b-16	44.1	19.3	18.0	46.4	14.3	17.5	86M
ViT-b-32	40.1	16.0	15.8	40.1	12.4	15.9	86M
EfficientNetV2-S	45.3	18.8	19.8	49.0	15.1	19.1	20.3M
EfficientNetV2-L	47.1	20.0	20.1	50.6	15.5	19.5	117M
ResNet50	41.9	16.8	17.5	44.8	12.9	18.1	23M

Table 5.6: Comparison of architectures on Auto Arborist LA using the cross-entropy loss. We adopt the EfficientNet V2-S as our backbone because of its small size and relatively strong performance.

Method	Val			Test		
	Acc	AR	AP	Acc	AR	AP
Baseline	49.9	23.4	20.8	55.8	18.6	21.9
GATv2	55.6	22.3	26.1	58.7	16.3	25.7
+Debiasing	55.6	23.3	23.8	61.4	18.3	25.7
+Teacher	55.4	23.1	23.0	61.6	18.8	25.3
+ Both	55.0	23.8	22.5	62.7	19.4	24.6

Table 5.7: We ablate each component of the SHIELD-GNN pipeline on Auto Arborist LA. Baseline refers to the predictions produced by DivideMix, GATv2 refers to a GATv2 trained with the cross-entropy loss, +Debiasing refers a GATv2 with test-time debiasing only, +Teacher refers to a GATv2 with the teacher penalized loss only, and +Both refers to the full SHIELD-GNN pipeline.

a robust training loop like DivideMix. Next, we ablate each portion of the SHIELD-GNN pipeline (section 5.3). From the ablation study, we show that each portion of the pipeline improves average recall, though there is a trade-off between validation accuracy and average recall. However, on unseen data, SHIELD-GNN improves both average recall and accuracy, showing that the trade-off is likely useful in this domain.

Hyperparameter Sensitivity

We evaluate the SHIELD-GNN’s hyperparameter sensitivity on Santa Monica, the test set from Auto Arborist LA experiments (table 5.8, table 5.9). The model’s performance is qualitatively smooth across the searched space and the selected hyperparameters achieve a reasonable trade-off of accuracy and average recall on test data. This indicates that SHIELD-GNN is likely not very sensitive to small hyperparameter perturbations.

Temperature (T)	Debias λ				
	0	0.25	0.5	0.75	1.0
4	16.4	17.6	18.4	18.8	19.2
2	17.1	18.0	18.9	19.5	19.5
1	18.3	18.9	19.2	19.6	19.7
0.5	18.8	19.3	19.4	19.6	20.7
0.25	18.7	19.1	19.3	19.5	20.6

Table 5.8: Class-average recall on the Auto Arborist LA test set (Santa Monica), computed over a grid of possible debias λ and temperature T .

Temperature (T)	Debias λ				
	0	0.25	0.5	0.75	1.0
4	58.8	60.3	61.5	61.8	60.5
2	59.4	60.9	61.9	62.3	61.1
1	60.6	61.4	62.0	62.3	61.5
0.5	61.6	62.3	62.7	62.7	62.1
0.25	61.5	62.1	62.7	62.5	62.1

Table 5.9: Accuracy on the Auto Arborist LA test set (Santa Monica), computed over a grid of possible debias λ and temperature T .

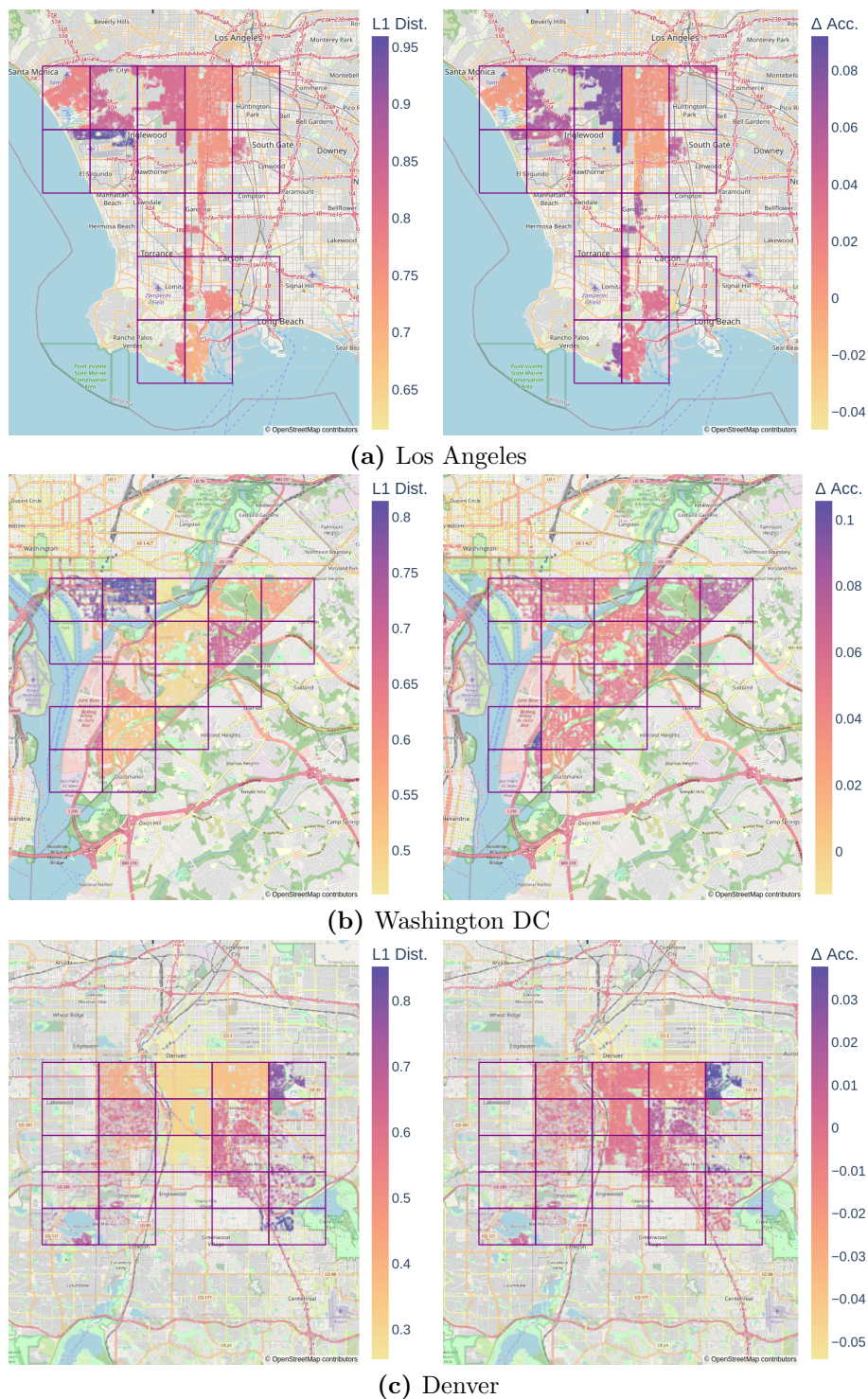


Figure 5.1: SHIELD-GNN performance visualized spatially on Los Angeles (top), Washington DC (Middle), and Denver (Bottom). We show ℓ_1 label distribution distance of test set chunks from the train set (left) and the change in chunk accuracy from DivideMix to SHIELD-GNN (right). We often see the largest increases in accuracy in sections where the distribution distance is high, suggesting that SHIELD-GNN helps improve robustness to small geospatial shifts. We use OpenStreetMap [50] for visualization.

Chapter 6

Discussion

6.1 Distribution Shift in Auto Arborist

As Auto Arborist Dataset is a large catalog trees across various cities, there is considerable distribution shift and bias in the dataset. Beery *et al.* [6] consider the label distribution shift in their initial work. We reproduce this on the public subset of the dataset in fig. 6.1. However, they note there are other sources of distribution. For example, in fig. 6.3, we can qualitatively see that cities can have visually distinct aerial appearances. Similarly, even within a genus there can be significant visual differences amongst species, which can fall along geographic lines. For example, the *Platanus* genus has many visually distinct species in North America [10], including the California Sycamore (*Platanus racemosa*) found primarily in the American Southwest [18] and the American Sycamore, which is found along the Atlantic Coast [77]. Variation like this helps explain the relatively poor model generalization over large geospatial variations in section 5.2.

Beery *et al.* [6] also introduce a notion of "aerially distinctive" classes *i.e.* trees which are easier to recognise from aerial images and consequently are easy for a model to recognise. We demonstrate this phenomenon coarsely using class accuracies in LA in fig. 6.4.

6.2 "Class-Indicative" Features

We propose the concept of "class-indicative" features to help explain DivideMix's tail-smoothing on graph-valued data. We define class indicative features as having a significant ($>\sim 60\%$) proportion of zero or near-zero of linear softmax regression loss, $\mathcal{L}_{CE}(\hat{y}, y) = -\sum_c y_c \log \hat{y}_c$, for some class assignment y . Given a loss threshold τ , let x_i be an embedding of a dataset with labels y and $f_i(x_i)$ be a softmax regressor of x_i on y . Let the class-indication score be defined as

$$S_i = \frac{1}{|V|} \sum_j^{|V|} (\mathcal{L}_{CE}(f_i(x_i)_j, y_j) < \tau). \quad (6.1)$$

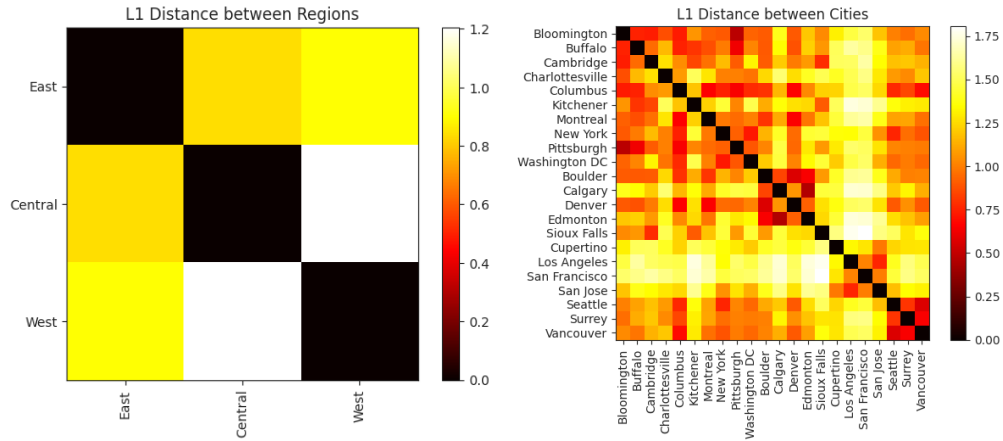


Figure 6.1: ℓ_1 distribution distance between regions (left) and cities (right). We reproduce the figure from Beery *et al.* [6] showing that these large label shifts occur on the public subset, but only represents one axis of distribution shift present in the data.

We say that x_1 is more class-indicative than x_2 if $S_1 > S_2$. This ordering of feature sets shows how easy it is for a linear model to fit to y and consequently how easily a graph neural network could use neighbor features for a target node.

6.3 Motivating SHIELD-GNN

An Interesting Observation

In section 5.2, we show that the DivideMix base predictor (at the same model complexity) performs better than the baseline cross entropy predictor on both accuracy and average recall. Particularly, focusing on AR, we see that there is a difference in tail-class performance of these two methods. Therefore when applying the GNN to this base set of features, we'd expect to see the DivideMix features continue to outperform the cross-entropy features, hopefully by a considerable margin.

However, what we actually observe is that accuracy increases significantly for both models (53.4 validation accuracy for cross entropy), but AR shows a divergence, cross entropy AR goes up to about 22.5%, but DivideMix AR goes *down* to 22.3%, which is even lower than cross entropy features. This is counter-intuitive: the base features are more sensitive to tail classes (and predicts them more often), and yet the GNN is unable to maintain this sensitivity out of the box and instead smooths over those tail predictions.

The Effects of Regularizers on the GNN

The root cause of the poor performance of the GNN on DivideMix features likely involves the network optimization process, so a reasonable first place to check would be regularizing the GNN. We try three different regularizers: label smoothing, weight decay, and dropout. Each swept over a reasonable set of values, yet none seem effective at handling this effect and over-regularization degrades performance (table 6.1). This suggests that the solution to tail oversmoothing is not as simply as applying regularizers.

Noisy Labels Again

The next reasonable hypothesis could be that our network could be suffering from the same issue that plagued our base model: noisy labels. We can effectively test this hypothesis in two ways. We can either bootstrap the GNN using DivideMix, or we can use the ELR loss without modification.

Consider two remedies that involve the original DivideMix pseudolabels and features: first treating the pseudolabels as the target and second weighting examples by the base models' predicted probability of the given label. The first remedy doesn't effectively solve the issue, perhaps because it is easy for the model to simply recover something similar to the original linear predictor as the GATv2 an expressive model [9]. The second remedy also fails to improve the performance of the network and in effect discards a large portion of our datapoints. Next, we can use the ELR loss naively on the GNN to try to isolate and mitigate the impact of the noisy labels. This is also ineffective and the tail class performance still decreases.

We show the performance of each of these potential remedies in table 6.2.

Class Indicative Features

We've now covered two portions of the optimization process: the weights and the labels. We therefore look to the other component of the optimization process: the learned representation. Specifically, we may worry that the learning objective in-sample is becoming too easy because the features are a simple linear transformation away from a low risk set of predictions for many examples. See section 6.2 for a full definition.

Risk Minimization on Class Indicative Features

Broadly, DivideMix (and many learning-with-noisy-labels frameworks [42, 36, 12]) create two cases: predicted clean label and predicted noisy label. In the case where the label is clean, it's easy for the GNN to recover a low loss prediction as it can just mimic a linear model. However, for pseudolabeled examples (which are on average much higher entropy 0.07 vs 0.77) the network can't as easily recover a low loss prediction and therefore it must begin attending more to its neighbors to achieve low loss—leveraging class-indicative neighbors to

Method	Value	Val		Test	
		AR	Acc	AR	Acc
Dropout	0.0	22.3	55.5	16.2	58.5
	0.1	22.3	55.6	16.2	58.3
	0.2	22.4	55.7	16.3	58.8
	0.3	22.3	55.6	16.2	58.3
	0.4	22.3	55.7	16.1	58.0
	0.5	21.9	55.7	16.2	58.2
	0.6	22.4	55.8	16.0	58.4
	0.7	22.4	55.8	16.1	58.3
Label Smoothing	0.0	22.3	55.5	16.2	58.5
	0.1	22.3	55.7	16.3	58.4
	0.2	22.4	55.9	16.3	58.7
	0.3	22.3	55.9	16.6	59.1
	0.4	22.3	56.0	16.5	59.1
	0.5	22.2	56.1	16.5	59.1
	0.6	22.0	56.0	16.6	58.0
	0.7	21.8	56.0	16.5	58.8
Weight Decay	0.0	22.3	55.5	16.2	58.5
	5e-5	22.3	55.6	16.2	58.4
	1e-4	22.3	55.6	16.2	58.4
	5e-4	22.1	55.8	16.2	58.1
	1e-3	22.2	56.1	16.4	58.4
	5e-3	22.3	56.3	16.3	58.7
	1e-2	21.8	56.3	16.0	58.2
	5e-2	17.4	52.9	13.0	54.6
DivideMix	-	23.4	49.9	18.6	55.8
SHIELD-GNN	-	23.8	55.0	19.4	62.7

Table 6.1: We sweep various regularizers (Dropout, Label Smoothing, and Weight Decay) over and report validation and test average recall and accuracy. Note that while the regularizers are able to improve accuracy (by about 1% over not using them), they are unable to meaningfully improve AR, which harms performance out of distribution, on the test set.

	Val		Test	
	AR	Acc	AR	Acc
DMix Pseudo Labels	20.5	51.5	16.0	57.1
DMix Weighting	20.0	51.7	15.2	56.2
ELR Loss	22.4	55.7	16.4	58.6
DivideMix	23.4	49.9	18.6	55.8
SHIELD-GNN	23.8	55.0	19.4	62.7

Table 6.2: We try three potential remedies for noisy labels impacting the GAT’s training: first fitting to DivideMix pseudolabels, next weighting loss by the DivideMix probability of the label, and finally using the ELR loss.

do so. The model is able to attend to its neighbors of the target class and therefore learn a bad prior: to over-rely on neighbors for higher entropy predictions.

Finally, we hypothesize that when the network is taken out of sample, the features are naturally higher entropy and less class-indicative, leading to the network defaulting on relying more on using neighbor predictions. This harms tail classes because of class conditional homophily—*i.e.* that some classes (and tail classes in particular) may be less homophilic than others. We can justify that homophily can be a class-conditional property in our data by plotting node homophily on the Auto Arborist LA (all sets combined) and xView (all sets combined) in fig. 6.7. Let V_i be the set of nodes of class i , $\mathcal{N}(v)$ be the neighbors of a node v , and y_w is the class label of a node w . Class i node homophily is defined as

$$\frac{1}{|V_i|} \sum_{v \in V_i} \frac{|\{(w, v) : w \in \mathcal{N}(v) \wedge y_v = y_w\}|}{|\mathcal{N}(v)|}. \quad (6.2)$$

From fig. 6.7, we see that homophily can vary by class with a positive correlation with log class size.

This distribution shift occurs in DivideMix because the algorithm relies on extensively optimizing predictions for “clean” samples and pseudo-labeled “unclean” samples. This naturally leads to highly class-indicative features in-sample. We visualize the difference between excessively class-indicative features in fig. 6.5, showing that class-indicative features’ clean clusters in-sample do not necessarily persist out-of-sample.

Moderating Selective Attention: Negative Entropy

The simplest way to moderate the effects of class-indicative features is to encourage the features to be more dispersed *i.e.* make the objective harder on average. We demonstrate this in fig. 6.6 using negative entropy penalized classifiers: a set of toy models trained on the objective $\mathcal{L}_{CE} + \beta \sum_i \hat{y}_i \log \hat{y}_i$, which allow us to easily regulate the level of class-indication in the feature set via the entropy weight β .

From fig. 6.6, it's clear that moderating class-indication using the negative entropy penalty is effective and achieves strong AR (fig. 6.6), however this model is not able to achieve competitive performance on accuracy and does not handle the noise in the samples.

Moderating Selective Attention: Teacher Penalty

We turn to another way of forcing the GNN to attend to the target features in all cases—by having the GNN reproduce the base model's predictions using a teacher penalty.

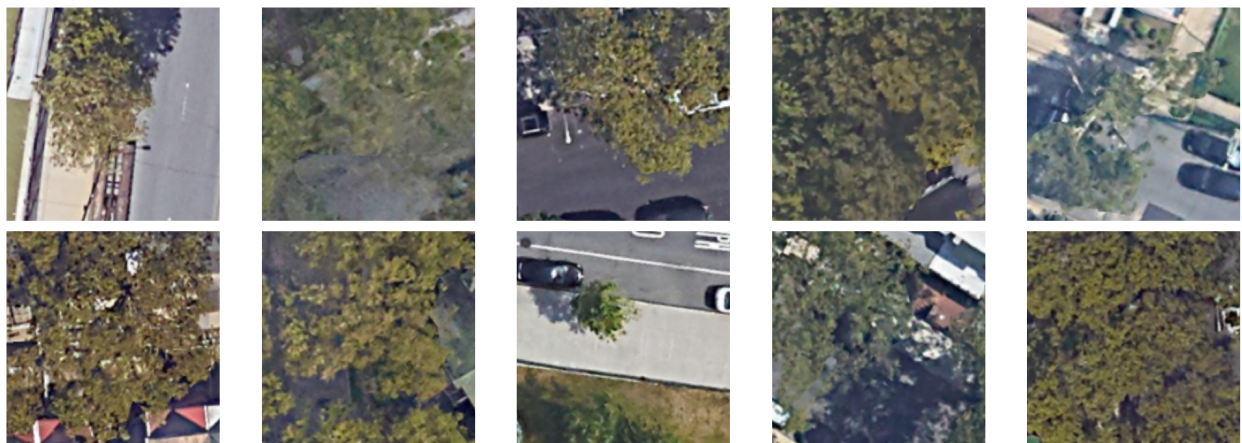
We now conceptually re-analyze the two cases induced by DivideMix. In the first case where DivideMix retained the original label, we see mostly the same behaviour because we can achieve low risk for both the teacher and label loss components just using the target node. However, in the second case, the model must look to it's neighbors to help predict the original label, but is still forced to attend to the target node in order to satisfy the teacher loss. This is shown qualitatively in fig. 6.8, where SHIELD-GNN is typically more consistent with the baseline model, especially in cases of high neighborhood certainty. This approach is able to take advantage of noise robust feature extraction as we do not assume a modification on the features (like the negative entropy penalty) and can proceed to use the robust features.



Figure 6.2: Visual Distribution Shift. We randomly sample 10 aerial images from Los Angeles (top), San Francisco (middle) and New York (bottom) to show qualitative differences between sets of images from each city.



(a) Platanus in Los Angeles



(b) Platanus in New York

Figure 6.3: Within class distribution shift is common in Auto Arborist due to the relatively broad genus level labels. Platanus (Planes tree, Sycamore) trees are an example of this as the genus encompasses are many visually distinct species [10].

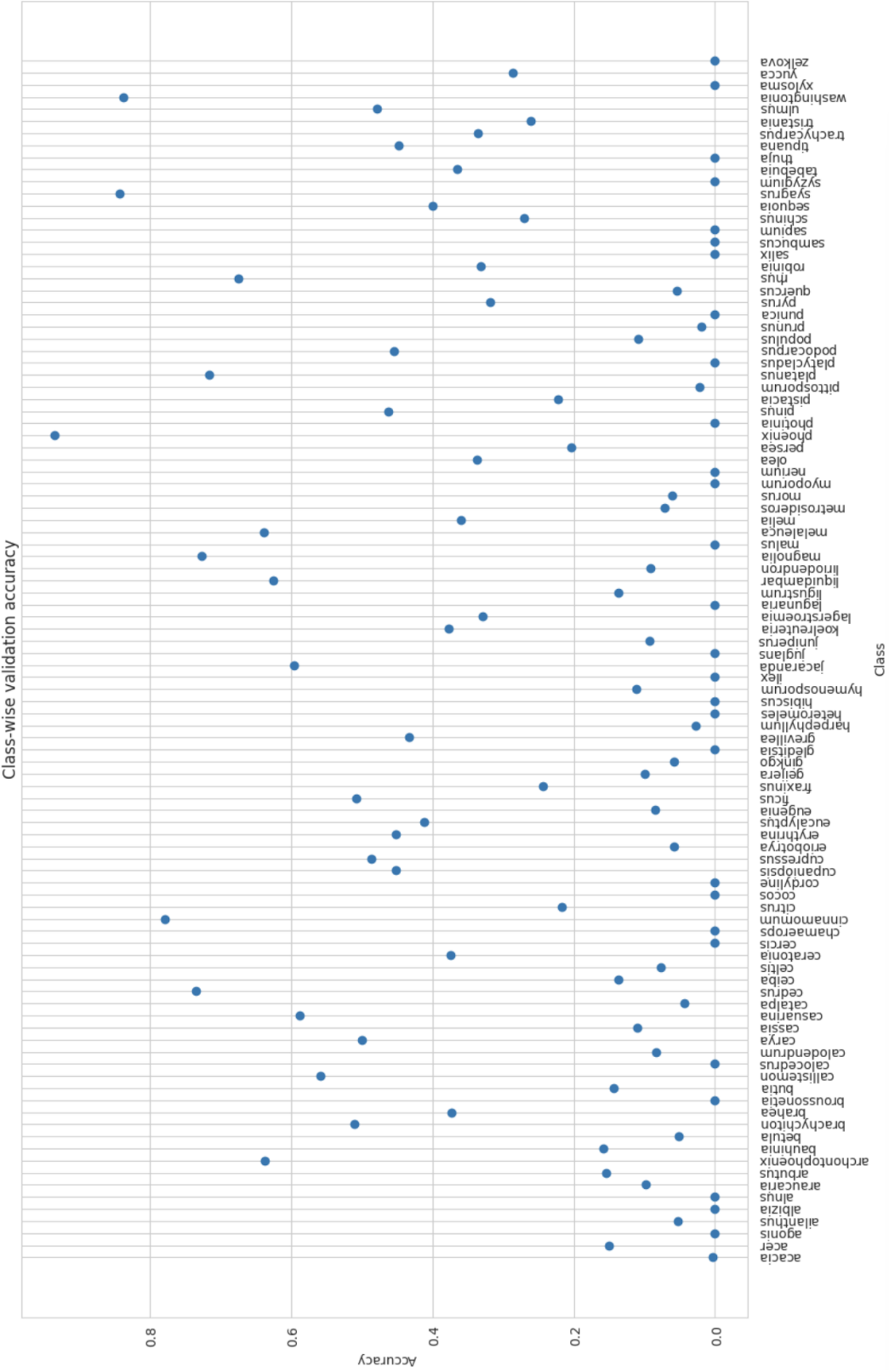


Figure 6.4: Class-wise LA validation accuracy. Highly "aerially distinctive" trees like Phoenix and Washingtonia (both palms) have better validation accuracy than other genera. This indicates that Beey *et al.* were correct and there are some visual differences amongst genera which aid in classification.

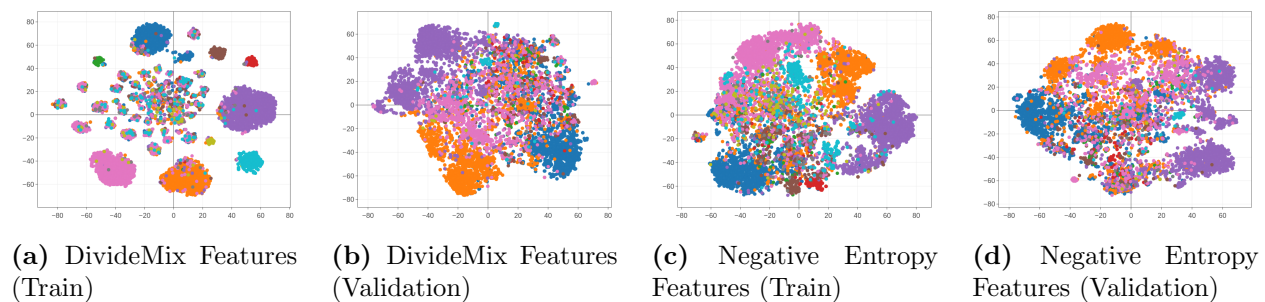


Figure 6.5: t-SNE visualization of the induced distribution shift between the training and validation sets on 10 randomly selected classes from Auto Arborist LA.

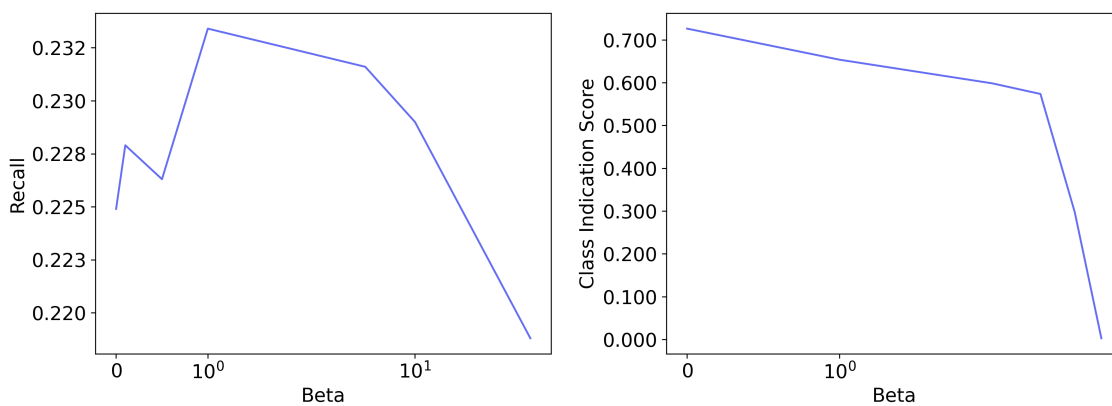


Figure 6.6: Negative entropy penalty and average recall. We show class average recall as a function of the penalty weight (left) in log-scale. Initially, as the penalty increases and we get less class-indicative features improve AR, but then as the penalty starts damaging the representation, we see a corresponding drop off in performance. Class indication scores ($\tau = 1$, right) show that the negative entropy classifier properly regulates class-indicative features.

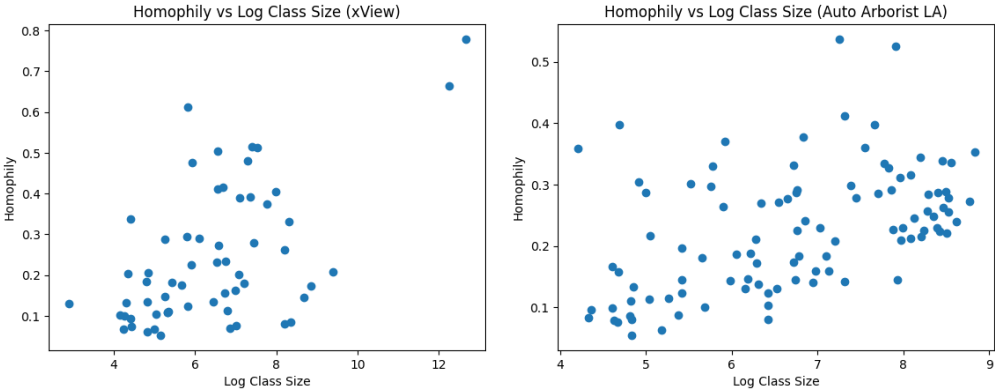


Figure 6.7: Class Conditional Homophily. Here we show homophily (the node-average proportion of neighbors which share the same label as the target node) as a function of log class size. It’s clear that there is significant variance in homophily amongst classes which is correlated with class size. In the case of xView with extreme imbalance, the top two classes are considerably more homophilic than other classes.

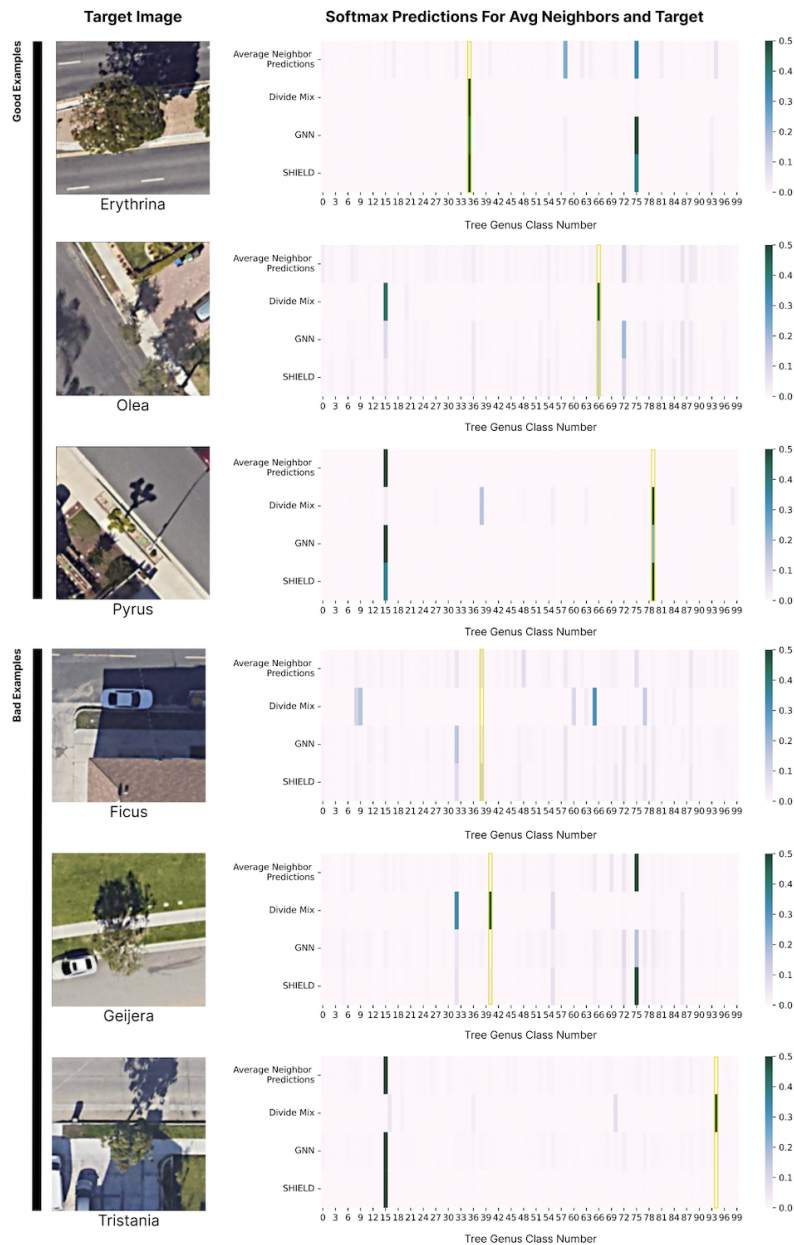


Figure 6.8: Desirable and Undesirable Behavior of SHIELD-GNN. Here we show examples of desirable (top three) and undesirable (bottom three) behaviors of SHIELD-GNN from the validation set. For each example, we show the average neighbor prediction (the softmax of the average of neighbor logits), the base DivideMix softmax prediction, the base GNN softmax prediction, and finally the SHIELD-GNN prediction. The correct genus is highlighted in yellow. In each of the top three examples, SHIELD-GNN produces a prediction which is significantly closer to DivideMix resulting in a more accurate classification. We then display two failure modes in the bottom three images, the first image shows a situation where we correctly predict the genus even though there is clearly no tree. The second failure mode is when SHIELD-GNN does not overcome the neighbor sway and makes a confidently incorrect prediction in line with the neighbors, not the base prediction.

Chapter 7

Limitations

While our approach does achieve good results on Auto Arborist, it’s important to acknowledge limitations. We introduced three hyperparameters: edge radius: r , softening temperature: T , and debiasing weight: λ_D in addition to the hyperparameters included in the feature extraction pipeline. To assess the impact of these hyperparameters, we perform a sensitivity study on Auto Arborist LA in section 5.3.

We also assume object region proposal has been performed to collect candidate chips. In practice, this assumption may be justified by our method’s noise robustness and existing two-stage object detection algorithms, which decouple region proposal and object recognition [25, 56, 22, 23]. Along this same line, the model does not have a null class. However, Vaze *et al.* [67] assert that a strong closed-set classifier is potentially robust in open-set circumstances where a null class may be useful. In addition, the debiasing module and graph structure rely on geospatially clustered inference sets, which can be an unrealistic assumption in some tasks.

Finally, Auto Arborist is a biased dataset—it only catalogs trees that are both on public land and are street-view visible. Consequently, testing the model’s generalization to private land trees will likely prove challenging as existing datasets primarily focus on public land trees [6] or denser non-urban forestry [44]. In chapter 8, we further discuss the challenges and opportunities in this area.

Chapter 8

Extension: Unsupervised Object Discovery

8.1 Motivation

Here, we present the first steps and an evaluation protocol for extending work presented in this thesis into fully automated tree censusing.

The Auto Arborist dataset represents the largest tree census of its kind, but still suffers from unique biases due to its data collection and filtering method [6]. As a result, SHIELD-GNN relies on trees being detected and cropped from high-quality aerial imagery. One such justification for this limitation is existing methods like DeepForest from Weinstein *et al.* [76] or more general instance segmentation foundation models like SAM [33]. However, DeepForest wasn't trained on urban forestry and as such can often miss trees. Similarly, SAM doesn't have a good representation for individual trees and can miss less qualitatively standard-looking trees (fig. 8.1).

We, therefore, look to unsupervised segmentation on Auto Arborist to help domain adapt SAM to tree detection. Wang *et al.* [69] presented CutLER and MaskCut, which use DINO features to perform instance segmentation without supervision. Adapting SAM can be cast as an unsupervised domain adaptation problem [21, 64, 27], where we use the self-supervised representations learned by DINO to adapt SAM to tree crown segmentation.

8.2 Evaluation Protocol Proposal

In their introduction of CutLER, Wang *et al.* show that models trained on these unsupervised MaskCut and DINO generated masks can self-improve over several iterations of training on a prior checkpoint's pseudolabels using a modified loss function to prioritize predictions on under-explored regions. However, in their paper, Wang *et al.* train on YFCC [65] and ImageNet, which may not have the same structural biases as Auto Arborist. This therefore

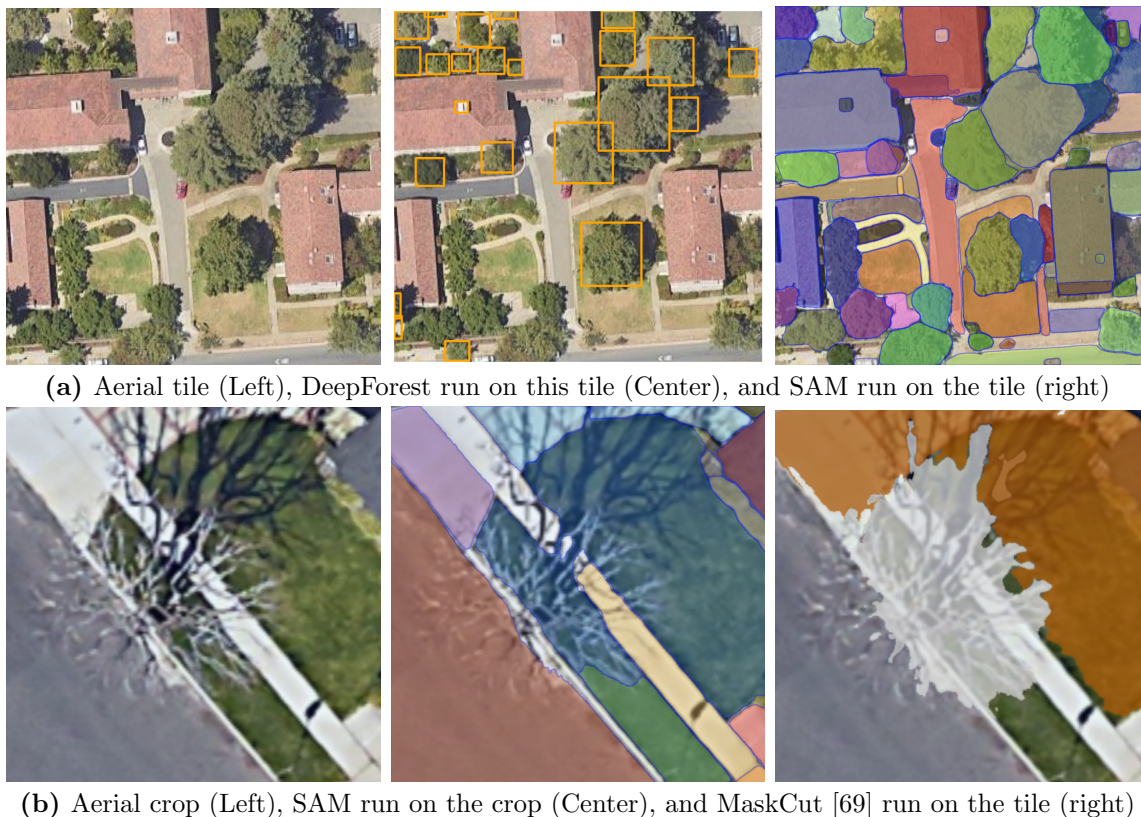


Figure 8.1: Failure Modes of Existing Tree Detection. Here we visualize the failure modes of DeepForest and SAM for tree detection. **(a)** We show that DeepForest often misses trees and is relative sensitive to image quality, whereas SAM can generate more masks, but sometimes groups crowns together. **(b)** However, SAM is also unable to recognize some trees (particularly deciduous trees) and single objects. MaskCut is able to address this, but doesn't produce as robust segmentations as SAM.

suggests an experiment to see a model could overcome the biases of it's training dataset, where we train a model on Auto Arborist and evaluate detections on non-public land trees.

We propose an evaluation protocol for such an experiment below.

Evaluation for this task is challenging as ground-truth data is relatively sparse, however, Ventura *et al.* manually label points on tree crowns in Southern California [68] and Auto Arborist coordinates can provide additional weak supervision. Prior work in this domain often involves evaluation on existing segmentation datasets [54], which doesn't translate to our domain.

Our evaluation protocol begins by collect candidate segmentations on an overhead tile. Then use human-labeled points from Ventura *et al.* to match masks to points by mapping masks to points within their boundaries. We then perform hungarian matching to match points to masks using mask centroids to handle the overlapping masks and many points

issues. However, this scheme could be gamed by very large masks, therefore we add a mask occlusion score, where we penalize overlap between masks *and* masks generated via uniform grid prompting from SAM. The intuition here is that SAM is good at segmentation common objects in urban areas, so intersecting with it's masks is a sign of inflated masks. However, in a case like fig. 8.1 (b), this would unfairly penalize masks of trees which SAM cannot detect. Precision and recall can then be computed on your set of masks and along with occlusion statistics with other masks. However, this does not test the full pipeline of object detection to classification. In order to test this, we propose using Auto Arborist tree (which have coordinates) and trees from city park surveys to test the algorithms full-pipeline accuracy. An example of such a dataset is [Boston's urban tree data](#).

Chapter 9

Conclusion and Future Work

9.1 Conclusion

Classifying objects in aerial imagery is valuable to a diverse set of Earth observation applications, including ecological monitoring, humanitarian aid and disaster response, and urban planning. However, these applications are challenging for computer vision: objects of interest are often fine-grained, the distribution of these objects is often long-tailed, there are significant geospatial distribution shifts, and training and inference data are frequently noisy as ground truth is often captured using ground-level measurements, at a specific point in time, which do not always match what is visible from the air at the time of aerial data acquisition.

In this work, we seek to produce accurate classification of objects in GPS-registered aerial imagery despite these challenges. Prior and related work in both graph structured classification and aerial image classification typically use unsupervised feature extractors, which are not robust to label noise, and do not take advantage of local structure similarities on the graph [73, 49, 19, 15, 45]. We propose SHIELD-GNN, which outperforms baselines by making use of (1) robust initial feature extraction, (2) exploitation of local geospatial structure by using a GNN, and (3) improvement on GNN tail-class performance by reducing the negative impact of class-indicative features for non-homophilic connected components of the graph. We demonstrate the gains of our method on two diverse aerial object classification benchmarks, Auto Arborist which focuses on fine-grained tree genus classification in cities, and xView which focuses on more coarse grained aerial object classification. Our method outperforms DivideMix by up to 7% and standard cross-entropy by 14%, which, in the case of Auto Arborist, corresponds to thousands of trees that would not need to be human-categorized with expensive ground-level surveys, a significant gain for resource-constrained urban planners and arborists.

9.2 Future Work

Broadly, there are three branches of future work which could be motivated by work presented in this thesis. First, there still is much that needs to be done to turn SHIELD-GNN into a usable system for automated tree censusing, for example, tree detection from aerial imagery which is then classified. Next, we have shown that the tail swamping with noise-robust features occurs on Auto Arborist and xView, but we should verify that the phenomenon occurs on other datasets that meet the long-tailed, noisy, and graph-valued criterion outlined in chapter 3. Finally, Auto Arborist is the largest collection of imagery around trees and therefore could serve as an ImageNet-like [16] pretraining scheme for downstream tasks.

Further Validation

We validated the tail class swamping and correction hypothesis on two datasets: Auto Arborist and xView. However, it would be helpful to understand how SHIELD-GNN performs on more standard benchmark datasets. For example, Paper100M [29], a large scale citation graph could help show that this effect occurs at scale. This could help us further analyze the exact conditions required for tail class smoothing in graph-network classification.

Tree Detection

We detail current progress towards tree detection in chapter 8. The goal of this work is to be able to extend the Auto Arborist dataset into a semi-supervised dataset on both public and private land. From this work and with powerful classifiers, urban planners and arborists can get a fine-grained breakdown of complete tree populations in their locale.

Tree Detection Pretraining

To aid in the goal of using Auto Arborist as large-scale pretraining, we have released the weights of models trained on each of the three regions of the dataset and will be releasing weights trained on the entire dataset. The weights and an example of running inference can be found [here](#).

Bibliography

- [1] E. Anyanwu and I. Kanu. “The role of urban forest in the protection of human environmental health in geographically-prone unpredictable hostile weather conditions”. In: *International Journal of Environmental Science and Technology* 3 (Mar. 2006). DOI: 10.1007/BF03325926.
- [2] Devansh Arpit et al. “A Closer Look at Memorization in Deep Networks”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 233–242. URL: <https://proceedings.mlr.press/v70/arpit17a.html>.
- [3] Peter M Atkinson and Adrian RL Tatnall. “Introduction neural networks in remote sensing”. In: *International Journal of remote sensing* 18.4 (1997), pp. 699–709.
- [4] Pedro H. C. Avelar et al. “Superpixel Image Classification with Graph Attention Networks”. In: *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. 2020, pp. 203–209. DOI: 10.1109/SIBGRAPI51738.2020.00035.
- [5] Sohaib Baroud et al. “A Brief Review of Graph Convolutional Neural Network Based Learning for classifying remote sensing images”. In: *Procedia Computer Science* 191 (2021). The 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), The 16th International Conference on Future Networks and Communications (FNC), The 11th International Conference on Sustainable Energy Information Technology, pp. 349–354. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2021.07.047>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050921014447>.
- [6] Sara Beery et al. “The Auto Arborist Dataset: A Large-Scale Benchmark for Multiview Urban Forest Monitoring Under Domain Shift”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 21262–21275. DOI: 10.1109/CVPR52688.2022.02061.
- [7] David Berthelot et al. *MixMatch: A Holistic Approach to Semi-Supervised Learning*. 2019. arXiv: 1905.02249 [cs.LG].

- [8] Leslie Brandt et al. “A framework for adapting urban forests to climate change”. In: *Environmental Science and Policy* 66 (Dec. 2016), pp. 393–402. ISSN: 1462-9011. DOI: 10.1016/j.envsci.2016.06.005. URL: <http://dx.doi.org/10.1016/j.envsci.2016.06.005>.
- [9] Shaked Brody, Uri Alon, and Eran Yahav. *How Attentive are Graph Attention Networks?* 2022. arXiv: 2105.14491 [cs.LG].
- [10] Warren D Brush. “Distinguishing characters of North American sycamore woods”. In: *Botanical Gazette* 64.6 (1917), pp. 480–496.
- [11] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. 2021. arXiv: 2104.14294 [cs.CV].
- [12] Jian Chen et al. “Label-retrieval-augmented diffusion models for learning from noisy labels”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [13] Weijie Chen et al. “Self-Supervised Noisy Label Learning for Source-Free Unsupervised Domain Adaptation”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 10185–10192. DOI: 10.1109/IROS47612.2022.9981099.
- [14] Rachel Connolly et al. “The association of green space, tree canopy and parks with life expectancy in neighborhoods of Los Angeles”. In: *Environment International* 173 (2023), p. 107785. ISSN: 0160-4120. DOI: <https://doi.org/10.1016/j.envint.2023.107785>. URL: <https://www.sciencedirect.com/science/article/pii/S0160412023000582>.
- [15] Tsimur Davydzhenka, Pejman Tahmasebi, and Mark Carroll. “Improving remote sensing classification: A deep-learning-assisted model”. In: *Computers Geosciences* 164 (2022), p. 105123. ISSN: 0098-3004. DOI: <https://doi.org/10.1016/j.cageo.2022.105123>. URL: <https://www.sciencedirect.com/science/article/pii/S0098300422000814>.
- [16] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- [17] Yao Ding et al. “Multi-feature fusion: Graph neural network and CNN combining for hyperspectral image classification”. In: *Neurocomputing* 501 (2022), pp. 246–257. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2022.06.031>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231222007329>.
- [18] A James Downer and Ben Faber. “Effect of Eucalyptus cladocalyx mulch on establishment of California sycamore (*Platanus racemosa*)”. In: *Journal of Applied Horticulture* 7.2 (2005), pp. 90–94.
- [19] Keyu Duan et al. *SimTeG: A Frustratingly Simple Approach Improves Textual Graph Learning*. 2023. arXiv: 2308.02565 [cs.CL].

- [20] Matthias Fey and Jan E. Lenssen. “Fast Graph Representation Learning with PyTorch Geometric”. In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*. 2019.
- [21] Yaroslav Ganin and Victor Lempitsky. “Unsupervised Domain Adaptation by Back-propagation”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 1180–1189. URL: <https://proceedings.mlr.press/v37/ganin15.html>.
- [22] Ross Girshick. *Fast R-CNN*. 2015. arXiv: 1504.08083 [cs.CV].
- [23] Ross Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. arXiv: 1311.2524 [cs.CV].
- [24] John Hartley et al. “Neural networks memorise personal information from one sample”. en. In: *Scientific Reports* 13.1 (Dec. 2023). Publisher: Nature Publishing Group, p. 21366. ISSN: 2045-2322. DOI: 10.1038/s41598-023-48034-3. URL: <https://www.nature.com/articles/s41598-023-48034-3> (visited on 03/01/2024).
- [25] Kaiming He et al. *Mask R-CNN*. 2018. arXiv: 1703.06870 [cs.CV].
- [26] Kaiming He et al. *Masked Autoencoders Are Scalable Vision Learners*. 2021. arXiv: 2111.06377 [cs.CV].
- [27] Judy Hoffman et al. “CyCADA: Cycle-Consistent Adversarial Domain Adaptation”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 1989–1998. URL: <https://proceedings.mlr.press/v80/hoffman18a.html>.
- [28] Grant Van Horn et al. *The iNaturalist Species Classification and Detection Dataset*. 2018. arXiv: 1707.06642 [cs.CV].
- [29] Weihua Hu et al. *Open Graph Benchmark: Datasets for Machine Learning on Graphs*. 2021. arXiv: 2005.00687 [cs.LG].
- [30] Yuansheng Hua, Lichao Mou, and Xiao Xiang Zhu. “Relation Network for Multilabel Aerial Image Classification”. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.7 (2020), pp. 4558–4572. DOI: 10.1109/TGRS.2019.2963364.
- [31] Endalkachew Abebe Kebede et al. “Assessing and addressing the global state of food production data scarcity”. In: *Nature Reviews Earth & Environment* (2024), pp. 1–17.
- [32] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].
- [33] Alexander Kirillov et al. “Segment Anything”. In: *arXiv:2304.02643* (2023).
- [34] Boris Knyazev et al. *Image Classification with Hierarchical Multigraph Networks*. 2019. arXiv: 1907.09000 [cs.CV].

- [35] Darius Lam et al. *xView: Objects in Context in Overhead Imagery*. 2018. arXiv: 1802.07856 [cs.CV].
- [36] Junnan Li, Richard Socher, and Steven C. H. Hoi. *DivideMix: Learning with Noisy Labels as Semi-supervised Learning*. 2020. arXiv: 2002.07394 [cs.CV].
- [37] Tianhong Li, Dina Katabi, and Kaiming He. *Return of Unconditional Generation: A Self-supervised Representation Generation Method*. 2024. arXiv: 2312.03701 [cs.CV].
- [38] Yansheng Li et al. “Multi-label remote sensing image scene classification by combining a convolutional neural network and a graph neural network”. In: *Remote Sensing* 12.23 (2020), p. 4003.
- [39] Zheng Li et al. “Deep Learning-Based Object Detection Techniques for Remote Sensing Images: A Survey”. In: *Remote Sensing* 14.10 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14102385. URL: <https://www.mdpi.com/2072-4292/14/10/2385>.
- [40] Jiali Liang, Yufan Deng, and Dan Zeng. “A Deep Neural Network Combined CNN and GCN for Remote Sensing Scene Classification”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), pp. 4325–4338. DOI: 10.1109/JSTARS.2020.3011333.
- [41] Jian Liang, Ran He, and Tieniu Tan. “A Comprehensive Survey on Test-Time Adaptation under Distribution Shifts”. In: *arXiv preprint arXiv:2303.15361* (2023).
- [42] Sheng Liu et al. “Early-Learning Regularization Prevents Memorization of Noisy Labels”. In: *Advances in Neural Information Processing Systems* 33 (2020).
- [43] Emmanuel Maggiori et al. “Fully convolutional neural networks for remote sensing image classification”. In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2016, pp. 5071–5074. DOI: 10.1109/IGARSS.2016.7730322.
- [44] Sergio Marconi et al. “A data science challenge for converting airborne remote sensing data into ecological information”. en. In: (2019-02-28 00:02:00 2019). DOI: <https://doi.org/10.7717/peerj.5843>. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=925470.
- [45] M.S. Minu and R. Aroul Canessane. “Deep learning-based aerial image classification model using inception with residual network and multilayer perceptron”. In: *Microprocessors and Microsystems* 95 (2022), p. 104652. ISSN: 0141-9331. DOI: <https://doi.org/10.1016/j.micpro.2022.104652>. URL: <https://www.sciencedirect.com/science/article/pii/S0141933122001867>.
- [46] A. Nassar, S. Lefevre, and J. Wegner. “Simultaneous Multi-View Instance Detection With Learned Geometric Soft-Constraints”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2019, pp. 6558–6567. DOI: 10.1109/ICCV.2019.00666. URL: <https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00666>.

- [47] Ahmed Samy Nassar et al. “GeoGraph: Graph-Based Multi-view Object Detection with Geometric Cues End-to-End”. en. In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Vol. 12352. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 488–504. ISBN: 978-3-030-58570-9 978-3-030-58571-6. DOI: 10.1007/978-3-030-58571-6_29. URL: https://link.springer.com/10.1007/978-3-030-58571-6_29 (visited on 11/13/2023).
- [48] David J Nowak and Eric J Greenfield. “US Urban Forest Statistics, Values, and Projections”. In: *Journal of Forestry* 116.2 (Mar. 2018), pp. 164–177. ISSN: 1938-3746. DOI: 10.1093/jofore/fvx004. URL: <http://dx.doi.org/10.1093/jofore/fvx004>.
- [49] Masanori Onishi and Takeshi Ise. “Explainable identification and mapping of trees using UAV RGB image and deep learning”. In: *Scientific Reports* 11 (Jan. 2021). DOI: 10.1038/s41598-020-79653-9.
- [50] OpenStreetMap contributors. *Planet dump retrieved from <https://planet.osm.org>*. <https://www.openstreetmap.org>. 2017.
- [51] Maxime Oquab et al. *DINOV2: Learning Robust Visual Features without Supervision*. 2024. arXiv: 2304.07193 [cs.CV].
- [52] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. arXiv: 1912.01703 [cs.LG].
- [53] Hongbin Pei et al. *Geom-GCN: Geometric Graph Convolutional Networks*. 2020. arXiv: 2002.05287 [cs.LG].
- [54] Lu Qi et al. *PointINS: Point-based Instance Segmentation*. 2021. arXiv: 2003.06148 [cs.CV].
- [55] Sanqing Qu et al. “BMD: A General Class-Balanced Multicentric Dynamic Prototype Strategy for Source-Free Domain Adaptation”. In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan et al. Cham: Springer Nature Switzerland, 2022, pp. 165–182. ISBN: 978-3-031-19830-4.
- [56] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *CoRR* abs/1506.01497 (2015). arXiv: 1506.01497. URL: <http://arxiv.org/abs/1506.01497>.
- [57] T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. *A Survey on Oversmoothing in Graph Neural Networks*. arXiv:2303.10993 [cs]. Mar. 2023. DOI: 10.48550/arXiv.2303.10993. URL: <http://arxiv.org/abs/2303.10993> (visited on 02/29/2024).
- [58] F. S. Jr. Santamour. “TREES FOR URBAN PLANTING : DIVERSITY UNIFORMITY , AND COMMON SENSE”. In: 1999. URL: <https://api.semanticscholar.org/CorpusID:45651160>.
- [59] Zhihao Shi et al. *Label Deconvolution for Node Representation Learning on Large-scale Attributed Graphs against Learning Bias*. 2023. arXiv: 2309.14907 [cs.LG].

- [60] Utkarsh Singhal et al. *Multi-Spectral Image Classification with Ultra-Lean Complex-Valued Models*. 2022. arXiv: 2211.11797 [cs.CV].
- [61] Siddharth Srivastava and Gaurav Sharma. *OmniVec: Learning robust representations with cross modal sharing*. 2023. arXiv: 2311.05709 [cs.CV].
- [62] Tao Sun, Cheng Lu, and Haibin Ling. “Prior Knowledge Guided Unsupervised Domain Adaptation”. In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan et al. Cham: Springer Nature Switzerland, 2022, pp. 639–655. ISBN: 978-3-031-19827-4.
- [63] Yu Sun et al. “Test-Time Training with Self-Supervision for Generalization under Distribution Shifts”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 9229–9248. URL: <https://proceedings.mlr.press/v119/sun20b.html>.
- [64] Yu Sun et al. “Unsupervised Domain Adaptation through Self-Supervision”. In: *CoRR* abs/1909.11825 (2019). arXiv: 1909.11825. URL: <http://arxiv.org/abs/1909.11825>.
- [65] Bart Thomee et al. “YFCC100M: the new data in multimedia research”. In: *Communications of the ACM* 59.2 (Jan. 2016), pp. 64–73. ISSN: 1557-7317. DOI: 10.1145/2812802. URL: <http://dx.doi.org/10.1145/2812802>.
- [66] Jamie Tolan et al. “Very high resolution canopy height maps from RGB imagery using self-supervised vision transformer and convolutional decoder trained on aerial lidar”. In: *Remote Sensing of Environment* 300 (Jan. 2024), p. 113888. ISSN: 0034-4257. DOI: 10.1016/j.rse.2023.113888. URL: <http://dx.doi.org/10.1016/j.rse.2023.113888>.
- [67] Sagar Vaze et al. *Open-Set Recognition: a Good Closed-Set Classifier is All You Need?* 2022. arXiv: 2110.06207 [cs.CV].
- [68] Jonathan Ventura et al. *Individual Tree Detection in Large-Scale Urban Environments using High-Resolution Multispectral Imagery*. 2022. arXiv: 2208.10607 [cs.CV].
- [69] Xudong Wang et al. “Cut and learn for unsupervised object detection and instance segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3124–3134.
- [70] Xudong Wang et al. “Debiased Learning from Naturally Imbalanced Pseudo-Labels”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 14627–14637. DOI: 10.1109/CVPR52688.2022.01424.
- [71] Yi Wang, Conrad M Albrecht, and Xiao Xiang Zhu. *Self-supervised Vision Transformers for Joint SAR-optical Representation Learning*. 2022. arXiv: 2204.05381 [cs.CV].
- [72] Xinye Wanyan et al. *DINO-MC: Self-supervised Contrastive Learning for Remote Sensing Imagery with Multi-sized Local Crops*. 2023. arXiv: 2303.06670 [cs.CV].

- [73] Emily Waters, Mahdi Maktabdar Oghaz, and Lakshmi Babu Saheer. *Urban Tree Species Classification Using Aerial Imagery*. 2021. arXiv: 2107.03182 [cs.CV].
- [74] Tong Wei et al. *Robust Long-Tailed Learning under Label Noise*. 2021. arXiv: 2108.11569 [cs.LG].
- [75] Xiu-Shen Wei et al. *Fine-Grained Image Analysis with Deep Learning: A Survey*. 2021. arXiv: 2111.06119 [cs.CV].
- [76] Ben G Weinstein et al. “DeepForest: A Python package for RGB deep learning tree crown delineation”. In: *Methods in Ecology and Evolution* 11.12 (2020), pp. 1743–1751.
- [77] OO Wells, JR Toliver, et al. “Geographic variation in sycamore (*Platanus occidentalis* L.)”. In: *Silvae genetica* 36.3-4 (1987), pp. 154–159.
- [78] Zonghan Wu et al. “A comprehensive survey on graph neural networks”. In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.
- [79] Liyao Yuan. “Remote Sensing Image Classification Methods Based on CNN: Challenge and Trends”. In: *2021 International Conference on Signal Processing and Machine Learning (CONF-SPML)*. Nov. 2021, pp. 213–218. DOI: 10.1109/CONF-SPML54095.2021.00048. URL: <https://ieeexplore.ieee.org/document/9706946> (visited on 02/29/2024).
- [80] Yifan Zhang et al. *Deep Long-Tailed Learning: A Survey*. 2023. arXiv: 2110.04596 [cs.CV].
- [81] Jianan Zhao et al. *Learning on Large-scale Text-attributed Graphs via Variational Inference*. 2023. arXiv: 2210.14709 [cs.LG].
- [82] Haigang Zhu et al. “Orientation robust object detection in aerial images using deep convolutional neural network”. In: *2015 IEEE International Conference on Image Processing (ICIP)*. 2015, pp. 3735–3739. DOI: 10.1109/ICIP.2015.7351502.
- [83] Jason Zhu et al. “TextGNN: Improving Text Encoder via Graph Neural Network in Sponsored Search”. In: *Proceedings of the Web Conference 2021*. WWW ’21. ACM, Apr. 2021. DOI: 10.1145/3442381.3449842. URL: <http://dx.doi.org/10.1145/3442381.3449842>.

Appendix A

Implementation Details

We implement our backbone algorithms in PyTorch [52] and use PyTorch Geometric [20] for GNN implementation. We based our implementation off of code released by Li *et al.* [36]. We detail the hyperparameter settings in more detail in table A.1 for Auto Arborist and table A.2 for xView (we borrow the table format from Li *et al.* [37]). Our backbone networks use PyTorch ImageNet pretraining with a linear head to predict the correct number of classes. This classification head is warmed up for an epoch before full finetuning.

A.1 Noise Robust Feature Extraction

We detail our modifications to DivideMix in algorithm 1. Our primary modification involves using class conditional gaussian mixture models to split samples as we found that some classes would retain no labels leading to them never being predicted.

A.2 SHIELD-GNN

We describe the SHIELD-GNN training loop in algorithm 2. We use a learning rate of 1e-3, batch size of 5000, and weight decay of 1e-5 in the training.

Algorithm 1: Our modification of DivideMix. See Li *et al.* [36] for more detailed psuedocode.

Input: $\theta^{(1)}$ and $\theta^{(2)}$, training dataset $(\mathcal{X}, \mathcal{Y})$, clean probability threshold τ , class conditional epochs ξ

```

while  $e < \text{MaxEpoch}$  do
  if  $e < \xi$  then
    // Fit class conditional GMMs to losses of class  $c$ 
     $\mathcal{W}_c^{(1)} = \text{GMM}(\mathcal{X}_c, c, \theta^{(2)}), \forall c \in \mathcal{Y}$ 
     $\mathcal{W}_c^{(2)} = \text{GMM}(\mathcal{X}_c, c, \theta^{(1)}), \forall c \in \mathcal{Y}$ 
  else
    // Fit unconditional GMMs
     $\mathcal{W}^{(1)} = \text{GMM}(\mathcal{X}, \mathcal{Y}, \theta^{(2)})$ 
     $\mathcal{W}^{(2)} = \text{GMM}(\mathcal{X}, \mathcal{Y}, \theta^{(1)})$ 
  end
  for  $k = 1, 2$  do
    if  $e < \xi$  then
      // Split samples class-conditionally
       $\mathcal{X}_e^{(k)} = \{(x_i, y_i, w_i) | w_i \geq \tau, \forall (x_i, y_i, w_i) \in (\mathcal{X}_c, c, \mathcal{W}_c^{(k)}), \forall c \in Y\}$ 
       $\mathcal{U}_e^{(k)} = \{x_i | w_i < \tau, \forall (x_i, w_i) \in (\mathcal{X}_c, \mathcal{W}_c^{(k)}), \forall c \in Y\}$ 
    else
       $\mathcal{X}_e^{(k)} = \{(x_i, y_i, w_i) | w_i \geq \tau, \forall (x_i, y_i, w_i) \in (\mathcal{X}, \mathcal{Y}, \mathcal{W}^{(k)})\}$ 
       $\mathcal{U}_e^{(k)} = \{x_i | w_i < \tau, \forall (x_i, w_i) \in (\mathcal{X}, \mathcal{W}^{(k)})\}$ 
    end
    // Continue DivideMix...
  end
end

```

Algorithm 2: SHIELD-GNN psuedocode.

Input: $\theta^{(1)}$ and $\theta^{(2)}$, training dataset with edges $(\mathcal{X}, \mathcal{Y}, \mathcal{E})$, GNN θ_G , Temperature T

```

 $\mathcal{X}_e, \hat{\mathcal{Y}} = \text{concat}(\theta^{(1)}(\mathcal{X}), \theta^{(2)}(\mathcal{X}))$ 
 $\hat{\mathcal{Y}} = \text{Softn}(\hat{\mathcal{Y}}, T)$ 
while  $e < \text{MaxEpoch}$  do
   $\ell = \text{L}_{\text{SHIELD-GNN}}(\theta_G(\mathcal{X}_e, \mathcal{E}), \mathcal{Y}, \hat{\mathcal{Y}})$ 
   $\text{SGD}(\ell, \theta_G)$ 
end

```

config	value
architecture	efficientnet-v2-s
#params	2*20.3M + 0.2M
optimizer	Adam [32]
learning rate	5e-4
weight decay	1e-5
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
batch size	128
learning rate schedule	backbone: constant GNN: Decrease on plateau
T	0.5
τ	0.5
class conditional epochs	20
warmup epochs	5

Table A.1: Auto Arborist SHIELD-GNN implementation details.

config	value
architecture	efficientnet-v2-s
#params	2*20.3M + 0.2M
optimizer	Adam
learning rate	5e-4
weight decay	1e-5
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
batch size	64
learning rate schedule	backbone: constant GNN: Decrease on plateau
T	0.5
τ	0.5
class conditional epochs	20
warmup epochs	5

Table A.2: xView SHIELD-GNN implementation details.