

Visual and Spatial Representation Learning with Applications in Ecology

Thesis by
Elijah Cole

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

The logo for the California Institute of Technology (Caltech), featuring the word "Caltech" in a bold, orange, sans-serif font.

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2023
Defended May 10, 2023

© 2023

Elijah Cole

ORCID: 0000-0001-6623-0966

All rights reserved

ACKNOWLEDGEMENTS

To my labmates, Ron Appel, Sara Beery, Krzysztof Chalupka, Laure Delisle, Rogério Guimarães, David Hall, Neehar Kondapaneni, Oisin Mac Aodha, Joseph Marino, Markus Marks, Mason McGill, Kevin Mei, Caroline Murphy, Grant Van Horn, Matteo Ronchi, Serim Ryou, Cristina Segalin, Suzanne Stathatos, Jennifer Sun, and Tony Zhang: thank you for your friendship.

To my advisor, Pietro Perona: thank you for your encouragement, guidance, and curiosity.

To my grandmother, Martha Mae Vickers, and my grandfather, Merle Edward Cole: thank you for your example and your wisdom.

To Shane Lubold: thank you for sharing the journey.

ABSTRACT

Machine learning has the potential to empower scientists, physicians, and other human experts working to solve problems of societal importance. To realize this goal, we need algorithms that can distill useful knowledge from real-world data. However, most machine learning research focuses on benchmarks that seldom reflect real-world challenges, such as learning from limited, noisy, or weak supervision. This thesis develops new benchmarks, algorithms, and problem settings that link fundamental machine learning research to impactful applications in ecology. In Part I, we provide context and motivation for our work. How and why should machine learning researchers work with domain experts on real-world problems? What is the appeal of ecology specifically? Part II focuses on visual representation learning with an emphasis on label efficiency. We discuss the strengths and limitations of self-supervised learning, the relationship between concept specificity and representation learning, and multi-label learning with minimal labeled data. Part III covers our work in the emerging field on spatial representation learning. In particular, we consider the problem of modeling the spatial distribution of plant and animal species. We review this important ecological problem from a machine learning perspective before showing how deep learning can transform the way these models are applied (using spatial models to assist image classifiers) and developed (jointly learning spatial distributions and representations). Finally, Part IV concludes and highlights opportunities for future work.

PUBLISHED CONTENT AND CONTRIBUTIONS

- [1] Elijah Cole, Grant Van Horn, Christian Lange, Alexander Shepard, Patrick Leary, Scott Loarie, Pietro Perona, and Oisín Mac Aodha. “Spatial Implicit Neural Representations for Global-Scale Species Mapping”. In: *International Conference on Machine Learning*. 2023.
E.C. participated in designing the project, developing the methods, running the experiments, and writing the manuscript.
- [2] Elijah Cole, Suzanne Stathatos, Björn Lütjens, Tarun Sharma, Justin Kay, Jason Parham, Benjamin Kellenberger, and Sara Beery. “Teaching Computer Vision for Ecology”. In: *arXiv preprint arXiv:2301.02211* (2023). DOI: 10.48550/arXiv.2301.02211.
E.C. participated in designing the project and writing the manuscript.
- [3] Elijah Cole, Kimberly Wilber, Grant Van Horn, Xuan Yang, Marco Fornoni, Pietro Perona, Serge Belongie, Andrew Howard, and Oisín Mac Aodha. “On Label Granularity and Object Localization”. In: *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*. Springer. 2022, pp. 604–620. DOI: 10.48550/arXiv.2207.10225.
E.C. participated in designing the project, developing the methods, running the experiments, and writing the manuscript.
- [4] Elijah Cole, Xuan Yang, Kimberly Wilber, Oisín Mac Aodha, and Serge Belongie. “When does contrastive visual representation learning work?” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14755–14764. DOI: 10.48550/arXiv.2105.05837.
E.C. participated in designing the project, developing the methods, running the experiments, and writing the manuscript.
- [5] Sara Beery*, Elijah Cole*, Joseph Parker, Pietro Perona, and Kevin Winner. “Species distribution modeling for machine learning practitioners: A review”. In: *ACM SIGCAS conference on computing and sustainable societies*. 2021, pp. 329–348. DOI: 10.48550/arXiv.2107.10400.
E.C. participated in designing the project and writing the manuscript.
- [6] Elijah Cole, Oisín Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. “Multi-label learning from single positive labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 933–942. DOI: 10.48550/arXiv.2106.09708.
E.C. participated in designing the project, developing the methods, running the experiments, and writing the manuscript.
- [7] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisín Mac Aodha. “Benchmarking representation learning for natural

world image collections”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12884–12893. DOI: [10.48550/arXiv.2103.16483](https://doi.org/10.48550/arXiv.2103.16483).

E.C. participated in designing the project, developing the methods, running the experiments, and writing the manuscript.

- [8] Sara Beery, Elijah Cole, and Arvi Gjoka. “The iWildCam 2020 Competition Dataset”. In: *CVPR Workshop on Fine-Grained Visual Categorization (2020)*. DOI: [10.48550/arXiv.2004.10340](https://doi.org/10.48550/arXiv.2004.10340).

E.C. participated in designing the project, developing the methods, and writing the manuscript.

- [9] Elijah Cole, Benjamin Deneu, Titouan Lorieul, Maximilien Servajean, Christophe Botella, Dan Morris, Nebojsa Jojic, Pierre Bonnet, and Alexis Joly. “The geolifeclf 2020 dataset”. In: *arXiv preprint arXiv:2004.04192 (2020)*. DOI: [10.48550/arXiv.2004.04192](https://doi.org/10.48550/arXiv.2004.04192).

E.C. participated in designing the project, developing the methods, and writing the manuscript.

- [10] Oisín Mac Aodha, Elijah Cole, and Pietro Perona. “Presence-only geographical priors for fine-grained image classification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9596–9606. DOI: [10.48550/arXiv.1906.05272](https://doi.org/10.48550/arXiv.1906.05272).

E.C. participated in designing the project, developing the methods, running the experiments, and writing the manuscript.

TABLE OF CONTENTS

Acknowledgements	iii
Abstract	iv
Published Content and Contributions	v
Table of Contents	vi
I Introduction	1
Chapter I: Introduction	2
1.1 Background	2
1.2 Visipedia in 2023	3
1.3 Themes	5
1.4 Thesis Organization	7
II Visual Representation Learning	12
Chapter II: Benchmarking Representation Learning for Natural World Image Collections	13
2.1 Abstract	13
2.2 Introduction	14
2.3 Related Work	16
2.4 The iNaturalist 2021 Dataset	18
2.5 NeWT: Natural World Tasks	20
2.6 Experiments	22
2.7 Conclusion	29
2.8 Acknowledgements	29
Chapter III: When Does Contrastive Visual Representation Learning Work?	34
3.1 Abstract	34
3.2 Introduction	34
3.3 Related Work	37
3.4 Methods	39
3.5 Experiments	40
3.6 Conclusion	50
3.7 Acknowledgements	51
Chapter IV: On Label Granularity and Object Localization	55
4.1 Abstract	55
4.2 Introduction	55
4.3 Related Work	57
4.4 Background	59
4.5 The iNatLoc500 Dataset	61

4.6 Experiments	64
4.7 Discussion	69
4.8 Conclusion	71
4.9 Acknowledgements	71
Chapter V: Multi-Label Learning from Single Positive Labels	76
5.1 Abstract	76
5.2 Introduction	76
5.3 Related Work	78
5.4 Problem Statement	80
5.5 Multi-Label Learning	81
5.6 Learning From Only Positive Labels	83
5.7 Experiments	87
5.8 Limitations	91
5.9 Conclusion	92
5.10 Acknowledgements	92

III Spatial Representation Learning 97

Chapter VI: Species Distribution Modeling for Machine Learning Practitioners: A Review	98
6.1 Abstract	98
6.2 Introduction	98
6.3 Representing the distribution of species	100
6.4 Species Distribution Models	102
6.5 Other types of ecological models	115
6.6 Common challenges and risks	119
6.7 What data is available and accessible?	122
6.8 Open Problems	124
6.9 Conclusion	125
6.10 Acknowledgements	125
Chapter VII: Presence-Only Geographical Priors for Fine-Grained Image Classification	142
7.1 Abstract	142
7.2 Introduction	142
7.3 Related Work	144
7.4 Methods	146
7.5 Experiments	151
7.6 Conclusion	158
7.7 Acknowledgements	158
Chapter VIII: Spatial Implicit Neural Representations for Global-Scale Species Mapping	163
8.1 Abstract	163
8.2 Introduction	163
8.3 Related Work	165
8.4 Methods	167

8.5 Experiments	170
8.6 Conclusion	178
8.7 Acknowledgements	178
IV Conclusion	183
Chapter IX: Conclusion	184
9.1 Lessons Learned	184
9.2 Future Work	185

Part I

Introduction

Chapter 1

INTRODUCTION

1.1 Background

This thesis is rooted in the Visipedia project¹, a long-running machine learning research program led by Pietro Perona and Serge Belongie. Visipedia was initially conceived as a "visual interface for Wikipedia that is able to answer visual queries and enables experts to contribute and organize visual knowledge" [28]. When Visipedia began around 2009, image data was considered to be "digital dark matter" in the sense that it clearly existed and took up space, but it was effectively inaccessible because its semantic content could not be searched or analyzed in a scalable way. Computer vision algorithms were being actively developed, but they did not yet work well enough to make Visipedia a reality.

Computer vision algorithms of the time could not make image data simple enough to analyze without throwing away important information. While humans can understand images effortlessly, an image is nothing more than a grid of numbers. Even a small color image with 256 pixels per side is *high-dimensional* because it consists of nearly 200,000 different values. As a general rule, high dimensional objects are difficult for algorithms to work with. To get around this, researchers in the late 1990s designed procedures to extract *features* that capture some of the information in the image using a (relatively) *low-dimensional* list of numbers (on the order of 1000). These features were often hand-designed summaries of patterns of edges and colors. While they were significantly simpler than the original images, these features turned out to be too simple to solve important vision tasks like object recognition. Fundamentally, the quality of the features determines the difficulty of the task, and the features were not good enough to make the tasks tractable.

The situation began to change around 2012 with the success of deep convolutional neural networks [20] trained on large datasets like ImageNet [9]. This launched the era of *representation learning*: instead of using hand-designed methods to turn images into useful features, we can use large labeled datasets to learn the transformations (implemented as deep neural networks) that facilitate a given downstream task. Suddenly, deep neural networks were able to solve some visual tasks at levels

¹<https://visipedia.org/>

close to human performance. Computer vision systems have continued to improve with refined convolutional neural networks [14, 23] and the emergence of the vision transformer architecture [10], combined with the use of larger datasets and more computation. Today, we have computer vision algorithms that are close to solving some of the main technical obstacles described in [28] — see e.g. recent work on building powerful general-purpose models for recognition, segmentation, retrieval, and detection in images [19, 27].

The "visual interface for Wikipedia" was never implemented as such, but the field of computer vision has made significant strides towards Visipedia-style systems. Today, images are searchable — this technology can even be found in commercial products like Google Lens [13]. However, the most exciting aspect of Visipedia is "the process of information discovery and the dynamic interaction of people and machines" [3] which is the foundation for self-improving knowledge systems. This aspect has enabled some of Visipedia's highest profile successes, including human-in-the-loop machine-learning-powered community science apps like iNaturalist [16], Seek [17], and Merlin [31]. For a more detailed trajectory of the Visipedia project, see the previous theses of Visipedia-focused students of Perona and Belongie [37, 4, 36, 34, 38, 8, 33, 1].

One question that is often asked about Visipedia is why the project focuses on ecology.² In principle, the Visipedia concept would apply equally well to medicine, art history, or any other field of expertise. The decision to launch Visipedia in a single focused area was driven by practicality [28, 3]. Ecology was chosen due to the convergence of interesting technical questions, excellent domain expert partners with a pressing need for automation, and problems of societal importance. This thesis follows in the same tradition, and consists of projects in which scientific questions about the natural world meet fundamental questions about machine learning.

1.2 Visipedia in 2023

While Visipedia is always evolving, I would offer the following definition for 2023:

Visipedia is a research program focused on understanding how to use data and machine learning to empower communities of experts by distilling, sharing, and exploring expert knowledge.

²I am using the term "ecology" as a shorthand for the subset of ecology related to conservation and biodiversity monitoring.

This definition intentionally omits explicit mention of images. In recent years, Visipedia has branched out to many domains beyond static images, including audio [31, 32], video [32, 18], text [12, 5], and (and in this thesis) spatial data [24, 2, 6].

In the era of large language and vision models, the face of machine learning is changing rapidly and the societal consequences are uncertain. In this context, I believe the Visipedia ethos — in which machine learning research is a means to empower domain experts and benefit society — takes on renewed importance. Even before large language and vision models, research on machine learning (and particularly deep learning) incurred significant economic, environmental, and social costs [7]. However, the incentives in machine learning research do not always lead to real progress on important problems [25, 26, 11, 22, 35, 15]. In my opinion, one of the best ways to deliver benefits to society is to do Visipedia-style machine learning research that empowers scientists, physicians, and other human experts working to solve problems of societal importance. I like this approach to machine learning research for several reasons:

- **Both parties often benefit in a collaboration between machine learning scientists and domain experts.** The domain expert gets a sense for the strengths and limitations of machine learning, and (hopefully) new tools that help them do their work more effectively. The machine learning scientist gets an (often humbling) education in the difference between artificial research problems and real-world problems, which may highlight limitations in existing algorithms and inspire further research.
- **Working on a real problem makes it easier to understand what is useful.** In a machine learning paper, spending tens of thousands of dollars on computation to achieve a 1% performance improvement is often hailed as a victory. Depending on who is footing the bill, your domain expert collaborators might be less impressed.
- **Real problems encourage system-level thinking.** It is easy for machine learning researchers to develop a sort of tunnel vision, where we become fixated on coming up with an innovative change to the architecture or loss function. However, in a real problem the benefits of making these tweaks may be dwarfed by the benefits of using existing data in a clever way or rethinking the interactions between the system and its users.

These elements will recur throughout this thesis, which focuses on machine learning research inspired by pressing problems in ecology. However, I believe that the benefits of this approach to machine learning research are general across domains. Whenever possible, I would encourage machine learning researchers to partner with domain experts and dive into important application areas — new machine learning questions are seldom far behind.

1.3 Themes

This section highlights common themes and perspectives in this thesis.

Specialist and Generalist Tasks

Visual tasks can be divided into two categories: *generalist tasks* and *specialist tasks*. Generalist tasks (e.g. classifying [29] or detecting [21] everyday objects) are those that could be solved by almost anybody in the general population. Labels for generalist tasks are relatively cheap and abundant. In contrast, specialist tasks are those that require rare knowledge such as diagnosing a disease or identifying a the species of a plant or animal (which is studied in Chapters 2, 3, and 7). The machine learning community tends to focus on generalist tasks, but many of the most impactful applications of machine learning are specialist tasks. The short supply of expert knowledge combined with the high potential for impact invites collaboration between domain experts and machine learning researchers to develop algorithms that are accurate, reliable, and data efficient.

Benchmarks and Progress

What does it mean to say we are making progress on a machine learning problem? The machine learning community has a remarkably consistent answer to this question: benchmarks. A *benchmark* consists of a *dataset* and a *protocol* for training and evaluating an algorithm using that dataset. For instance, one of the most famous benchmark datasets in machine learning is the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) benchmark, often simply referred to as ImageNet [9, 29]. The dataset consists of 1.2M images drawn from 1000 categories of everyday objects. The protocol specifies which subset of the data may be used for algorithm development, which subset of the data may be used for performance evaluation, and how performance metrics should be computed. Benchmarks have two important functions:

1. A benchmarks precisely defines a technical problem. This allows the machine

learning community to study problems (e.g. ImageNet classification) in a decentralized but consistent way.

2. A benchmark facilitates fair comparisons between algorithms. Given any two algorithms, a benchmark is supposed to provide a quantitative answer to the question: "Which algorithm is better?"

Despite their utility, benchmarks can also be problematic for at least two reasons:

1. Focusing on popular benchmarks can limit progress on other problems. A popular benchmark like ImageNet may continue to be a focal point for research long after the marginal performance improvements become quite small. This is a questionable use of resources when there are important real-world problems which are getting relatively little attention from the machine learning community.
2. Interesting technical questions may go unstudied if our benchmarks are not diverse enough. The kinds of machine learning questions that crop up depend on the properties of the data one is studying. For instance, most of the categories in ImageNet are *coarse-grained*, meaning that they are easy to distinguish from each other (e.g. *castle*, *coffee mug*, *cowboy boot*). If an algorithm works well on ImageNet classification, will it also work well in contexts like species classification where the categories are *fine-grained*? How would we know unless we have fine-grained benchmarks?

This thesis takes a step towards mitigating these problems by introducing new benchmarks that reflect important scientific problems and highlighting how these benchmarks lead to new machine learning insights. Chapters 2, 4, 5, 7, and 8 each introduce new benchmarks and articulate new machine learning research questions. I believe linking machine learning benchmarks to scientific problems is a pathway to richer, more diverse, and more impactful machine learning research.

Expertise and Algorithms

How should one think about the role of domain expertise in machine learning systems? In an influential 2019 blog post [30], Richard Sutton described an idea he calls *The Bitter Lesson*:

Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation.

The basic claim is that general purpose learning algorithms trained with more data and computational resources will eventually outperform algorithms that have domain knowledge "baked in" by their designers. There are plenty of cases where this has turned out to be true. The deep learning revolution in computer vision is a notable example. Even there, convolutional neural networks include some prior knowledge (e.g. translation invariance), but now we are seeing transformers — with fewer built-in assumptions, with more data and computation — beginning to pull ahead. If we want to put machine learning to work for domain experts, where does this leave us? First, I claim that "leveraging computation" is not always straightforward. Setting up the right learning problem (what data to use, how to use it, how to measure success) often requires domain expertise. For instance, some species are virtually indistinguishable by eye, so no scaling up of data or computation will solve the problem. Chapter 7, shows how this problem can be solved by incorporating spatial and temporal information in addition to image data. Second, there are many domains where large labeled datasets do not exist and are not on track to be collected any time soon. We still need solutions that work well in such settings, and building in some domain knowledge to reduce the need for labeled data can be quite valuable. For instance, Chapter 5 shows that it is possible to train multi-label image classifiers using drastically reduced label budgets by incorporating simple priors into the learning process. The Bitter Lesson should not discourage us from thinking hard about domain structure if it helps us to solve important real-world problems.

1.4 Thesis Organization

The rest of this thesis is divided into three parts: visual representation learning (Part II), spatial representation learning (Part III), and conclusions (Part IV).

Part II concerns visual representation learning, the problem of learning embeddings of images that are useful for solving visual tasks. Chapter 2 introduces a new benchmark for visual self-supervised learning grounded in ecologically meaningful tasks. Chapter 3 builds on this work to provide new fundamental insights into visual self-supervised learning by carefully analyzing the properties of these algorithms relative to the needs of domain expert users. Chapter 4 introduces a new benchmark

for weakly supervised object localization and demonstrates how to use label hierarchy information to improve the performance and data efficiency of many different object localization algorithms. Chapter 5 poses the new problem of single positive multi-label learning and contributes the first benchmarks and algorithms to solve this problem.

In Part III covers spatial representation learning, the problem of learning embeddings of locations in time and space that facilitate geospatial tasks. Chapter 6 provides a review of species distribution modeling intended for machine learning researchers. Chapter 7 demonstrates how species distribution models can be used to improve image classifiers. Chapter 8 provides the first large-scale benchmarks for species distribution modeling and demonstrates the potential of implicit neural representations for this task.

Part IV discusses conclusions and future work.

Funding Acknowledgements

The work in this thesis was supported by an NSF Graduate Research Fellowship (Grant No. DGE1745301), the Caltech Resnick Sustainability Institute, the U.S. Fish and Wildlife Service (Grant No. F22AP01490-00), and the Nissan Corporation.

References

- [1] Sara Meghan Beery. “Where the Wild Things Are: Computer Vision for Global-Scale Biodiversity Monitoring”. PhD thesis. California Institute of Technology, 2023.
- [2] Sara Beery*, Elijah Cole*, Joseph Parker, Pietro Perona, and Kevin Winner. “Species distribution modeling for machine learning practitioners: A review”. In: *ACM SIGCAS conference on computing and sustainable societies*. 2021, pp. 329–348. DOI: [10.48550/arXiv.2107.10400](https://doi.org/10.48550/arXiv.2107.10400).
- [3] Serge Belongie and Pietro Perona. “Visipedia circa 2015”. In: *Pattern Recognition Letters* 72 (2016), pp. 15–24.
- [4] Steven Branson. “Interactive learning and prediction algorithms for computer vision applications”. PhD thesis. University of California, San Diego, 2012.
- [5] Peter Ebert Christensen et al. “Searching for Structure in Unfalsifiable Claims”. In: *arXiv preprint arXiv:2209.00495* (2022).
- [6] Elijah Cole, Grant Van Horn, Christian Lange, Alexander Shepard, Patrick Leary, Scott Loarie, Pietro Perona, and Oisín Mac Aodha. “Spatial Implicit

- Neural Representations for Global-Scale Species Mapping”. In: *International Conference on Machine Learning*. 2023.
- [7] Kate Crawford. *The atlas of AI: Power, politics, and the planetary costs of artificial intelligence*. Yale University Press, 2021.
- [8] Yin Cui. *Learning from Fine-grained and Long-tailed Visual Data*. Cornell University, 2019.
- [9] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *CVPR*. 2009.
- [10] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *ICLR*. 2021.
- [11] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. “Are we really making much progress? A worrying analysis of recent neural recommendation approaches”. In: *Proceedings of the 13th ACM conference on recommender systems*. 2019, pp. 101–109.
- [12] Maxwell Forbes et al. “Neural naturalist: generating fine-grained image comparisons”. In: *arXiv preprint arXiv:1909.04101* (2019).
- [13] *Google Lens*. <https://lens.google/>, accessed May 1, 2023.
- [14] Kaiming He et al. “Deep residual learning for image recognition”. In: *CVPR*. 2016.
- [15] Peter Henderson et al. “Deep reinforcement learning that matters”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.
- [16] *iNaturalist*. www.inaturalist.org, accessed Mar 7 2022.
- [17] *iNaturalist*. *Seek by iNaturalist*. <https://apps.apple.com/us/app/seek-by-inaturalist/id1353224144>. 2020.
- [18] Justin Kay et al. “The Caltech Fish Counting Dataset: A Benchmark for Multiple-Object Tracking and Counting”. In: *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*. Springer. 2022, pp. 290–311.
- [19] Alexander Kirillov et al. “Segment anything”. In: *arXiv preprint arXiv:2304.02643* (2023).
- [20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *NeurIPS*. 2012.
- [21] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *ECCV*. 2014.
- [22] Zachary C Lipton and Jacob Steinhardt. “Troubling Trends in Machine Learning Scholarship: Some ML papers suffer from flaws that could mislead the public and stymie future research.” In: *Queue* 17.1 (2019), pp. 45–77.

- [23] Zhuang Liu et al. “A convnet for the 2020s”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11976–11986.
- [24] Oisín Mac Aodha, Elijah Cole, and Pietro Perona. “Presence-only geographical priors for fine-grained image classification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9596–9606. DOI: 10.48550/arXiv.1906.05272.
- [25] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. “A metric learning reality check”. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*. Springer. 2020, pp. 681–699.
- [26] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. “Unsupervised domain adaptation: A reality check”. In: *arXiv preprint arXiv:2111.15672* (2021).
- [27] Maxime Oquab et al. “DINOv2: Learning Robust Visual Features without Supervision”. In: *arXiv preprint arXiv:2304.07193* (2023).
- [28] Pietro Perona. “Vision of a visipedia”. In: *Proceedings of the IEEE* 98.8 (2010), pp. 1526–1534.
- [29] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *IJCV* (2015).
- [30] Richard Sutton. “The Bitter Lesson”. In: *Incomplete Ideas (blog)* (2019).
- [31] Cornell University. *Merlin Bird Id*. <https://apps.apple.com/us/app/merlin-bird-id-by-cornell-lab/id773457673>. 2020.
- [32] Grant Van Horn et al. “Exploring Fine-Grained Audiovisual Categorization with the SSW60 Dataset”. In: *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*. Springer. 2022, pp. 271–289.
- [33] Grant Richard Van Horn. “Towards a Visipedia: Combining Computer Vision and Communities of Experts”. PhD thesis. California Institute of Technology, 2019.
- [34] Andreas Veit. “Learning Conditional Models for Visual Perception”. PhD thesis. Cornell University, 2018.
- [35] Kiri Wagstaff. “Machine learning that matters”. In: *arXiv preprint arXiv:1206.4656* (2012).
- [36] Catherine Lih-Lian Wah. “Leveraging Human Perception and Computer Vision Algorithms for Interactive Fine-Grained Visual Categorization”. PhD thesis. University of California, San Diego, 2014.
- [37] Peter Welinder. *Hybrid Human-Machine Vision Systems: Image Annotation using Crowds, Experts and Machines*. California Institute of Technology, 2012.

- [38] Kimberly Wilber. “Learning Perceptual Similarity from Crowds and Machines”. PhD thesis. Cornell University, 2018.

Part II

Visual Representation Learning

*Chapter 2***BENCHMARKING REPRESENTATION LEARNING FOR
NATURAL WORLD IMAGE COLLECTIONS**

- [1] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. “Benchmarking representation learning for natural world image collections”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12884–12893. DOI: [10.48550/arXiv.2103.16483](https://doi.org/10.48550/arXiv.2103.16483).

2.1 Abstract

Recent progress in self-supervised learning has resulted in models that are capable of extracting rich representations from image collections without requiring any explicit label supervision. However, to date the vast majority of these approaches have restricted themselves to training on standard benchmark datasets such as ImageNet. We argue that fine-grained visual categorization problems, such as plant and animal species classification, provide an informative testbed for self-supervised learning. In order to facilitate progress in this area we present two new natural world visual classification datasets: iNat2021 and NeWT. The former consists of 2.7M images from 10k different species uploaded by users of the citizen science application iNaturalist. We designed the latter, NeWT, in collaboration with domain experts with the aim of benchmarking the performance of representation learning algorithms on a suite of challenging natural world binary classification tasks that go beyond standard species classification. These two new datasets allow us to explore questions related to large-scale representation and transfer learning in the context of fine-grained categories. We provide a comprehensive analysis of feature extractors trained with and without supervision on ImageNet and iNat2021, shedding light on the strengths and weaknesses of different learned features across a diverse set of tasks. We find that features produced by standard supervised methods still outperform those produced by self-supervised approaches such as SimCLR. However, improved self-supervised learning methods are constantly being released and the iNat2021 and NeWT datasets are a valuable resource for tracking their progress.

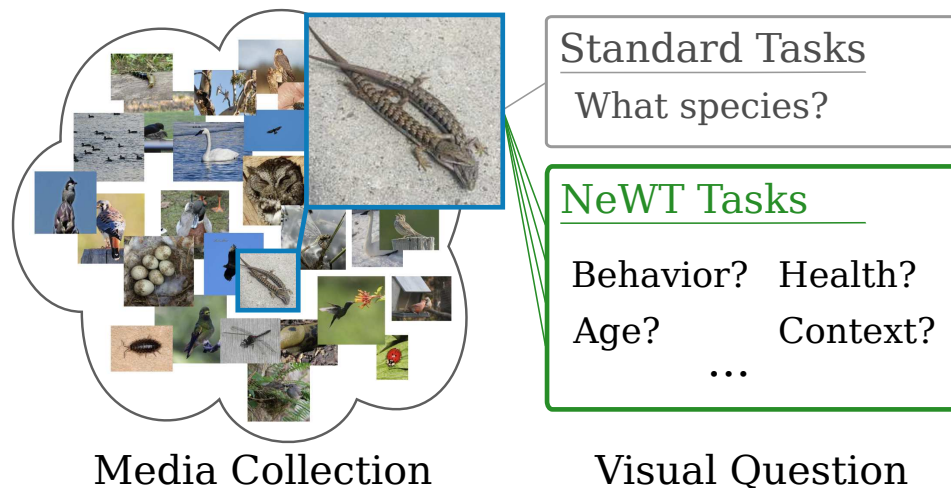


Figure 2.1: Existing fine-grained image datasets are typically focused on a single task e.g. species identification. As natural world media collections grow, we have the opportunity to extract information beyond species labels to answer important ecological questions. For example, with the help of community scientists, researchers from the NHMLA were able to curate over 500 images of alligator lizards mating, a phenomenon seldomly recorded in the existing scientific literature [16]. We analyze if trained feature extractors can answer similar novel image understanding questions with minimal additional training and present **NeWT**, a diverse benchmark of natural world visual understanding tasks such as animal health, life-stage, and behavior, among others.

2.2 Introduction

Learning representations of images through self-supervision alone has seen impressive advancement over the last few years. There are tantalizing results that show self-supervised methods, fine-tuned with 1% of the training labels, reaching the performance of their fully supervised counterparts [7]. In many domains, aggregating large amounts of data is typically not the bottleneck. Rather, it is the subsequent labeling of that data that consumes vast amounts of money and time. This is further compounded in fine-grained domains, e.g. medicine or the natural world, where sufficiently well trained annotators are few or their time is expensive. If the benefits of self-supervised learning come to full fruition, then the applicability and impact of computer vision models across many domains will see a rapid increase.

One particular domain that is well suited for this type of advancement is the study of the natural world through photographs collected by communities of enthusiasts. Websites such as iNaturalist [28] and eBird [54] amass large collections of media annually. To date, there are 60M images in iNaturalist spanning the tree of life and 25M images of birds from around the world in eBird, both representing point-in-time

records of wildlife. Identifying the species in an image has been well studied by the computer vision community [60, 32, 1, 58], however this is only the tip of the iceberg in terms of questions one may wish to answer using these vast collections. These datasets contain evidence of the health and state of the individuals depicted, along with their behavior. Having an automated system mine this data for these types of properties could help scientists fill in missing pieces of basic natural history information that are crucial for our understanding of global biodiversity and help measure the loss of biodiversity due to human impact [4].

To give one example, science is ignorant to the nesting requirements of thousands of bird species, including the vulnerable Pink-throated Brilliant (*Heliodoxa gularis*) [67]. Knowing how and where this species builds its nest is a crucial piece of information needed when discussing conservation based interventions, particularly as it pertains to the ability of this species to exist in degraded and fragmented habitats [67]. While nothing can replace the capabilities of a biologist in the field, citizen science projects like eBird and iNaturalist are collecting raw images that could help answer some of these questions. However, herein lies the problem. It is currently a daunting task to label training datasets for these specialized questions that would satisfy the data appetite of an off-the-shelf deep network.

Self-supervised learning is one potential solution that could alleviate the labeling burden by taking advantage of large media collections. While most research on self-supervised learning focuses on ImageNet [52], in this work we expand these techniques to the natural world domain and fine-grained classification. Following Goyal et al. [18], we maintain that a good representation should generalize to many different tasks, with limited supervision or fine-tuning. We do not investigate self-supervised learning as an initialization scheme for a model that is further optimized and finetuned, but rather as a way to learn feature representations themselves. Importantly, [18] point out that self-supervised feature learning and subsequent feature evaluation on the same dataset does not test the generalization of the features. Inspired by this, we present a new large-scale pretraining dataset and new benchmark tasks specifically designed to enable us to ask questions about the generalization of self-supervised learning on natural world image collections.

We make the following three contributions:

- **iNat2021** - A new large-scale image dataset collected and annotated by community scientists that contains over 2.7M images from 10k different species.

- **NeWT** - A new suite of 164 challenging natural world visual benchmark tasks that are motivated by real world image understanding use cases.
- A detailed evaluation of self-supervised learning in the context of natural world image collections. We show that despite recent progress, self-supervised features still lag behind supervised variants.

2.3 Related Work

Learning Visual Representations

Transfer learning using features extracted from deep networks that have been trained via supervision on large datasets results in powerful features that can be applied to many downstream tasks [12, 63]. However, there is evidence to suggest that pretraining on datasets such as ImageNet [52] is less effective on fine-grained categories when the labels are not well represented in the source dataset [34]. Self-supervised learning, i.e. learning visual representations without requiring explicit label supervision, is an exciting research area that, if successful, could provide a much more scalable way to learn representations for a wide variety of tasks, including fine-grained ones.

Earlier work in self-supervised learning in vision involved framing the learning problem via proxy tasks, e.g. predicting context from image patches [11, 49], image colorization [65], or predicting image rotation [17], to name a few. The most effective recent approaches have focused on contrastive learning based training objectives [23, 22], where the aim is to learn features from images such that augmented versions of the same image are nearby in the feature space, and other images are further away. This can require a large batch size during training to ensure that there are a sufficient number of useful negatives [6], which necessitates large compute resources during training. Recent advances include memory banks to address the need for large batches [61, 25, 8], additional embedding layers [7], and more advanced augmentations [5], among others.

In our experiments, we compare the performance of several leading self-supervised learning algorithms [6, 8, 5, 7] to conventional supervised learning in the context of fine-grained pretraining to try to understand what gap, if any, exists between the features learned by these very different paradigms on natural world image classification tasks.

Benchmarking Representation Learning

Like Cui et al. [10], we are also interested in understanding how well models trained on large-scale natural world datasets can transfer to downstream fine-grained tasks. However, [10] only explored transfer learning using fully supervised, as opposed to self-supervised, training. [53] combined self-supervised and meta learning and showed improved few-shot classification accuracy for fine-grained categories. Instead of jointly training our models, we decouple feature learning from classification so that we can better understand generalization performance.

Our work can be seen as a continuation of recent attempts to benchmark the performance of self-supervised learning e.g. [18, 33, 64]. We swap out their pretext tasks for more recent approaches and utilize natural world evaluation datasets containing a mix of fine and coarse-grained visual concepts to test the generalization of the learned features. This is in contrast to standard computer vision datasets or synthetic tasks [46] that are commonly used for evaluation.

The majority of existing self-supervised methods train on ImageNet [52]. There are some exceptions, such as [18] and [19], that also train on alternative datasets such as YFCC100M [55] and Places205 [66], respectively. We present results obtained by learning representations obtained through self-supervision alone on a large-scale natural world dataset — as opposed to just linear evaluation [45, 5, 13] or finetuning in this domain [25].

Fine-Grained Datasets

The vision community is not lacking in image datasets. The set of existing datasets include those that are large-scale and span broad category groups, e.g. [52, 35], through to smaller, but densely annotated, ones, e.g. [14, 40, 38, 21]. In addition, there are a number of domain specific (i.e. "fine-grained") datasets covering object categories such as airplanes [44, 59], birds [60, 1, 57, 37], dogs [32, 50, 42], fashion [31], flowers [47, 48], food [2, 26], leaves [39], vehicles [36, 41, 62, 15], and, of course, human faces [27, 51, 20, 3]. Most closely related to our work are the existing iNaturalist species classification datasets [58, 30], which contain a set of coarse and fine-grained species classification problems.

Distinct from these existing datasets, our new NeWT dataset presents a rich set of evaluation tasks that are not solely focused on one type of visual challenge e.g. species classification. Instead, NeWT contains a wide variety of tasks encompassing behavior, health, and context, among others. Most importantly, our tasks are

dataset	# classes	# train	# val	# test	min # ims	max # ims	avg # ims
iNat2017 [58]	5,089	579,184	95,986	182,707	9	3919	114
iNat2018 [30]	8,142	437,513	24,426	149,394	2	1,000	54
iNat2019 [30]	1,010	265,213	3,030	35,350	16	500	263
iNat2021 mini	10,000	500,000	*100,000	*500,000	50	50	50
iNat2021	10,000	2,686,843	*100,000	*500,000	152	300	267

Table 2.1: Comparison of iNat2021 dataset to previous iterations. iNat2021 is more than five times larger than existing large-scale species classification datasets, making it a valuable tool for benchmarking representation learning. Min, max, and avg refer to the number of images per class in the respective training sets. *Both variants of iNat2021 use the same validation and test sets.

informed by natural world domain experts and are thus grounded in real-world use cases. Paired with our new iNat2021 dataset, which contains five times more training images and nearly 20% more categories than the largest previous version [58], they serve as a valuable tool to enable us to better understand and evaluate progress in both transfer and self-supervised learning in challenging visual domains.

2.4 The iNaturalist 2021 Dataset

Dataset Overview

While several large-scale natural world datasets already exist, the current largest one, iNat2017 [58], only contains half the number of training images as ImageNet [52]. To better facilitate research in representation learning for this domain, we introduce a new image dataset called iNat2021. iNat2021 consists of 2.7M training images, 100k validation images, and 500k test images, and represents images from 10k species spanning the entire tree of life. In addition to its overall scale, the main distinguishing feature of iNat2021 is that it contains at least 152 images in the training set for each species. We provide a comparison to existing datasets in Table 2.1 and a breakdown of the image distribution in Table 2.3. Unlike previous iterations, we have split the training and testing images in iNat2021 by a specific date and have allowed a particular photographer to have images in both the train and test splits. There is an intuitive interpretation to this decision: we are retroactively building a computer vision training dataset, composed of data that was submitted *over a year ago*, to classify the most observed species in the *last year*, which is our test set. While there are many ways we could have decided the train and test split criteria, we believe this is particularly natural and lends itself well to future updates (the date split simply increases by a year). A detailed description of the steps we took to create the dataset are outlined in the supplementary material.

train split	top-1	top-2	top-3	top-4	top-5
iNat2021 mini	0.654	0.759	0.806	0.833	0.851
iNat2021	0.760	0.848	0.882	0.901	0.914
iNat2021 mini *	0.616	0.722	0.769	0.798	0.818
iNat2021 *	0.746	0.836	0.872	0.891	0.904

Table 2.2: Top-K Accuracy on the iNat2021 test set. Models marked with a * have been initialized with random weights, otherwise ImageNet initialization is used.

In addition to the full sized dataset, we have also created a smaller version (iNat2021 mini) that contains 50 training images per species, sampled from the full train split. These two different training splits allows researchers to explore the benefits of training algorithms on five times more data. The mini dataset also keeps the training set size reasonable for desktop-scale experiments. In addition to the images themselves, we also include latitude, longitude, and time data for each, facilitating research that incorporates additional meta data to improve fine-grained classification accuracy, e.g. [43, 9].

Comparisons to iNat2017-2019

In Table 2.1 we compare the new iNat2021 dataset with previous datasets built from iNaturalist. iNat2017 was the first large-scale species classification dataset [58]. iNat2018 addressed the long tail problem inherent in large-scale media repositories. iNat2019 attempted to focus specifically on genera with large number of species (at least 10), resulting in a smaller dataset consisting of many 10-way fine-grained classification problems. Our iNat2021 dataset is similar to iNat2017 and iNat2018 in terms of its large-scale scope, however we incorporate the iNat2019 style focus on fine-grained challenges with our introduction of the NeWT collection of evaluation datasets, see Section 2.5. While we have effectively removed the long tail training distribution that was the focus of other iNat datasets, we have included sufficient images per species where this phenomena can still be studied by systematically removing data. More data per species has the effect of decreasing the difficulty of iNat2021 in the purely supervised setting, but we believe that the additional images for each category are essential to enable us to systematically evaluate the effectiveness of self-supervised learning for natural world visual categories.

Baseline Supervised Experiments

We train ResNet50 [24] networks, both with and without ImageNet initialization, to benchmark the performance of iNat2021. Table 2.2 shows the top-k accuracy












	Iconic Group	Species Count	Train Images	Full ACC	Mini ACC
	Insects	2,526	663,682	0.813	0.715
	Fungi	341	90,048	0.786	0.707
	Plants	4,271	1,148,702	0.800	0.692
	Mollusks	169	44,670	0.756	0.670
	Animalia	142	37,042	0.747	0.654
	Fish	183	45,166	0.725	0.640
	Arachnids	153	40,687	0.704	0.582
	Birds	1,486	414,847	0.662	0.537
	Mammals	246	68,917	0.590	0.496
	Reptiles	313	86,830	0.554	0.430
	Amphibians	170	46,252	0.526	0.417

Table 2.3: Number of species, training images, and mean test accuracy in iNat2021 for each iconic group. ‘Animalia’ is a catch-all category that contains species that do not fit in the other iconic groups. For the mini train split, each species has 50 train images.

achieved when training using the full and mini datasets, and Table 2.3 shows the top-1 accuracy broken down by iconic groups. The model trained on the mini dataset results in a top-1 accuracy of 65.4%, while the full model achieves 76.0%, showing that an increase from 500k training images to 2.7M results in an ~ 11 percentage point increase in accuracy. The corresponding top-1 results for the validation set are 65.8% and 76.4%. On average, insects are the best performing iconic group, and amphibians are the worst performing group. While these average statistics are interesting, we do not believe they demonstrate that insects are necessarily "easier" to identify than amphibians. We are most likely seeing a bias in the iNat2021 dataset. Perhaps, on average, it is easier to take a close-up photograph of an insect than it is to photograph an amphibian. Or perhaps the amphibian species have more visual modalities than insects. Finally, we observe that models trained from randomly initialized weights perform slightly worse than those trained from ImageNet initialization, but the gap closes when training on the full dataset.

2.5 NeWT: Natural World Tasks

Large media repositories, such as Flickr, the Macaulay Library, and iNaturalist, have been utilized to create species classification datasets such as CUB [60], Bird-Snap [1], NABirds [57], and the collection of iNaturalist competition datasets [58]. These datasets have become standard experimental resources for computer vision researchers and have been used to benchmark the progress of classification mod-

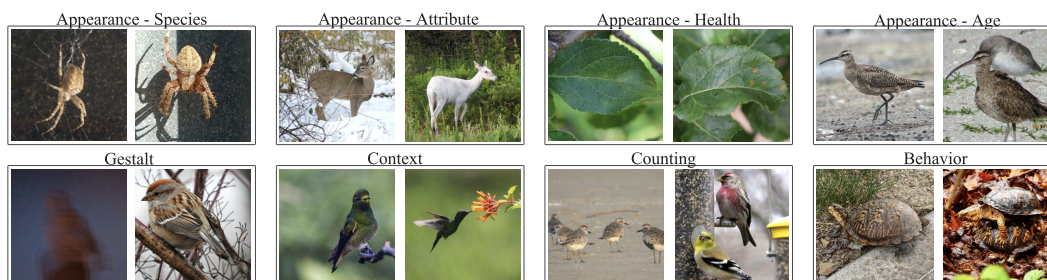


Figure 2.2: Example image pairs from a binary classification task within each coarse task grouping of the NeWT dataset.

els over the last decade. Improvements on these datasets have in turn led to the incorporation of these models into useful applications that assist everyday users in recognizing the wildlife around them, e.g. [39, 29, 56]. However, there are far more questions that biologists and practitioners would like to ask of these large media repositories in addition to "What species is in this photo?" For example, an ornithologist may like to ask, "Does this photo contain a nest?" or "Does this photo show an adult feeding a nestling?" Similarly, a herpetologist may like to ask, "Does this photo show mating behavior for the Southern Alligator Lizard?" Researchers can certainly answer these questions themselves for a few images. The problem is the scale of these archives, and the fact that they are continually growing. Can a computer vision model be used to answer these questions? While we do not have large collections of datasets labeled with nests or eggs or mating behavior, we do have large-scale species classification datasets. This raises the question about the adaptability of a model trained for species classification to these new types of questions. Similarly, with the recent advances in self-supervised learning there is the potential for a self-supervised model to be readily adapted to answer these varied tasks. To help address these questions we have constructed a collection of Natural World Tasks (NeWT) that can be used to benchmark current representation learning methods.

NeWT is composed of 164 highly curated binary classification tasks sourced from iNaturalist, the Macaulay Library, and the NABirds dataset, among others. No images from NeWT occur in the iNat2021 training dataset, and the images in tasks not sourced from iNaturalist are reasonably similar to images found on iNaturalist. This makes the iNat2021 dataset a perfect pretraining dataset for NeWT. Unlike some of the potential data quality issues found in iNat2021 (see supplementary material), each task in NeWT has been vetted for data quality with the assistance of domain experts. While species classification still plays a large role in NeWT (albeit reduced

down to difficult fine-grained pairs of species), the addition of other types of tasks makes this dataset uniquely positioned to determine how well different pretrained models can answer various natural world questions. Each task has approximately uniform positive and negative samples, as well as approximately uniform train and test samples. The size of each task is modest, on the order of 50-100 images per class per split (for a total of 200-400 images per task), which makes them very convenient for training and evaluating linear classifiers. We have coarsely categorized the tasks into eight groups (see Figure 2.2 for visual examples) with the total number of binary tasks per group in parentheses:

- **Appearance - Age (14)** Tasks where the age of the species is the decision criteria, e.g. "Is this a hatch-year Whimbrel?"
- **Appearance - Attribute (7):** Tasks where a specific attribute of an organism is used to make the decision, e.g. "Is the deer leucistic?"
- **Appearance - Health (9):** Tasks where the health of the organism is the decision criteria, e.g. "Is the plant diseased?"
- **Appearance - Species (102):** Tasks where the goal is to distinguish two visually similar species. This can include species from iNat2021, but with new, unseen training data, and tasks from species not included in iNat2021.
- **Behavior (16)** Tasks where the evidence of a behavior is the decision criteria, e.g. "Are the lizards mating?"
- **Context (8)** Tasks where the immediate or surrounding context of the organism is the decision criteria, e.g. "Is the hummingbird feeding at a flower?"
- **Counting (2)** Tasks where the number of specific instances is the decision criteria, e.g. "Are there multiple bird species present?"
- **Gestalt (6)** Tasks where the quality, composition, or type of photo is the decision criteria, e.g. "Is this a high quality or low quality photograph of a bird?"

2.6 Experiments

Here we present an analysis of different learned image representations trained on multiple datasets and evaluate their effectiveness on existing fine-grained datasets and NeWT.

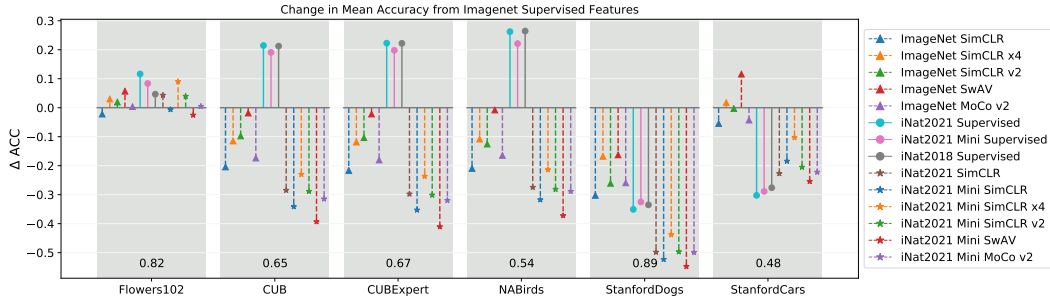


Figure 2.3: **Fine-grained evaluation.** The mean top-1 accuracy difference between "off-the-shelf" supervised ImageNet features and other pretraining strategies on existing fine-grained datasets. For context, the accuracy of the ImageNet features are printed above the dataset labels along the x-axis. All methods utilize a ResNet50 backbone architecture, and all experiments use features extracted by the last convolution block (dim=2048) to train a linear SVM using SGD (x4 models have dim=8192). Techniques that make use of supervised pretraining have a solid stem line, while techniques that use self-supervision for pretraining have a dashed stem line. Techniques that utilize ImageNet have a triangle marker, techniques that utilize an iNat dataset with supervision have a circle marker, and techniques that utilize an iNat dataset with a self-supervision training objective have a star marker. Several patterns are apparent: (1) Self-supervised methods rarely do better than "off-the-shelf" supervised ImageNet features. (2) Pretraining on iNat datasets with supervision leads to better results on downstream tasks that contain categories similar to those found in iNat datasets (i.e. flowers and birds), but this does not hold for self-supervised objectives. (3) Self-supervised models trained on ImageNet do better than their iNat counterparts. For detailed accuracy numbers see the supplementary material.

Implementation Details

Given a specific configuration of {feature extractor, pretraining dataset, training objective}, our feature representation evaluation protocol is the same for all experiments. Every experiment uses the ResNet50 [24] model as the feature extractor, with some experiments modifying the width multiplier parameter of the network to 4. We consider ImageNet, iNat2018, iNat2021, and the iNat2021 mini dataset for the pretraining dataset. The training objective can either be a supervised classification loss (standard cross-entropy) or one of the following self-supervised objectives: SimCLR [6], SimCLR v2 [7], SwAV [5], or MoCo v2 [8].

The supervised experiments using iNat2021 mini and iNat2018 are trained for 65-90 epochs, starting from ImageNet initialization, and we used the model checkpoint that performed the best on the respective validation set. The supervised experiments using iNat2021 were trained for 20 epochs, also starting from ImageNet initialization. For self-supervised techniques pretrained on ImageNet, we make use

of model checkpoint files accompanying the official implementation of the method. For models self-supervised on iNat datasets we used default parameters from the respective techniques unless otherwise stated. Our experiments using SimCLR v2 on iNat datasets do not incorporate knowledge distillation from a larger network nor the MoCo style memory mechanism; instead we train the ResNet50 backbone using a 3-layer projection head instead of the 2-layer projection head found in the original SimCLR objective. See the supplementary material for additional details on model training.

After training the ResNet50 model on the selected dataset, it is then used as a feature extractor on "downstream" evaluation datasets. Images are resized so the smaller edge is 256 then we take a center crop of 224x224, which is then passed through the model. No other form of augmentation is used. Features are extracted from the last convolutional block of the ResNet50 model and have a dimension of 2048 unless the width of the network was modified to 4, in which case the dimension is 8192. A linear model is then trained on these features and the associated ground truth class labels. Details of the linear model are provided below. We use top-1 accuracy on the held out test set of the respective "downstream" dataset as the evaluation metric for the linear model. We compare different feature representations by measuring the relative change in accuracy when using supervised ImageNet features as the baseline (Δ ACC in Figure 2.3 and Figure 2.4). We chose supervised ImageNet features as the baseline because these features are readily accessible to nearly all practitioners, requiring zero additional training and very little computational resources. To facilitate reproducibility, all pretrained models are accessible from our GitHub project page.

Experiments on Fine-Grained Datasets

In this section we demonstrate the utility of iNat2021 as a pretraining dataset for existing fine-grained datasets. The extracted features are evaluated on Flowers102 [48], CUB [60], NABirds [57], StanfordDogs [32], and StanfordCars [36]. We also present results on CUBExpert, which is the standard CUB dataset but the class labels have been verified and cleaned by domain experts [57]. For these experiments, the linear model is a SVM trained using SGD for a maximum of 3k epochs with a stopping criteria tolerance of $1e-5$. For every experiment, we use 3-fold cross validation to determine the appropriate regularization constant $\alpha \in [1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 0.1, 1, 10]$.

We present the relative accuracy changes in relation to supervised ImageNet features

for the various techniques in Figure 2.3. Please consult the supplementary material for specific accuracy values. Overall we find that supervised techniques produce the best features for all datasets except Stanford Cars, where the SwAV model trained on ImageNet produced the best features. The iNat2021 supervised model is the best performing on Flowers102, CUB, and CUBExpert; the iNat2018 supervised model is the best on NABirds, narrowly eclipsing the iNat2021 supervised model (0.806 vs. 0.804 top-1 accuracy); and the supervised ImageNet model is the best on StanfordDogs. When considering self-supervised methods, the SwAV model trained on ImageNet is consistently the top performer except for the Flowers102 dataset, where the SimCLR x4 model trained on iNat2021 mini achieves better performance (using a 4x larger feature vector than the SwAV model).

In terms of pretraining datasets for self-supervised techniques, the ImageNet dataset appears better than the iNat2021 dataset: note the lines for self-supervised methods trained on iNat2021 and iNat2021 mini in Figure 2.3 are uniformly below their ImageNet counterparts for all datasets except Flowers102. While not particularly surprising for the Stanford Dogs and Cars datasets that differ fundamentally from the iNaturalist domain, this is a surprising result for the bird datasets: CUB, CUBExpert, and NABirds. The ImageNet dataset has about 60 species of birds with ~60k training images, while the iNat2021 dataset has 1,486 species with 414,847 and 74,300 training images in the large and mini splits respectively. Even with increased species and training samples, the ImageNet dataset outperforms the iNat2021 dataset on downstream bird tasks. Perhaps this is an artifact of the *types* of images within these datasets as opposed to the *domain* of the datasets. The self-supervised techniques considered in this work were designed for ImageNet, therefore their default augmentation strategy appears to be designed for objects that take up a large fraction of the image size. Applying these strategies to datasets where objects do not necessarily take up large fraction of the image size (like iNat2021) appears to be inappropriate. See the supplementary material for an analysis of the sizes of bird bounding boxes across the datasets.

Note that supervised methods can still recover discriminative features from the iNat datasets (see the performance of supervised iNat2021 and iNat2021 mini in Figure 2.3), so it should be feasible for self-supervised methods to leverage these datasets to learn better representations. Interestingly, the effect of data size is not very apparent in Figure 2.3 for the experiments that use the large and mini variants of the iNat2021 dataset. While performance on the actual iNat2021 improved by 11

percentage points when switching from the mini to the large (see Table 2.2), we do not see a similar level of improvement for downstream tasks.

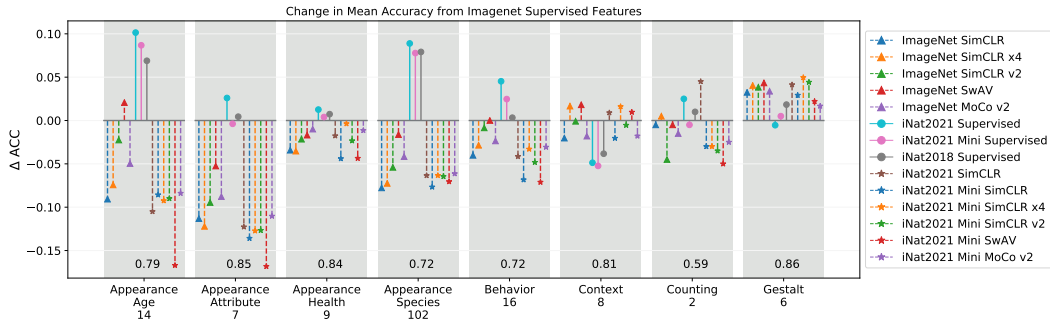


Figure 2.4: **NeWT evaluation.** The mean top-1 accuracy difference between "off-the-shelf" supervised ImageNet features and various other pretraining strategies on the NeWT dataset, divided into related groups. See Figure 2.3 for information regarding the plot organization and interpretation. Several patterns are apparent: (1) Supervised learning using iNaturalist data achieves better performance on NeWT tasks that focus on species appearance and behavior. (2) Self-supervised learning achieves better performance compared to supervised methods on the *Gestalt* tasks, i.e. tasks that do not focus on a particular individual. (3) For self-supervision, we do not see a consistent benefit to using iNat2021 over ImageNet (unlike Figure 2.3); sometimes pretraining on iNat2021 leads to better performance than pretraining on ImageNet, other times it is reversed. For detailed accuracy numbers see the supplementary material.

Experiments on NeWT

In this section we use the collection of binary tasks in NeWT as "downstream" classification tasks to investigate the effect of different pretraining methods. For these experiments the linear model is a SVM trained using liblinear for a maximum of 1k iterations with a stopping criteria tolerance of $1e-5$. For every experiment, we use 3-fold cross validation to determine the appropriate regularization constant $C \in [1e-4, 1e-3, 1e-2, 0.1, 1, 10, 1e2, 1e3]$.

The supervised ImageNet model achieved an average accuracy of 0.744 across all 164 NeWT tasks. The supervised iNat2021 model achieved the best average accuracy with a score of 0.806, followed by the supervised iNat2021 mini model at 0.793 and then the supervised iNat2018 model at 0.791. For self-supervised models, the SwAV model trained on ImageNet did the best at 0.733 average accuracy. We show the relative accuracy changes in relation to supervised ImageNet features for the various techniques in Figure 2.4. See the supplementary material for specific accuracy values.

For the *Appearance* based tasks in NeWT (which focus on a specific individual in the photo), we can see that there is a clear benefit to doing supervised pretraining on data from iNaturalist (using either iNat2018, iNat2021, or iNat2021 mini). *Species* classification, unsurprisingly, and *Age* have the biggest improvement followed by *Attribute* and then *Health*. We do not see the same benefit when using self-supervision for these *Appearance* based tasks. We instead find self-supervised models performing worse on average than ImageNet supervised features, even though they are trained on data from iNaturalist. Similarly, the *Behavior* tasks benefited from supervised pretraining on iNat datasets, but did not benefit from self-supervised pretraining. No method significantly improved performance on the *Context* tasks compared to supervised ImageNet features. All methods did relatively poorly on the two *Counting* tasks (0.59 baseline performance, note that chance is 50%). This could highlight the inappropriateness of using a classifier for detection style tasks, or it could highlight a particularly disappointing generalization behavior of these models. The SimCLR method trained on iNat2021 is a notable outlier in this experiment but the reason is unclear. Interestingly, all self-supervised models appear to provide a benefit over supervised ImageNet features and supervised iNat features for the *Gestalt* tasks, where the whole image needs to be analyzed as opposed to focusing on a particular subject.

Similar to the fine-grained datasets result, we see a reduced improvement between the iNat2021 large and mini datasets on the NeWT tasks as compared to evaluating on the iNat2021 test set. The SimCLR model achieved 0.678 mean accuracy using the iNat2021 mini split, and 0.689 with the full dataset. The supervised model went from 0.793 mean accuracy to 0.806. This result is surprising given the typical expectation of performance improvement when training with more data. Goyal et al. [18] perform experiments where they scale the amount of training data by a factor of 10, 50, and 100 and they see a larger performance gain for the ResNet50 model, albeit using Jigsaw [49] and Colorization [65] as pretext tasks, and Pascal VOC07 [14] as the downstream task. So either 5x more data is not a sufficient data increase, or self-supervision objectives like SimCLR behave differently.

While the experiments on existing fine-grained datasets in Figure 2.3 showed a benefit to using ImageNet over iNat2021 as the pretraining dataset for self-supervision, the NeWT results are much more mixed. For example SimCLR trained using ImageNet achieves better performance on average for the *Appearance - Age* tasks than SimCLR trained using iNat2021 (0.702 vs 0.688), but the results are flipped for the

Appearance - Species tasks (0.647 vs 0.661).

Discussion

We summarize our main findings:

Supervised ImageNet features are a strong baseline. The off-the-shelf supervised ImageNet features were often much better than the features derived from self-supervised models trained on either ImageNet or iNat2021. This applies to supervised iNat2021 features as well. It is currently easier to achieve downstream performance gains from a model trained with a supervised objective (assuming it is possible to get labels).

Fine-grained classification is challenging for self-supervised models. For most self-supervised methods performance is not close to supervised methods for the fine-grained datasets tested; see Figure 2.3. However, the SwAV method has closed the gap and is better in some cases (e.g. Stanford Cars). This trend did not hold when SwAV was trained on iNat2021 mini data.

Not all tasks are equal. Self-supervised features can be more effective compared to supervised ones for certain tasks (e.g. see the *Gestalt* tasks in NeWT in Figure 2.4). This highlights the value of benchmarking performance on a varied set of classification tasks, in addition to conventional object classification.

More data does not help methods as much for downstream tasks. While we observe a large boost in accuracy on the iNat2021 test set when we increase the amount of training data (+11 percentage points, see Tables 2.2 and 2.3), this boost is much smaller for both supervised and self-supervised models on the fine-grained datasets and NeWT (see the differences between iNat2021 large and mini for the supervised and SimCLR experiments in Figures 2.3 and 2.4).

Self-supervised ImageNet training settings do not necessarily generalize. The performance gap between supervised and self-supervised features on downstream tasks is closing when the feature extractor is trained on ImageNet. However, the gap between supervised and self-supervised features is much larger when the feature extractor is trained on iNat2021. This potentially points to self-supervised training settings being overfit to ImageNet e.g. via hyperparameters or the image augmentations used.

2.7 Conclusion

We presented, and benchmarked, the iNat2021 and NeWT datasets. The iNat2021 dataset contains 2.7M training images covering 10k species. As a large-scale image dataset we have shown its utility as a powerful pretraining network for a variety of existing fine-grained datasets as well as the NeWT dataset. Our NeWT dataset expands beyond the question of "What species is this?", to incorporate questions that challenge models to identify behaviors, health, and context questions as they relate to wildlife captured in photographs. Our experiments on NeWT reveal interesting performance differences between supervised and self-supervised learning methods. While supervised learning appears to still have an edge over existing self-supervised approaches, new methods are constantly being introduced by the research community. The iNat2021 and NeWT datasets should serve as a valuable resource for benchmarking these new techniques as they expose challenges not present in the standard datasets currently in use.

2.8 Acknowledgements

Thanks to the iNaturalist team and community for providing access to data, Eliot Miller and Mitch Barry for helping to curate NeWT, and to Pietro Perona for valuable feedback.

References

- [1] Thomas Berg et al. "Birdsnap: Large-scale fine-grained visual categorization of birds". In: *CVPR*. 2014.
- [2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. "Food-101 – Mining Discriminative Components with Random Forests". In: *ECCV*. 2014.
- [3] Q. Cao et al. "VGGFace2: A dataset for recognising faces across pose and age". In: *International Conference on Automatic Face and Gesture Recognition*. 2018.
- [4] Bradley J Cardinale et al. "Biodiversity loss and its impact on humanity". In: *Nature* (2012).
- [5] Mathilde Caron et al. "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *NeurIPS*. 2020.
- [6] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *ICML*. 2020.
- [7] Ting Chen et al. "Big self-supervised models are strong semi-supervised learners". In: *NeurIPS*. 2020.

- [8] Xinlei Chen et al. “Improved Baselines with Momentum Contrastive Learning”. In: *arXiv:2003.04297* (2020).
- [9] Grace Chu et al. “Geo-aware networks for fine-grained recognition”. In: *ICCV Workshops*. 2019.
- [10] Yin Cui et al. “Large scale fine-grained categorization and domain-specific transfer learning”. In: *CVPR*. 2018.
- [11] Carl Doersch, Abhinav Gupta, and Alexei A Efros. “Unsupervised visual representation learning by context prediction”. In: *ICCV*. 2015.
- [12] Jeff Donahue et al. “Decaf: A deep convolutional activation feature for generic visual recognition”. In: *ICML*. 2014.
- [13] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. “How Well Do Self-Supervised Models Transfer?” In: *CVPR*. 2021.
- [14] Mark Everingham et al. “The pascal visual object classes (voc) challenge”. In: *IJCV* (2010).
- [15] Timnit Gebru et al. “Fine-grained car detection for visual census estimation”. In: *AAAI*. 2017.
- [16] Ciara Giaimo. “Hold Me, Squeeze Me, Bite My Head”. In: *The New York Times* (Sept. 2020). URL: <https://www.nytimes.com/2020/09/29/science/lizard-sex-jaws.html>.
- [17] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. “Unsupervised representation learning by predicting image rotations”. In: *ICLR*. 2018.
- [18] Priya Goyal et al. “Scaling and benchmarking self-supervised visual representation learning”. In: *ICCV*. 2019.
- [19] Jean-Bastien Grill et al. “Bootstrap Your Own Latent-A New Approach to Self-Supervised Learning”. In: *NeurIPS* (2020).
- [20] Yandong Guo et al. “Ms-celeb-1m: A dataset and benchmark for large-scale face recognition”. In: *ECCV*. 2016.
- [21] Agrim Gupta, Piotr Dollar, and Ross Girshick. “Lvis: A dataset for large vocabulary instance segmentation”. In: *CVPR*. 2019.
- [22] Michael Gutmann and Aapo Hyvärinen. “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models”. In: *AISTATS*. 2010.
- [23] Raia Hadsell, Sumit Chopra, and Yann LeCun. “Dimensionality reduction by learning an invariant mapping”. In: *CVPR*. 2006.
- [24] Kaiming He et al. “Deep residual learning for image recognition”. In: *CVPR*. 2016.

- [25] Kaiming He et al. “Momentum contrast for unsupervised visual representation learning”. In: *CVPR*. 2020.
- [26] Saihui Hou, Yushan Feng, and Zilei Wang. “VegFru: A Domain-Specific Dataset for Fine-grained Visual Categorization”. In: *ICCV*. 2017.
- [27] Gary B. Huang et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. University of Massachusetts, Amherst, 2007.
- [28] *iNaturalist*. www.inaturalist.org, accessed Mar 7 2022.
- [29] *iNaturalist*. *Seek by iNaturalist*. <https://apps.apple.com/us/app/seek-by-inaturalist/id1353224144>. 2020.
- [30] *iNaturalist Challenge Datasets*. https://github.com/visipedia/inat_comp, accessed Nov 14 2020.
- [31] Menglin Jia et al. “Fashionpedia: Ontology, Segmentation, and an Attribute Localization Dataset”. In: *ECCV*. 2020.
- [32] Aditya Khosla et al. “Novel Dataset for Fine-Grained Image Categorization”. In: *First Workshop on Fine-Grained Visual Categorization*. 2011.
- [33] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. “Revisiting self-supervised visual representation learning”. In: *CVPR*. 2019.
- [34] Simon Kornblith, Jonathon Shlens, and Quoc V Le. “Do better imagenet models transfer better?” In: *CVPR*. 2019.
- [35] Ivan Krasin et al. “OpenImages: A public dataset for large-scale multi-label and multi-class image classification.” In: *Dataset available from https://storage.googleapis.com/openimages* (2017).
- [36] Jonathan Krause et al. “3d object representations for fine-grained categorization”. In: *ICCV Workshops*. 2013.
- [37] Jonathan Krause et al. “The unreasonable effectiveness of noisy data for fine-grained recognition”. In: *ECCV*. 2016.
- [38] Ranjay Krishna et al. “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *IJCV* (2017).
- [39] Neeraj Kumar et al. “Leafsnap: A computer vision system for automatic plant species identification”. In: *ECCV*. 2012.
- [40] Tsung-Yi Lin et al. “Microsoft COCO: Common objects in context”. In: *ECCV*. 2014.
- [41] Yen-Liang Lin et al. “Jointly optimizing 3d model fitting and fine-grained classification”. In: *ECCV*. 2014.
- [42] Jiongxin Liu et al. “Dog breed classification using part localization”. In: *ECCV*. 2012.

- [43] Oisín Mac Aodha, Elijah Cole, and Pietro Perona. “Presence-only geographical priors for fine-grained image classification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9596–9606. DOI: 10.48550/arXiv.1906.05272.
- [44] Subhransu Maji et al. “Fine-grained visual classification of aircraft”. In: *arXiv:1306.5151* (2013).
- [45] Ishan Misra and Laurens van der Maaten. “Self-supervised learning of pretext-invariant representations”. In: *CVPR*. 2020.
- [46] Alejandro Newell and Jia Deng. “How Useful is Self-Supervised Pretraining for Visual Tasks?” In: *CVPR*. 2020.
- [47] Maria-Elena Nilsback and Andrew Zisserman. “A visual vocabulary for flower classification”. In: *CVPR*. 2006.
- [48] Maria-Elena Nilsback and Andrew Zisserman. “Automated flower classification over a large number of classes”. In: *Indian Conference on Computer Vision, Graphics & Image Processing*. 2008.
- [49] Mehdi Noroozi and Paolo Favaro. “Unsupervised learning of visual representations by solving jigsaw puzzles”. In: *ECCV*. 2016.
- [50] O. M. Parkhi et al. “Cats and Dogs”. In: *CVPR*. 2012.
- [51] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. “Deep Face Recognition.” In: *BMVC*. 2015.
- [52] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *IJCV* (2015).
- [53] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. “When Does Self-supervision Improve Few-shot Learning?” In: *ECCV*. 2020.
- [54] Brian L Sullivan et al. “eBird: A citizen-based bird observation network in the biological sciences”. In: *Biological Conservation* (2009).
- [55] Bart Thomee et al. “YFCC100M: The new data in multimedia research”. In: *Communications of the ACM* (2016).
- [56] Cornell University. *Merlin Bird Id*. <https://apps.apple.com/us/app/merlin-bird-id-by-cornell-lab/id773457673>. 2020.
- [57] Grant Van Horn et al. “Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection”. In: *CVPR*. 2015.
- [58] Grant Van Horn et al. “The iNaturalist Species Classification and Detection Dataset”. In: *CVPR*. 2018.
- [59] Andrea Vedaldi et al. “Understanding objects in detail with fine-grained attributes”. In: *CVPR*. 2014.

- [60] Catherine Wah et al. “The caltech-ucsd birds-200-2011 dataset”. In: (2011).
- [61] Zhirong Wu et al. “Unsupervised feature learning via non-parametric instance discrimination”. In: *CVPR*. 2018.
- [62] Linjie Yang et al. “A large-scale car dataset for fine-grained categorization and verification”. In: *CVPR*. 2015.
- [63] Jason Yosinski et al. “How transferable are features in deep neural networks?” In: *NeurIPS*. 2014.
- [64] Xiaohua Zhai et al. “A large-scale study of representation learning with the visual task adaptation benchmark”. In: *arXiv:1910.04867* (2019).
- [65] Richard Zhang, Phillip Isola, and Alexei A Efros. “Colorful image colorization”. In: *ECCV*. 2016.
- [66] Bolei Zhou et al. “Places: A 10 million Image Database for Scene Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).
- [67] T. Züchner, C.J. Sharpe, and P.F.D. Boesman. “Pink-throated Brilliant (*Heliodoxa gularis*)”. In: *Birds of the World* (2020). URL: <https://birdsoftheworld.org/bow/species/pitbri1>.

*Chapter 3***WHEN DOES CONTRASTIVE VISUAL REPRESENTATION
LEARNING WORK?**

- [1] Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge Belongie. “When does contrastive visual representation learning work?” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14755–14764. DOI: [10.48550/arXiv.2105.05837](https://doi.org/10.48550/arXiv.2105.05837).

3.1 Abstract

Recent self-supervised representation learning techniques have largely closed the gap between supervised and unsupervised learning on ImageNet classification. While the particulars of pretraining on ImageNet are now relatively well understood, the field still lacks widely accepted best practices for replicating this success on other datasets. As a first step in this direction, we study contrastive self-supervised learning on four diverse large-scale datasets. By looking through the lenses of data quantity, data domain, data quality, and task granularity, we provide new insights into the necessary conditions for successful self-supervised learning. Our key findings include observations such as (i) the benefit of additional pretraining data beyond 500k images is modest, (ii) adding pretraining images from another domain does not lead to more general representations, (iii) corrupted pretraining images have a disparate impact on supervised and self-supervised pretraining, and (iv) contrastive learning lags far behind supervised learning on fine-grained visual classification tasks.

3.2 Introduction

Self-supervised learning (SSL) techniques can now produce visual representations which are competitive with representations generated by fully supervised networks for many downstream tasks [18]. This is an important milestone for computer vision, as removing the need for large amounts of labels at training time has the potential to scale up our ability to address challenges in domains where supervision is currently too difficult or costly to obtain. However, with some limited exceptions, the vast majority of current state-of-the-art approaches are developed and evaluated

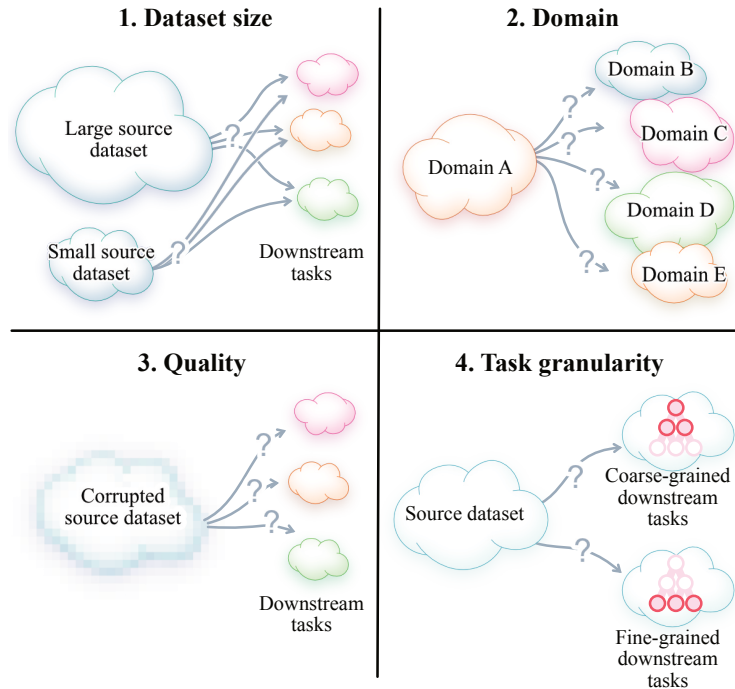


Figure 3.1: **What conditions are necessary for successful self-supervised pre-training on domains beyond ImageNet?** We investigate the impact of self-supervised and supervised training *dataset size*, the downstream *domain*, *image quality*, and the *granularity* of downstream classification tasks.

on standard datasets like ImageNet [40]. As a result, we do not have a good understanding of how well these methods work when they are applied to other datasets.

Under what conditions do self-supervised contrastive representation learning methods produce "good" visual representations? This is an important question for computer vision researchers because it adds to our understanding of SSL and highlights opportunities for new methods. This is also an important question for domain experts with limited resources who might be interested in applying SSL to real-world problems. With these objectives in mind, we attempt to answer the following questions:

(i) What is the impact of data quantity? How many unlabeled images do we need for pretraining, and when is it worthwhile to get more? How much labeled data do we need for linear classifier training or end-to-end fine-tuning on a downstream task? In which regimes do self-supervised features rival those learned from full supervision?

(ii) What is the impact of the pretraining domain? How well do self-supervised

representations trained on one domain transfer to another? Can we learn more general representations by combining datasets? Do different pretraining datasets lead to complementary representations?

(iii) What is the impact of data quality? How robust are self-supervised methods to training time image corruption such as reduced resolution, compression artifacts, or noise? Does pretraining on corrupted images lead to poor downstream performance on uncorrupted images?

(iv) What is the impact of task granularity? Does SSL result in features that are only effective for "easy" classification tasks, or are they also useful for more challenging, "fine-grained" visual concepts?

We address the above questions through extensive quantitative evaluation across four diverse large-scale visual datasets (see Figure 3.1). We make several interesting observations and recommendations including:

- For an ImageNet-scale dataset, decreasing the amount of unlabeled training data by half (from 1M to 500k images) only degrades downstream classification performance by 1-2% (Figure 3.2). In many contexts this trade-off is reasonable, allowing for faster and cheaper pretraining. This also indicates that current self-supervised methods coupled with standard architectures may be unable to take advantage of very large pretraining sets.
- Self-supervised representations that are learned from images from the same domain as the test domain are much more effective than those learned from different domains (Table 3.1). Self-supervised training on our current datasets may not be sufficient to learn representations that readily generalize to many contexts.
- Neither (i) combining datasets before pretraining (Table 3.2) nor (ii) combining self-supervised features learned from different datasets (Table 3.3) leads to significant performance improvements. More work may be required before self-supervised techniques can learn highly generalizable representations from large and diverse datasets.
- Pretraining on corrupted images affects supervised and self-supervised learning very differently (Figure 3.4). For instance, self-supervised representations are surprisingly sensitive to image resolution.
- Current self-supervised methods learn representations that can easily disambiguate coarse-grained visual concepts like those in ImageNet. However, as

the granularity of the concepts becomes finer, self-supervised performance lags further behind supervised baselines (Figure 3.5). The contrastive loss may lead to coarse-grained features which are insufficient for fine-grained tasks.

3.3 Related Work

SSL for visual representations. Early self-supervised representation learning methods typically centered around solving hand-designed "pretext tasks" like patch location prediction [16], rotation prediction [20], inpainting [37], cross-channel reconstruction [59], sorting sequences of video frames [33], solving jigsaw puzzles [35], or colorization [58]. However, more recent work has explored *contrastive learning-based* approaches where the pretext task is to distinguish matching and non-matching pairs of augmented input images [27, 36, 48]. The prototypical example is SimCLR [9, 10], which is trained to identify the matching image using a cross-entropy loss. Other variations on the contrastive SSL framework include using a momentum encoder to provide large numbers of negative pairs (MoCo) [26, 12], adaptively scaling the margin in MoCo (EqCo) [62], and contrasting clustering assignments instead of augmented pairs (SwAV) [7]. Moving beyond the contrastive loss entirely, some papers recast the problem in a "learning-to-rank" framework (S2R2) [52], use simple feature prediction (SimSiam) [11], or predict the output of an exponential moving average network (BYOL) [24]. [4] investigates the role of negatives in contrastive learning, though we note that BYOL and SimSiam avoid using negatives explicitly. In this work, our focus is on self-supervised visual classification. We do not explore alternative settings such as supervised contrastive learning [31], contrastive learning in non-vision areas like language [39] or audio [41], or other methods that aim to reduce the annotation burden for representation learning such as large-scale weak supervision [34].

SSL beyond ImageNet. ImageNet classification has long been viewed as the gold standard benchmark task for SSL, and the gap between supervised and self-supervised performance on ImageNet has steadily closed over the last few years [9, 26, 24, 7]. There is now a growing expectation that SSL should reduce our dependence on manual supervision in challenging and diverse domains which may *not* resemble the traditional object classification setting represented by ImageNet. A number of papers have studied how well self-supervised representations pre-trained on ImageNet perform on downstream tasks like fine-grained species classification [56], semantic segmentation [5], scene understanding [24], and instance

segmentation [26].

More recently, researchers have begun to study the effectiveness of contrastive learning when *pretraining* on datasets other than ImageNet. In the case of remote sensing, the unique properties of the data have motivated the development of domain-specific contrastive learning techniques [30, 2]. In the medical domain, where images tend to be very dissimilar to ImageNet, it has been shown that contrastive pretraining on domain-specific images leads to significant gains compared to pretraining on ImageNet [43, 10]. [32] compared the representations learned from five different datasets, and showed that in most cases the best performing representations came from pretraining on similar datasets to the downstream task. In the case of fine-grained data, [51] found that contrastive pretraining on images of animals and plants did not lead to superior performance on downstream bird classification compared to pretraining on ImageNet. These apparently conflicting observations may be explained by the relationship between the pretraining and downstream data distributions, which we investigate in our experiments. [60] and [50] pretrained on several different datasets and showed that there was surprisingly little impact on downstream detection and segmentation performance, unless synthetic data was used for pretraining [60]. [47] pretrained on very large datasets (JFT-300M [44] and YFCC100M [46]), but did not observe an improvement over ImageNet pretraining in the standard regime.

We build on the above analysis by performing controlled, like-for-like, comparisons of SSL on several large datasets. This allows us to separate dataset-specific factors from general patterns in SSL performance, and deliver new insights into the necessary conditions for successful pretraining.

Analysis of SSL. A number of works have explored questions related to the conditions under which SSL is successful. [42] showed that self-supervised representations generalize better than supervised ones when the downstream concepts of interest are less semantically similar to the pretraining set. [18] showed that contrastive pretraining on ImageNet performs well on downstream tasks related to object recognition in natural images, while leaving more general study of pretraining in different domains to future work. While these works show that SSL on ImageNet can be effective, our experiments demonstrate that current SSL methods can perform much worse than supervised baselines on non-ImageNet domains, e.g. fine-grained classification.

Existing work has also investigated other aspects of SSL, e.g. [38] examined the

invariances learned, [8] showed that easily learned features can inhibit the learning of more discriminative ones, [60, 50, 9] explored the impact of different image augmentations, [8, 50] compared representations from single vs. multi-object images, and [9, 21] varied the backbone model capacity. Most relevant to our work are studies that vary the amount of data in the pretraining dataset, e.g. [57, 60, 32, 50]. We extend this analysis by presenting a more detailed evaluation of the impact of the size of the unlabeled and labeled datasets, and investigate the role of data quality, data domain, and task granularity.

3.4 Methods

Datasets. We perform experiments on four complementary large-scale datasets: ImageNet [15], iNat21 [50], Places365 [61], and GLC20 [13]. Collectively, these datasets span many important visual properties, including curated vs. "in-the-wild" images, fine- vs. coarse-grained categories, and object-centric images vs. scenes. Each dataset has at least one million images, which allows us to make fair comparisons against the traditional ImageNet setting. ImageNet (1.3M images, 1k classes) and Places365 (1.8M images, 365 classes) are standard computer vision datasets, so we will not describe them in detail. For ImageNet, we use the classic ILSVRC2012 subset of the full ImageNet-21k dataset. For Places365, we use the official variant "Places365-Standard (small images)" where all images have been resized to 256x256. iNat21 (2.7M images, 10k classes) contains images of plant and animal species and GLC20 (1M images, 16 classes) consists of remote sensing images. As both are recent datasets, we discuss them in the supplementary material.

Fixed-size subsets. For some experiments we control for dataset size by creating subsampled versions of each dataset with sizes: 1M, 500k, 250k, 125k, and 50k images. We carry out this selection only once, and the images are chosen uniformly at random. We refer to these datasets using the name of the parent dataset followed by the number of images in parentheses, e.g. ImageNet (500k). Note that subsets of increasing size are *nested*, so e.g. ImageNet (500k) includes all of the images in ImageNet (250k). These subsets are also *static* across experiments, e.g. ImageNet (500k) always refers to the same set of 500k images. With the exception of Figures 3.2 and 3.3, we use the full dataset for any type of supervised training (i.e. linear evaluation, fine tuning, or supervised training from scratch). We always report results on the same test set for a given dataset, regardless of the training subset used.

Training details. All experiments in this paper are based on a ResNet-50 [25] backbone, which is standard in the contrastive learning literature [9, 7, 26]. We primarily perform experiments on SimCLR [9], a simple and popular contrastive learning method that contains all the building blocks for state-of-the-art self-supervised algorithms. We follow the standard protocol of first training with self-supervision alone and then evaluating the learned features using linear classifiers or end-to-end fine-tuning. Unless otherwise specified, we use hyperparameter settings based on [9] for all methods and datasets. While this may not lead to maximal performance, it is likely to be representative of how these methods are used in practice. Due to the high computational cost of contrastive pretraining, extensive hyperparameter tuning is not feasible for most users. We also consider MoCo [26] and BYOL [24] in Figure 3.3. Full training details are provided in the supplementary material.

3.5 Experiments

We now describe our experiments in which we investigate the impact of data quantity, data domain, data quality, and task granularity on the success of contrastive learning.

Data quantity

First we consider the question of how much data is required to learn a "good" representation using SSL. There are two important notions of data quantity: (i) the number of *unlabeled images* used for pretraining and (ii) the number of *labeled images* used to subsequently train a classifier. Since labels are expensive, we would like to learn representations that generalize well with as few labeled images as possible. While unlabeled images are cheap to acquire, they still incur a cost because pretraining time is proportional to the size of the pretraining set. To understand when SSL is cost-effective, we need to understand how performance depends on these two notions of data quantity.

To study this question, we pretrain SimCLR using different numbers of unlabeled images. Each pretrained representation is then evaluated using different numbers of labeled images. In Figure 3.2 we present these results for iNat21 (left column), ImageNet (center column), and Places365 (right column). We also include results for supervised training from scratch (in black). We show linear evaluation results in the top row and corresponding fine-tuned results in the bottom row. Each curve in a figure corresponds to a different pretrained representation. The points along a curve correspond to different amounts of supervision used to train a linear classifier or fine-tune the network.

There is little benefit beyond 500k pretraining images. The gap between the 500k (blue) and 1M (orange) pretraining image curves is typically less than 1-2% in top-1 accuracy. This means that for a dataset with one million images, we can trade a small decrease in accuracy for a 50% decrease in pretraining time. If a 2-4% top-1 accuracy drop is acceptable, then the pretraining set size can be reduced by a factor of four (from 1M to 250k). However, the difference between 50k (pink) pretraining images and 250k (green) pretraining images is substantial for each dataset, often in excess of 10% top-1 accuracy. We conclude that SimCLR seems to saturate well before we get to ImageNet-sized pretraining sets. This is consistent with observations from the supervised learning literature, though more images are required to reach saturation [34].

Self-supervised pretraining can be a good initializer when there is limited supervision available. In the bottom row of Figure 3.2 we see that when only 10k or 50k labeled images are available, fine-tuning a SimCLR representation is significantly better than training from scratch. When supervision is plentiful, fine-tuned SimCLR representations achieve performance similar to supervised training from scratch. It is interesting to compare this to findings from the supervised setting which suggest that networks which are initially trained on distorted (i.e. augmented) images are unable to recover when subsequently trained with undistorted ones [1].

Self-supervised representations can approach fully supervised performance for some datasets, but only by using lots of labeled images. The ultimate goal of SSL is to match supervised performance without the need for large amounts of labeled data. Suppose we consider the right-most point on the black curves in Figure 3.2 as a proxy for "good" supervised performance. Then in both the linear and fine-tuned cases, the gap between SimCLR (pretrained on 1M images) and "good" supervised performance is quite large unless well over 100k labeled images are used. For instance, the gap between "good" supervised performance and a classifier trained using 50k labeled images on top of SimCLR (1M) is around 11% (11%) for Places365, 23% (21%) for ImageNet, and 58% (56%) for iNat21 in the linear (and fine-tuned) case. Although SSL works well when lots of supervision is available, further innovation is needed to improve the utility of self-supervised representations in the low-to-moderate supervision regime.

iNat21 is a valuable SSL benchmark. Figure 3.2 shows a surprisingly large gap (~ 30%) between supervised and self-supervised performance on iNat21 in the high supervision regime. In Figure 3.3 we see that other SSL methods exhibit similar

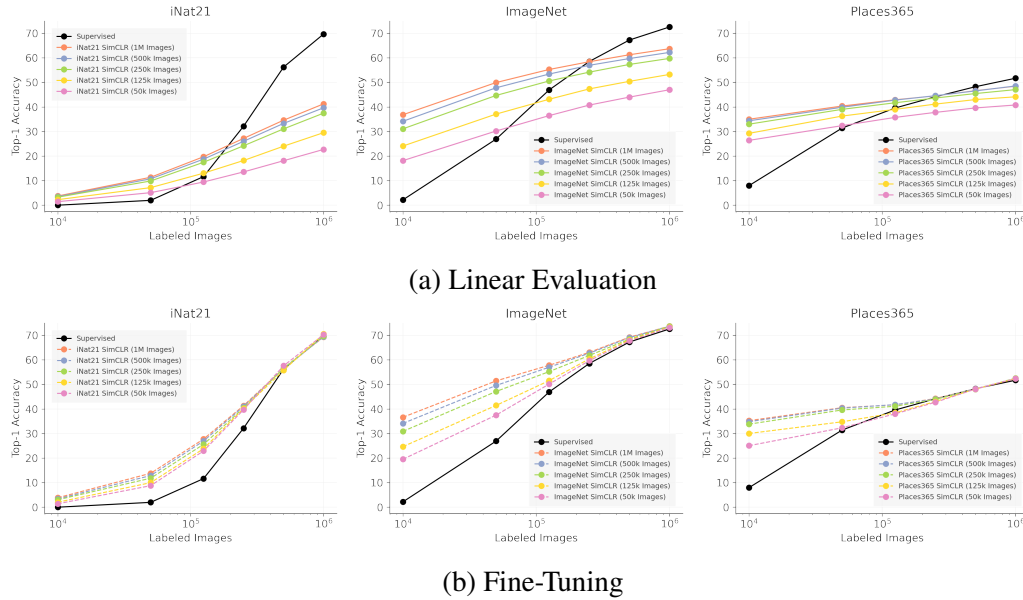


Figure 3.2: **How much data does SimCLR need?** Linear evaluation results (top row) and fine-tuning results (bottom row) as a function of the number of *unlabeled images* used for pretraining and the number of *labeled images* used for downstream supervised training. The "Supervised" curve (black) corresponds to training from scratch on different numbers of labeled images. It is the same for the top and bottom plots in each column. Most SSL papers focus on the "high data" regime, using $\sim 10^6$ images (e.g. all of ImageNet) for both pretraining and classifier supervision, but there are significant opportunities for improvement in the "low-data" regime. Even with 10^6 labeled images for linear classifier training, SimCLR performs far worse than supervised learning on iNat21, suggesting that iNat21 could be a more useful SSL benchmark than ImageNet in future.

limitations. The newer BYOL outperforms MoCo and SimCLR, but a considerable gap ($\sim 25\%$) remains. The high supervised performance shows that the task is possible, yet the self-supervised performance remains low. It seems that iNat21 reveals challenges for SSL that are not apparent in ImageNet, and we believe it is a valuable benchmark for future SSL research.

Data domain

In the previous section we observed that increasing the pretraining set size yields rapidly diminishing returns. In this section we consider a different design choice: *what kind of images* should we use for pretraining? Since most contrastive learning papers only pretrain on ImageNet, this question has not received much attention. We take an initial step towards an answer by studying the properties of SimCLR representations derived from four pretraining sets drawn from different domains.

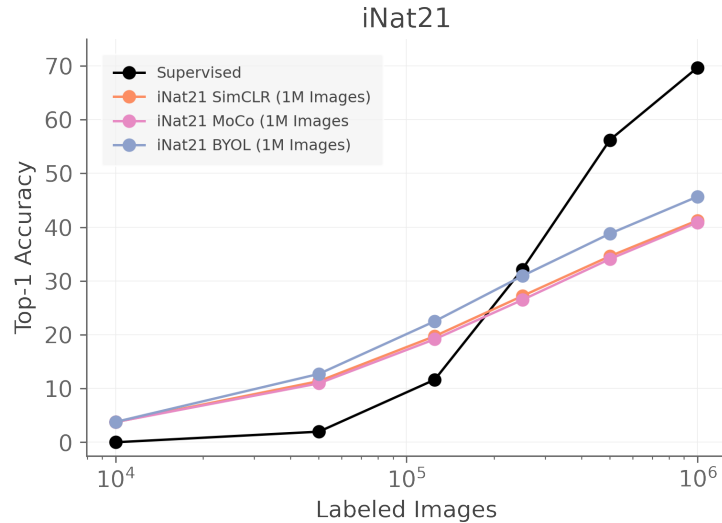


Figure 3.3: **How does SimCLR compare to other self-supervised methods?** Linear evaluation results on iNat21 for SimCLR, MoCo, and BYOL. All methods are pretrained on 1M images for 1000 epochs and follow the same linear evaluation protocol. The more recent BYOL performs better than the others, but a large gap remains to supervised performance.

We train SimCLR on iNat21 (1M), ImageNet (1M), Places365 (1M), and GLC20 (1M). By holding the pretraining set size constant, we aim to isolate the impact of the different visual domains. We present in-domain and cross-domain linear evaluation results for each representation in Table 3.1. In Table 3.2 we consider the effect of pretraining on *pooled datasets*, i.e. new image collections built by shuffling together existing datasets. Finally, in Table 3.3 we study different *fused representations*, which are formed by concatenating the outputs of different feature extractors.

Pretraining domain matters. In Table 3.1 we see that in-domain pretraining (diagonal entries) consistently beats cross-domain pretraining (off-diagonal entries). The gap can be surprisingly large, e.g. in-domain pretraining provides a 12% boost on iNat21 compared to the best cross-domain pretraining (ImageNet). One might have expected that a visually diverse dataset like ImageNet would lead to a better self-supervised representation than a more homogeneous dataset like GLC20 (even when evaluating on GLC20) but this is not what we observe.

The off-diagonal entries of Table 3.1 show that training SimCLR on ImageNet leads to the best cross-domain performance, while GLC20 leads to the worst cross-domain performance. Since the pretraining protocols and dataset sizes are held constant, we suggest that the characteristics of the image sets themselves are responsible

Pretraining	iNat21	ImageNet	Places365	GLC20
iNat21 (1M) SimCLR	0.493	<u>0.519</u>	0.416	0.707
ImageNet (1M) SimCLR	<u>0.373</u>	0.644	<u>0.486</u>	<u>0.716</u>
Places365 (1M) SimCLR	0.292	0.491	0.501	0.693
GLC20 (1M) SimCLR	0.187	0.372	0.329	0.769
Supervised (All Images)	0.791	0.741	0.539	0.826

Table 3.1: **Does pretraining domain matter?** Linear evaluation results for representations derived from different million-image datasets. We train the linear classifiers using the full training sets. The results in the "Supervised" row correspond to supervised training from scratch on the full training set. We report MAP for GLC20 and top-1 accuracy for other datasets. In all cases, in-domain pretraining outperforms cross-domain pretraining. In each column we highlight the **best** and second-best results.

for the differences we observe. The strong cross-domain performance of SimCLR pretrained on ImageNet may be due to *semantic similarity* — perhaps it is better to pretrain on a dataset that is semantically similar to the downstream task, even in a self-supervised context. This makes sense because there are classes in ImageNet that are similar to classes in iNat21 (animals) and Places365 (scenes). This also explains the weak performance of GLC20, since remote sensing imagery is not similar to the other datasets.

Adding cross-domain pretraining data does not necessarily lead to more general representations. We have seen that pretraining on different domains leads to representations with significantly differing capabilities. This leads to a natural question: *what happens if we combine our datasets and then learn a representation?*

Table 3.2 gives linear evaluation results for SimCLR pretrained on different "pooled" datasets. In each row, n images from dataset A and m images from dataset B are shuffled together to produce a pretraining set of size $n + m$. For instance, the pretraining dataset in the first row of Table 3.2 consists of 250k iNat21 images and 250k ImageNet images shuffled together.

If we compare the "In-Domain (500k)" row against the (equally sized) pooled datasets in the first three rows of Table 3.2, we see that the in-domain pretraining on 500k images is always better. Similarly, the "In-Domain (1M)" row beats the 1M-image pooled dataset (consisting of 250k images from the four datasets). The more diverse pooled pretraining sets always lead to worse performance compared to the more homogeneous pretraining sets of the same size.

Table 3.2 also allows us to say whether it is worthwhile to *add* pretraining data

Pretraining				Evaluation		
<i>iNat21</i>	<i>ImageNet</i>	<i>Places365</i>	<i>GLC20</i>	<i>iNat21</i>	<i>ImageNet</i>	<i>Places365</i>
250k	250k	-	-	0.444	0.597	0.467
-	250k	250k	-	0.334	0.596	0.490
250k	-	250k	-	0.428	0.531	0.483
250k	250k	250k	250k	0.410	0.574	0.482
In-Domain (250k)				0.451	0.608	0.485
In-Domain (500k)				0.477	0.629	0.499
In-Domain (1M)				0.493	0.644	0.501

Table 3.2: **The effect of dataset pooling.** Linear evaluation results for self-supervised representations derived from *pooled datasets*, where two or more datasets are shuffled together. We train the linear classifiers using the full training sets. The "In-Domain" results correspond to pretraining on subsets of the dataset named at the top of the column. Pooling datasets increases pretraining set size and diversity, but we find that performance *decreases* relative to comparable in-domain pretraining. The "In-Domain (1M)" row corresponds to the diagonal entries of Table 3.1.

from a different domain (as opposed to swapping out some in-domain data for some data from a different domain, as we have been discussing so far). The "In-Domain (250k)" row is better than the 1M-image pooled dataset and almost all of the 500k-image pooled datasets. It seems that adding pretraining data from a different domain typically *hurts* performance. In contrast, Figure 3.2 shows that increasing the amount of *in-domain* pretraining data consistently improves performance.

We hypothesize that the reason for this lackluster performance is that diverse images are easier to tell apart, which makes the contrastive pretext task easier. If the contrastive task is too easy, the quality of the representation suffers [4, 8]. While more investigation is needed, the fact that increasing pretraining data diversity can hurt performance suggests a "diversity-difficulty trade-off" that should be considered when creating pretraining sets for SSL.

Self-supervised representations can be largely redundant. From Table 3.1 it is clear that pretraining on different datasets leads to representations that differ significantly. For instance, iNat21 SimCLR beats ImageNet SimCLR on iNat21 (+12.4%) and ImageNet SimCLR beats iNat21 SimCLR on ImageNet (+12.7%). Do these representations learn complementary information, or do they just capture the same information to different degrees?

To probe this question we concatenate features from different pretrained networks

ImageNet	iNat21	Dim.	ImageNet	iNat21
SimCLR	-	2048	0.647	0.380
-	SimCLR	2048	0.520	0.506
Sup.	-	2048	0.711	0.434
-	Sup.	2048	0.490	<u>0.769</u>
Sup.	Sup.	4096	0.712	0.772
SimCLR	SimCLR	4096	0.641	0.520
SimCLR & Sup.	-	4096	0.720	0.472
-	SimCLR & Sup.	4096	0.527	0.772
SimCLR	Sup.	4096	0.605	<u>0.769</u>
Sup.	SimCLR	4096	<u>0.717</u>	0.553

Table 3.3: **The effect of representation fusion.** Linear evaluation results for different combinations of supervised and self-supervised representations on ImageNet and iNat21. We train the linear classifiers using the full training sets. For comparability, the in-domain supervised results in this table (ImageNet Sup. evaluated on ImageNet and iNat21 Sup. evaluated on iNat21) are for linear classifiers trained on representations learned from full supervision. "Dim." is the representation dimensionality. In each column we highlight the **best** and second-best results.

and carry out linear evaluation on these "fused" representations. In Table 3.3 we present linear evaluation results for fused representations on ImageNet and iNat21. Combining ImageNet SimCLR and iNat21 SimCLR is worse than ImageNet SimCLR alone on ImageNet (-0.6%), but better than iNat21 SimCLR alone on iNat21 (+1.4%). These effects are small relative to the $> 12\%$ difference between ImageNet SimCLR and iNat21 SimCLR. This suggests that the two self-supervised representations are largely redundant.

There is a larger effect when combining supervised and self-supervised representations. For iNat21, adding ImageNet Sup. (i.e. supervised ImageNet features) on top of iNat21 SimCLR improves performance significantly (+4.7%). However, adding iNat21 Sup. on top of ImageNet SimCLR actually decreases performance (-4.2%). These results are consistent with the hypothesis that dataset semantics are important even for SSL. Since ImageNet is semantically broader than iNat21 (ImageNet has animal classes, but also many other things), features learned from ImageNet (supervised or self-supervised) should be more helpful for iNat21 than vice-versa.

Data quality

We have seen that the characteristics of the pretraining data can have a significant impact on the quality of self-supervised representations. In this section we dig deeper into this question by studying the impact of pretraining on artificially degraded

images. This serves two purposes. First, this is a practical question since there are many settings where image quality issues are pervasive e.g. medical imaging [45] or camera trap data [3]. Second, it can help us understand the robustness properties of SSL.

To create a corrupted dataset we apply a particular image corruption to each image in the dataset. This is a one-time offline preprocessing step, so corruptions that have a random component are realized only once per image. Given a corrupted dataset we then pretrain as normal. During linear evaluation, we use the original clean images for training and testing, i.e. the corrupted images are only used for pretraining.

In Figure 3.4 we present linear evaluation results on ImageNet for a simple but diverse set of corruptions. The zero point corresponds to pretraining on uncorrupted images, and we measure how much performance drops when pretraining on corrupted images. The "Salt and Pepper" corruption is salt and pepper noise applied independently to each pixel, in each channel, with probability 0.01. The "JPEG" corruption is JPEG compression with a very low quality level of 10. For "Resize", we resize each image so that the short side is 256 pixels while preserving the aspect ratio. This reduces the resolution of the crops used for training. For our downsampling corruptions, we follow the resize operation with downsampling by 2x or 4x and then upsampling by the same factor. This holds constant the image size and the fraction of the image occupied by each object, but reduces resolution. Implementation details and examples can be found in the supplementary.

Image resolution is critical for SSL. "Downsample (2x)" and "Downsample (4x)" are by far the most damaging corruptions for SimCLR, reducing accuracy by around 15% and 34%, respectively. Since SimCLR already involves extreme cropping, we might expect more robustness to changes in image resolution. This finding could be partially explained by the difficulty of generalizing to higher-resolution images during linear classifier training [49]. However, supervised pretraining faces the same challenge but the effect of downsampling is much less dramatic. This suggests that the performance drop is due to deficiencies in the features learned by SimCLR.

SSL is relatively robust to high-frequency noise. "JPEG" and "Salt & Pepper" both add high-frequency noise to the image. For SimCLR, these corruptions have a much milder impact than the downsampling corruptions. One possible explanation is that downsampling destroys texture information, which is known to be a particularly important signal for convolutional neural networks [19, 29]. For supervised pretraining the ranking of corruptions is very different, with "JPEG" landing

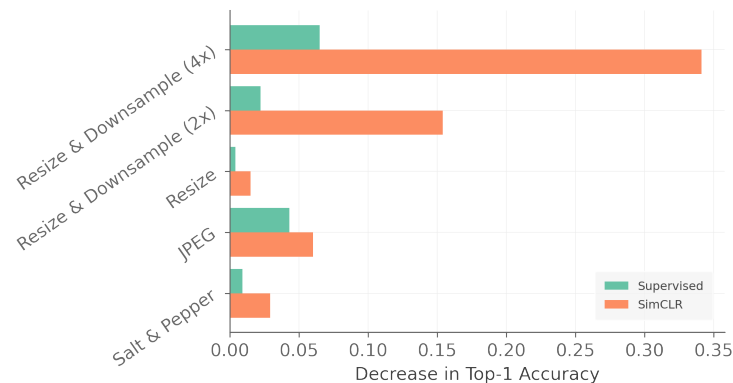


Figure 3.4: **What is the effect of pretraining image corruption?** Decrease in linear evaluation accuracy on ImageNet due to pretraining on corrupted versions of the ImageNet training set. The zero point corresponds to pretraining (supervised or SimCLR) on uncorrupted images followed by linear evaluation. "Supervised" and "SimCLR" have different zero points. All linear classifiers are trained using the full uncorrupted ImageNet training set.

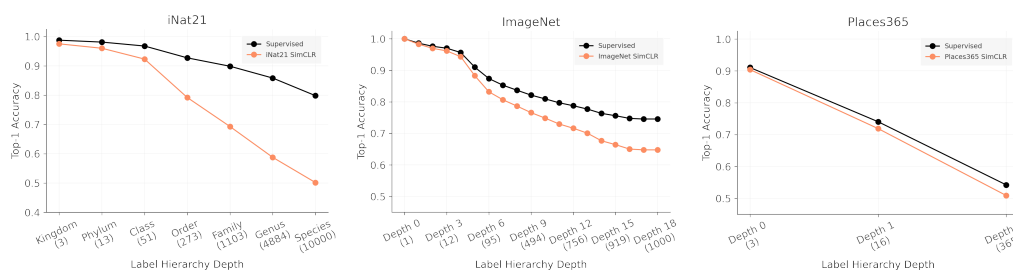


Figure 3.5: **How does performance depend on label granularity?** Linear evaluation at different levels of label granularity for iNat21, ImageNet, and Places365. Each plot compares supervised learning from scratch against a linear classifier trained on top of in-domain SimCLR. Both are trained using the full training sets. We plot top-1 accuracy against label granularity, which is more fine-grained as we move from left to right. The numbers on the x-axis are the class counts at a given level of the label hierarchy. We do not re-train at coarser granularity levels, we just change the evaluation label set. The definitions of the hierarchy levels are given in the supplementary material.

between 2x and 4x downsampling.

Task granularity

We have seen that the properties of pretraining datasets are important for determining the utility of self-supervised representations. But are there downstream tasks for which self-supervised representations are particularly well or poorly suited? We consider *fine-grained classification* and show that classification performance depends on *task granularity*, i.e. how fine or coarse the labels are. While there

are formal methods for measuring dataset granularity [14], we claim by intuition that iNat21 is more fine-grained than ImageNet, which is more fine-grained than Places365.

In Figure 3.5 we use label hierarchies (which are available for ImageNet, iNat21, and Places365) to explicitly study how performance depends on label granularity. We treat "distance from the root of the hierarchy" as a proxy for granularity, so labels further from the root are considered to be more fine-grained. We perform (i) linear classifier training (for SimCLR) and (ii) end-to-end training from scratch (for "Supervised") using the labels at the finest level of the taxonomy and re-compute accuracy values as we progressively coarsen the predictions and labels. We do not re-train at each level of granularity. A complete description of this process can be found in the supplementary materials.

The performance gap between SSL and supervised learning grows as task granularity becomes finer. We start with the iNat21 results in Figure 3.5. The supervised and SimCLR pretrained models perform similarly at the coarsest levels of the label hierarchy ("Kingdom"). Both models perform worse as task granularity increases, but the SimCLR model degrades much more rapidly ("Species"). This suggests that SimCLR may fail to capture fine-grained semantic information as effectively as supervised pretraining. We also observe a growing supervised/self-supervised gap for ImageNet and Places365. The magnitude of this gap seems to track dataset granularity, since iNat21 (most fine-grained) has the largest gap and Places365 (least fine-grained) has the smallest gap. The fact that supervised learning achieves high performance on iNat21 while SSL lags behind suggests that iNat21 could be a valuable benchmark dataset for the next phase of SSL research.

Are the augmentations destructive? State-of-the-art contrastive learning techniques are designed for ImageNet, so the default augmentation policy may be poorly tuned for other datasets [56]. For instance, if color is a key fine-grained feature for species classification then the "color jitter" augmentation used by SimCLR may destroy important information for iNat21 classification. Could this explain the rapid drop in performance exhibited by iNat21 SimCLR for fine-grained classes? Notice that there is a similar, though less extreme, fine-grained performance drop for ImageNet SimCLR in Figure 3.5. Since the ImageNet-tuned augmentations are presumably not destructive for ImageNet, it does not seem likely that this fully explains our observations.

Does contrastive learning have a coarse-grained bias? We hypothesize that

the contrastive loss tends to cluster images based on overall visual similarity. The intuition is that fine-grained features are often subtle, and subtle features are unlikely to be very useful for distinguishing between pairs of images in the contrastive pretext task. If our hypothesis is correct then the boundaries between different clusters would not be well-aligned with the boundaries between fine-grained classes. This effect could be overlooked when evaluating on coarse-grained classes, but would become apparent on a more fine-grained task. Additional analysis is required to fully understand this "granularity gap" in SSL, which we leave to future work.

3.6 Conclusion

We have presented a comprehensive set of experiments to address several aspects of the question: *when does contrastive visual representation learning work?* In Section 3.5 we found that we need fewer than 500k pretraining images before encountering severe diminishing returns. However, even the best self-supervised representations are still much worse than peak supervised performance without hundreds of thousands of labeled images for classifier training. In Section 3.5 we found that self-supervised pretraining on 1M images from different domains results in representations with very different capabilities, and that simple methods for combining different datasets do not lead to large gains. In Section 3.5 we showed that image resolution is critical for contrastive learning and, more broadly, that some image corruptions can degrade a self-supervised representation to the point of unusability while others have almost no impact. Finally, in Section 3.5 we found that supervised pretraining retains a substantial edge when it comes to fine-grained classification. These experiments highlight several areas where further research is needed to improve current SSL algorithms, most of which were not evident from traditional evaluation protocols, i.e. top-1 accuracy on ImageNet.

Limitations. We mainly perform experiments using one self-supervised method. We focus on SimCLR because it reflects the essence of state-of-the-art contrastive learning methods without introducing additional architectural complexities. While our MoCo and BYOL experiments are not much different from SimCLR, it is important to validate our results on other self-supervised methods. It would also be interesting to explore alternative backbone architectures [17, 6], though after controlling for training settings, ResNet-50 remains competitive with newer architectures [55, 54]. We study only classification tasks, so additional work is also required to understand how these results translate to segmentation [53] or detection [63, 28]. Finally, we only consider datasets up to roughly ImageNet scale. We

believe this is the most practical setting for most use cases, but it is possible that some patterns may be different for significantly larger datasets and models [22, 23].

3.7 Acknowledgements

We thank Mason McGill for detailed feedback, and Grant Van Horn, Christine Kaeser-Chen, Yin Cui, Sergey Ioffe, Pietro Perona, and the rest of the Perona Lab for insightful discussions. This work was supported by the Caltech Resnick Sustainability Institute, an NSF Graduate Research Fellowship (grant number DGE1745301), and the Pioneer Centre for AI (DNRF grant number P1).

References

- [1] Alessandro Achille, Matteo Rovere, and Stefano Soatto. “Critical learning periods in deep neural networks”. In: *ICLR*. 2019.
- [2] Kumar Ayush et al. “Geography-aware self-supervised learning”. In: *ICCV*. 2021.
- [3] Sara Beery, Grant Van Horn, and Pietro Perona. “Recognition in terra incognita”. In: *ECCV*. 2018.
- [4] Tiffany Tianhui Cai et al. “Are all negatives created equal in contrastive instance discrimination?” In: *arXiv:2010.06682* (2020).
- [5] Yue Cao et al. “Parametric instance classification for unsupervised visual feature learning”. In: *NeurIPS*. 2020.
- [6] Mathilde Caron et al. “Emerging properties in self-supervised vision transformers”. In: *ICCV*. 2021.
- [7] Mathilde Caron et al. “Unsupervised learning of visual features by contrasting cluster assignments”. In: *NeurIPS*. 2020.
- [8] Ting Chen, Calvin Luo, and Lala Li. “Intriguing properties of contrastive losses”. In: *NeurIPS*. 2021.
- [9] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *ICML*. 2020.
- [10] Ting Chen et al. “Big self-supervised models are strong semi-supervised learners”. In: *NeurIPS*. 2020.
- [11] Xinlei Chen and Kaiming He. “Exploring simple siamese representation learning”. In: *CVPR*. 2021.
- [12] Xinlei Chen et al. “Improved baselines with momentum contrastive learning”. In: *arXiv:2003.04297* (2020).

- [13] Elijah Cole, Benjamin Deneu, Titouan Lorieul, Maximilien Servajean, Christophe Botella, Dan Morris, Nebojsa Jojic, Pierre Bonnet, and Alexis Joly. “The geolifeclef 2020 dataset”. In: *arXiv preprint arXiv:2004.04192* (2020). doi: 10.48550/arXiv.2004.04192.
- [14] Yin Cui et al. “Measuring dataset granularity”. In: *arXiv:1912.10154* (2019).
- [15] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *CVPR*. 2009.
- [16] Carl Doersch, Abhinav Gupta, and Alexei A Efros. “Unsupervised visual representation learning by context prediction”. In: *ICCV*. 2015.
- [17] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *ICLR*. 2021.
- [18] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. “How Well Do Self-Supervised Models Transfer?” In: *CVPR*. 2021.
- [19] Robert Geirhos et al. “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness”. In: *ICLR*. 2019.
- [20] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. “Unsupervised representation learning by predicting image rotations”. In: *ICLR*. 2018.
- [21] Priya Goyal et al. “Scaling and benchmarking self-supervised visual representation learning”. In: *ICCV*. 2019.
- [22] Priya Goyal et al. “Self-supervised pretraining of visual features in the wild”. In: *arXiv preprint arXiv:2103.01988* (2021).
- [23] Priya Goyal et al. “Vision models are more robust and fair when pretrained on uncurated images without supervision”. In: *arXiv preprint arXiv:2202.08360* (2022).
- [24] Jean-Bastien Grill et al. “Bootstrap Your Own Latent-A New Approach to Self-Supervised Learning”. In: *NeurIPS* (2020).
- [25] Kaiming He et al. “Deep residual learning for image recognition”. In: *CVPR*. 2016.
- [26] Kaiming He et al. “Momentum contrast for unsupervised visual representation learning”. In: *CVPR*. 2020.
- [27] Olivier J Hénaff et al. “Data-Efficient Image Recognition with Contrastive Predictive Coding”. In: *arXiv:1905.09272* (2019).
- [28] Olivier J Hénaff et al. “Efficient visual pretraining with contrastive detection”. In: *ICCV*. 2021.
- [29] Katherine L Hermann, Ting Chen, and Simon Kornblith. “The origins and prevalence of texture bias in convolutional neural networks”. In: *NeurIPS*. 2020.

- [30] Jian Kang et al. “Deep unsupervised embedding for remotely sensed images based on spatially augmented momentum contrast”. In: *Transactions on Geoscience and Remote Sensing* (2020).
- [31] Prannay Khosla et al. “Supervised contrastive learning”. In: *NeurIPS*. 2020.
- [32] Klemen Kotar et al. “Contrasting Contrastive Self-Supervised Representation Learning Pipelines”. In: *ICCV*. 2021.
- [33] Hsin-Ying Lee et al. “Unsupervised representation learning by sorting sequences”. In: *ICCV*. 2017.
- [34] Dhruv Mahajan et al. “Exploring the limits of weakly supervised pretraining”. In: *ECCV*. 2018.
- [35] Mehdi Noroozi and Paolo Favaro. “Unsupervised learning of visual representations by solving jigsaw puzzles”. In: *ECCV*. 2016.
- [36] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv:1807.03748* (2019).
- [37] Deepak Pathak et al. “Context encoders: Feature learning by inpainting”. In: *CVPR*. 2016.
- [38] Senthil Purushwalkam and Abhinav Gupta. “Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases”. In: *NeurIPS*. 2020.
- [39] Nils Rethmeier and Isabelle Augenstein. “Long-Tail Zero and Few-Shot Learning via Contrastive Pretraining on and for Small Data”. In: *arXiv:2010.01061* (2020).
- [40] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *IJCV* (2015).
- [41] Aaqib Saeed, David Grangier, and Neil Zeghidour. “Contrastive learning of general-purpose audio representations”. In: *ICASSP*. 2021.
- [42] Mert Bulent Sariyildiz et al. “Concept generalization in visual representation learning”. In: *ICCV*. 2021.
- [43] Hari Sowrirajan et al. “MoCo pretraining improves representation and transferability of chest X-ray models”. In: *Medical Imaging with Deep Learning*. 2021.
- [44] Chen Sun et al. “Revisiting unreasonable effectiveness of data in deep learning era”. In: *ICCV*. 2017.
- [45] Siyi Tang et al. “Data valuation for medical imaging using Shapley value and application to a large-scale chest X-ray dataset”. In: *Scientific reports* (2021).
- [46] Bart Thomee et al. “YFCC100M: The new data in multimedia research”. In: *Communications of the ACM* (2016).

- [47] Yonglong Tian, Olivier J Henaff, and Aaron van den Oord. “Divide and Contrast: Self-supervised Learning from Uncurated Data”. In: *ICCV*. 2021.
- [48] Yonglong Tian, Dilip Krishnan, and Phillip Isola. “Contrastive multiview coding”. In: *ECCV*. 2020.
- [49] Hugo Touvron et al. “Fixing the train-test resolution discrepancy”. In: *NeurIPS*. 2019.
- [50] Wouter Van Gansbeke et al. “Revisiting Contrastive Methods for Unsupervised Learning of Visual Representations”. In: *NeurIPS*. 2021.
- [51] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisín Mac Aodha. “Benchmarking representation learning for natural world image collections”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12884–12893. DOI: 10.48550/arXiv.2103.16483.
- [52] Ali Varamesh et al. “Self-Supervised Ranking for Representation Learning”. In: *arXiv:2010.07258* (2020).
- [53] Wenguan Wang et al. “Exploring cross-image pixel contrast for semantic segmentation”. In: *ICCV*. 2021.
- [54] Ross Wightman, Hugo Touvron, and Hervé Jégou. “ResNet strikes back: An improved training procedure in timm”. In: *arXiv:2110.00476* (2021).
- [55] Tete Xiao et al. “Early convolutions help transformers see better”. In: *NeurIPS*. 2021.
- [56] Tete Xiao et al. “What Should Not Be Contrastive in Contrastive Learning”. In: *ICLR*. 2020.
- [57] Xingyi Yang et al. “Transfer Learning or Self-supervised Learning? A Tale of Two Pretraining Paradigms”. In: *arXiv:2007.04234* (2020).
- [58] Richard Zhang, Phillip Isola, and Alexei A Efros. “Colorful image colorization”. In: *ECCV*. 2016.
- [59] Richard Zhang, Phillip Isola, and Alexei A Efros. “Split-brain autoencoders: Unsupervised learning by cross-channel prediction”. In: *CVPR*. 2017.
- [60] Nanxuan Zhao et al. “What makes instance discrimination good for transfer learning?” In: *ICLR*. 2021.
- [61] Bolei Zhou et al. “Places: A 10 million Image Database for Scene Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).
- [62] Benjin Zhu et al. “EqCo: Equivalent Rules for Self-supervised Contrastive Learning”. In: *arXiv:2010.01929* (2020).
- [63] Barret Zoph et al. “Rethinking pre-training and self-training”. In: *NeurIPS*. 2020.

ON LABEL GRANULARITY AND OBJECT LOCALIZATION

- [1] Elijah Cole, Kimberly Wilber, Grant Van Horn, Xuan Yang, Marco Fornoni, Pietro Perona, Serge Belongie, Andrew Howard, and Oisín Mac Aodha. “On Label Granularity and Object Localization”. In: *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*. Springer. 2022, pp. 604–620. doi: 10.48550/arXiv.2207.10225.

4.1 Abstract

Weakly supervised object localization (WSOL) aims to learn representations that encode object location using only image-level category labels. However, many objects can be labeled at different levels of granularity. Is it an animal, a bird, or a great horned owl? Which image-level labels should we use? In this paper we study the role of label granularity in WSOL. To facilitate this investigation we introduce iNatLoc500, a new large-scale fine-grained benchmark dataset for WSOL. Surprisingly, we find that choosing the right training label granularity provides a much larger performance boost than choosing the best WSOL algorithm. We also show that changing the label granularity can significantly improve data efficiency.

4.2 Introduction

For many problems in computer vision, it is not enough to know *what* is in an image, we also need to know *where* it is. Examples can be found in many domains, including ecological conservation [19], autonomous driving [55], and medical image analysis [30]. The most popular paradigm for locating objects in images is object *detection*, which aims to predict a bounding box for every instance of every category of interest. Object *localization* is special case of detection where each image is assumed to contain exactly one object instance of interest, and the category of that object is known.

Standard approaches to object detection and localization require bounding boxes for training, which are expensive to collect at scale [37]. Weakly supervised object localization (WSOL) methods aim to sidestep this obstacle by learning to localize objects using only image-level labels at training time. The potential reduction in

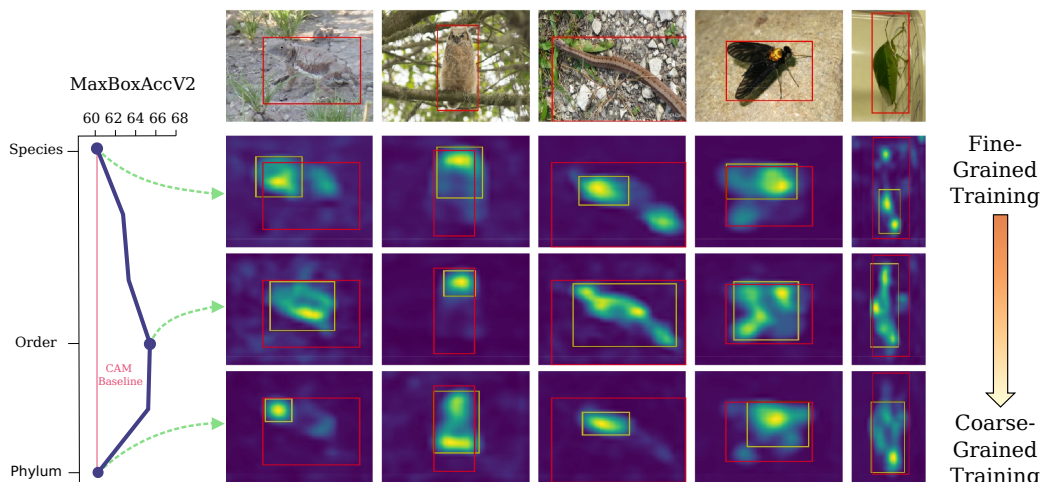


Figure 4.1: **Label granularity is a critical but understudied factor in weakly supervised object localization (WSOL).** We show five hand-picked examples from our iNatLoc500 dataset. Below each image we show class activation maps (CAMs) [63] derived from training a classifier at different granularity levels, with ground truth bounding boxes (red) and WSOL-based bounding boxes (yellow) superimposed. Conventional training does not consider label granularity and can lead to inferior localization performance (red line). Better WSOL results can be achieved by training with coarse (i.e. "order") labels, as opposed to fine-grained (i.e. "species") ones.

annotation cost which could result from effective weakly supervised methods has stimulated significant interest in WSOL over the last few years [60].

In this paper we explore the role of label granularity in WSOL. The *granularity* of a category is the degree to which it is specific, which can vary from coarse-grained (e.g. "animal") to fine-grained (e.g. "great horned owl") [54]. When we work with benchmark datasets in computer vision, we often take the given level of label granularity for granted. However, it is usually possible to make those labels more general or more specific. It is worth asking whether the label granularity we are given is the best one to use for a certain task. Label granularity matters for WSOL because the first step in most WSOL algorithms is to train a classifier using image-level category labels. By choosing a label granularity we are choosing which training images are grouped into categories. This affects the discriminative features learned by the classifier and ultimately determines the bounding box predictions. Is it possible to improve WSOL performance by controlling label granularity?

Unfortunately, it is difficult to explore label granularity in WSOL due to the limitations of existing datasets. The field of WSOL largely relies on CUB [52] and

ImageNet [41]. CUB has a consistent label hierarchy (i.e. one that can be used to measure label granularity), but it is small ($\sim 6k$ training images) and homogeneous (only bird categories). ImageNet is large and diverse, but lacks a consistent label hierarchy (see Sec. 4.5). Furthermore, [11] recently found that many purported algorithmic advances in WSOL over the last few years (which were based on these two datasets) perform no better than baselines when they are evaluated fairly. This calls for the development of more diverse and challenging benchmarks for WSOL.

Our primary contributions are as follows:

1. We explore the effect of label granularity on WSOL, and show that training at coarser levels of granularity leads to surprisingly large performance gains across many different WSOL methods compared to conventional training e.g. +5.1 MaxBoxAccV2 for CAM and +6.6 MaxBoxAccV2 for CutMix (see Fig. 4.3)
2. We demonstrate that training on coarse labels is more data efficient than conventional training. For instance, training at a coarser level achieves the same performance as conventional CAM with $\sim 15\times$ fewer labels (see Fig. 4.4).
3. We introduce the iNaturalist Localization 500 (iNatLoc500) dataset, which consists of 138k images for weakly supervised training and 25k images with manually verified bounding boxes for validation and testing. iNatLoc500 covers 500 diverse categories with a consistent hierarchical label space.

4.3 Related Work

Here we primarily focus on literature related to WSOL. See [60] for a broader overview of related techniques such as weakly supervised object detection [5, 6, 47].

Weakly Supervised Object Localization. The goal of WSOL is to determine the location of single objects in images using only image-level labels at training time. Early attempts at WSOL explored a variety of different approaches, such as adapting boosting-based methods [34], framing the problem as multiple instance learning [18, 20], and applying latent deformable part-based formulations [36].

Some foundational work in deep learning investigated the degree to which object localization comes "for free" when training supervised CNNs for image classification tasks [59, 35, 63]. In particular, the Class Activation Mapping (CAM) method of [63]

showed that CNNs can capture some object location information even when they are trained using only image-level class labels. This inspired a large body of work [61, 45, 62, 10, 26, 27] that attempted to address some of the shortcomings of CAM, e.g. by preventing the underlying model from only focusing on the most discriminative parts of an object [58] or increasing the spatial resolution of its outputs [43, 9].

Recently, [11] showed that when state-of-the-art WSOL methods are fairly compared (e.g. by controlling for the backbone architecture and operating thresholds), they are no better than the standard CAM [63] baseline. Thus, despite its simplicity, CAM is still a surprisingly effective baseline for WSOL. Subsequent work has explored further techniques for improving CAM-based methods [1, 28] and alternative approaches for estimating model coefficients [24].

Task Granularity and Localization. Despite the considerable interest in WSOL in recent years, many open questions remain. Examples include the effect of label granularity (e.g. coarse-grained labels like "bird" vs. fine-grained labels indicating the specific species of bird) and the effect of training set size. In the context of supervised object detection, [51] showed that *coarsening* category labels at training time can improve the localization performance of *object detectors*. It is unclear if the same phenomenon holds for WSOL. [53] explored the impact of label granularity for object detection on the OpenImages [29] dataset and observed a small performance improvement when training on finer labels. In the semi-supervised detection setting, [57] trained object detectors on OpenImages and ImageNet using both coarse-grained bounding box annotations and fine-grained image-level labels. [49] also explored semi-supervised detection with an approach that generates object proposals across multiple hierarchical levels. Unlike our work, these detection-based methods require bounding box information at training time. In addition, the label hierarchies for datasets like ImageNet and OpenImages are not necessarily good proxies for visual similarity or concept granularity (see Sec. 4.5).

For WSOL, [27] showed that aggregating class attribution maps at coarser hierarchical levels (e.g. "dog") results in more spatial coverage of the objects of interest, whereas maps for finer-scale concepts (e.g. "Afghan hound") only focus on subparts of the object. However, their analysis does not explore the impact of training at different granularity levels. It is also worth noting that their aggregation method only improves performance on CUB. Regarding data quantity, [11] studied the number of supervised examples used to tune the hyperparameters of CAM, but did not consider the impact of the number of examples used to train the image classifier.

Though not directly related to our work, we note that label granularity has been studied in many contexts other than object localization, including action recognition [44], knowledge tracing [13], animal face alignment [25], and fashion attribute recognition [21]. In the context of image classification, prior work has tackled topics like analyzing the emergence of hierarchical structure in trained classifiers [4], identifying patterns in visual concept generalization [42], and training finer-grained image classifiers using only coarse-grained labels [46, 40, 56, 48].

Datasets for Object Localization. Early work in WSOL (e.g. [34, 18, 32]) focused on relatively simple and small-scale datasets such as Caltech4 [17], the Weizmann Horse Database [7], or subsets of PASCAL-VOC [16]. With the rise of deep learning-based methods, CUB [52] and ImageNet [15, 41] became the standard benchmarks for this task. CUB [52] consists of images of 200 different categories of birds, where each image contains a single bird instance. ImageNet [15, 41] contains 1000 diverse categories and has significantly more images than CUB (>1M compared to ~6k). [11] proposed OpenImages30k, a 100-category localization-focused subset of the OpenImages V5 dataset [29]. An overview of these datasets is presented in Table 4.1.

These existing datasets are valuable, but they have shortcomings. CUB is small and homogeneous (only birds). OpenImages30k, as presented in [11], is not actually evaluated as a bounding box localization task. It is instead a per-pixel foreground object segmentation task where the ground truth also features some "ignore" regions that are excluded from the evaluation. Finally, while both OpenImages30k and ImageNet have label hierarchies, they do not reflect concept granularity in a consistent way. As a result, it is difficult to use them to better understand the relationship between concept granularity and localization. We discuss these issues in greater detail in Sec. 4.5. To address these shortcomings we introduce iNatLoc500, a new WSOL dataset composed of images from 500 fine-grained visual categories and equipped with a consistent label hierarchy.

4.4 Background

Weakly Supervised Object Localization (WSOL)

We begin by formalizing the WSOL setting. Let D_w be a set of *weakly labeled* images, i.e. $D_w = \{(x_i, y_i)\}_{i=1}^{N_w}$ where $x_i \in \mathbb{R}^{H \times W \times 3}$ is an image and $y_i \in \{1, \dots, C\}$ is an image-level label corresponding to one of C categories. Let D_f be a set of *fully labeled* images, i.e. $D_f = \{(x_i, y_i, \mathbf{b}_i)\}_{i=1}^{N_f}$ where x_i and y_i are defined as before

Table 4.1: Comparison of datasets for WSOL. The vast majority of WSOL papers use only CUB and ImageNet. The OpenImages30k dataset was introduced by [11], which also defines the splits we use for CUB and ImageNet. For each split we provide the minimum, maximum, and mean number of images per category, along with the total number of images in the split. Means are rounded to the nearest integer. The properties of these four datasets are discussed in detail in Sec. 4.5.

Dataset	# Cat.	train-weaksup (D_w)				train-fullsup (D_f)				test (D_{test})			
		Min	Max	Mean	Total	Min	Max	Mean	Total	Min	Max	Mean	Total
CUB [52]	200	29	30	30	6k	3	6	5	1k	11	30	29	5.8k
ImageNet [15]	1000	732	1300	1281	1.28M	10	10	10	10k	10	10	10	10k
OpenImages30k [2, 11]	100	230	300	298	30k	25	25	25	2.5k	50	50	50	5k
iNatLoc500	500	149	307	276	138k	25	25	25	12.5k	25	25	25	12.5k

and $\mathbf{b}_i \in \mathbb{R}^4$ is a bounding box for an instance of category y_i . In practice $N_w \gg N_f$. WSOL approaches typically comprise three steps:

(1) Train. Use D_w to train an image classifier $h_\theta : \mathbb{R}^{H \times W \times 3} \rightarrow [0, 1]^C$ by solving

$$\hat{\theta}(D_w) = \operatorname{argmin}_\theta \frac{1}{|D_w|} \sum_{(x_i, y_i) \in D_w} \mathcal{L}(h_\theta(x_i), y_i),$$

where \mathcal{L} is some training loss and θ represents the parameters of h . Different WSOL methods are primarily distinguished by the loss functions and training protocols they use to train h .

(2) Localize. For each $(x_i, y_i, \mathbf{b}_i) \in D_f$, predict a bounding box

$$\hat{\mathbf{b}}_i = g(x_i, y_i | h_{\hat{\theta}(D_w)})$$

according to some procedure $g : \mathbb{R}^{H \times W \times 3} \times \{1, \dots, C\} \rightarrow \mathbb{R}^4$. Typically g is a simple sequence of image processing operations applied to the feature maps of the trained classifier $h_{\hat{\theta}(D_w)}$.

(3) Evaluate. Let E denote a suitable WSOL error metric which compares the predicted boxes $\{\hat{\mathbf{b}}_i\}_{i=1}^{N_f}$ against the ground-truth boxes $\{\mathbf{b}_i\}_{i=1}^{N_f}$. Use the validation error $E(D_f | D_w)$ for model selection and hyperparameter tuning and then use a held-out test set D_{test} (which is fully labeled like D_f) to measure test error $E(D_{\text{test}} | D_w)$. See [11] for a discussion of WSOL performance metrics.

The role of low-shot supervised localization.

Without the fully labeled images D_f , the WSOL problem becomes ill-posed [11]. Since WSOL therefore requires at least a small number of bounding box annotations for validation, it is natural to ask how WSOL compares to few-shot object

localization? For our purposes, we define few-shot object localization methods as those which use only D_f for training and validation. Under this definition, the few-shot methods (which use only D_f) actually require strictly *less* data than WSOL (which requires both D_w and D_f). Since WSOL and few-shot object localization are practical alternatives, it is important to consider them together as in [11].

Label Hierarchies and Label Granularity

We define a *label hierarchy* (on a label set L) to be a directed rooted tree H whose leaf nodes (i.e. nodes $v \in H$ with no children) correspond to the labels in L . Edges in H represent "is-a" relationships, so a directed edge from $u \in H$ to $v \in H$ means that v (e.g. "bird") is a kind of u (e.g. "animal"). We overload L to refer to the label set and to the corresponding set of nodes in H . Let r denote the root node of H and let $d(u, v)$ denote the number of edges on the path from $u \in H$ to $v \in H$.

Coarsening a label. Because there is a unique path from the root node r to any leaf node $\ell \in L$, we can "coarsen" the label ℓ in a well-defined way by merging it with its parent node. We define the *coarsening operator* $c_k : H \rightarrow H$, which takes any node in the label hierarchy and returns the node which is k edges closer to the root. Thus, $c_0(\ell) = \ell$, $c_1(\ell)$ is the parent of ℓ , $c_2(\ell)$ is the grandparent of ℓ , and so on, with $c_k(\ell) = r$ for all $k \geq d(r, \ell)$.

Coarsening a dataset. We can describe a general "coarsened" version of $D_w = \{(x_i, y_i)\}_{i=1}^{N_w}$ as $D_w^{\mathbf{k}} = \{(x_i, c_{k_i}(y_i))\}_{i=1}^{N_w}$ where $\mathbf{k} = (k_1, \dots, k_{|D_w|})$. If we allow the entries of \mathbf{k} to be chosen completely independently, then we can encounter problems e.g. images with multiple valid labels. To prevent these cases, we require \mathbf{k} to be chosen such that $c_{k_i}(y_i) \in H$ is not a descendant of $c_{k_j}(y_j) \in H$ for any $i, j \in \{1, \dots, N_w\}$.

Problem statement. We can now formalize our key questions: How does \mathbf{k} affect $E(D_{\text{test}}|D_w^{\mathbf{k}})$? Are there choices of \mathbf{k} such that $E(D_{\text{test}}|D_w^{\mathbf{k}}) < E(D_{\text{test}}|D_w)$?

4.5 The iNatLoc500 Dataset

In this section we introduce the iNaturalist Localization 500 (iNatLoc500) dataset, a large-scale fine-grained dataset for weakly supervised object localization. We first detail the process of building the dataset and cleaning the localization annotations. We then discuss the key properties of the dataset and highlight the advantages of iNatLoc500 compared to three WSOL datasets that are currently commonly used (CUB, ImageNet, and OpenImages30k).

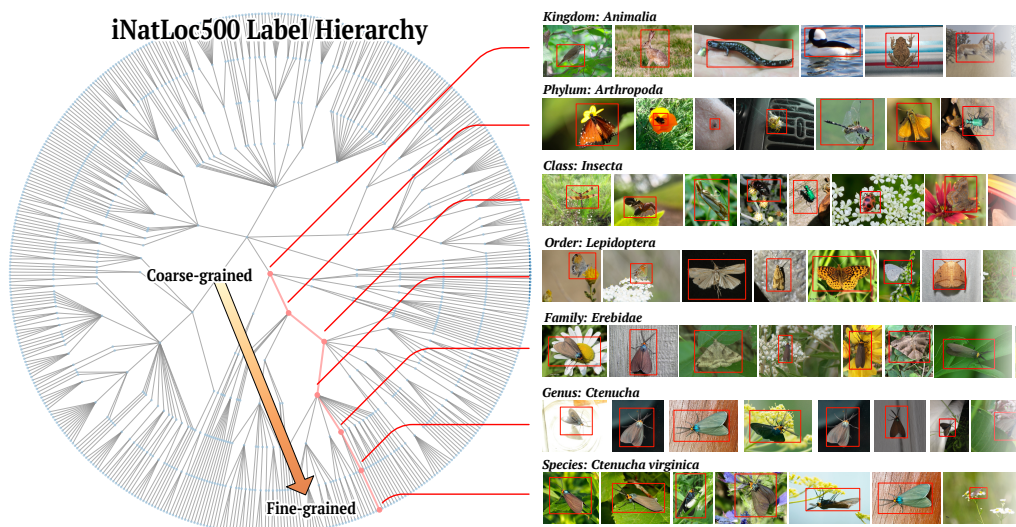


Figure 4.2: Sample images from iNatLoc500 at different levels of the label hierarchy, from coarse ("kingdom") to fine ("species"). Random images from coarse levels of the hierarchy tend to be much more varied than random images ones from finer levels.

iNatLoc500 has three parts: *train-weaksup* (D_w), *train-fullsup* (D_f), and *test* (D_{test}). Each image in the weakly supervised training set (D_w) has one image-level category label. Each image in the fully supervised validation set (D_f) and test set (D_{test}) has one image-level category label *and* one bounding box annotation. All bounding boxes have been manually validated. Split statistics are presented in Table 4.1 and sample images from the dataset can be found in Fig. 4.2. The dataset is publicly available.¹

Dataset Construction

The iNatLoc500 dataset is derived from two existing datasets: iNat17 [51] and iNat21 [50]. Both datasets contain images of plants and animals collected by the citizen science platform iNaturalist [23]. iNat21 is much larger than iNat17 (2.7M images, 10k species vs. 675k images, 5k species), but iNat17 has crowdsourced bounding box annotations. We draw from iNat21 for D_w and we draw from iNat17 for D_f and D_{test} .

Full details on the process of constructing iNatLoc500 can be found in the supplementary material, but we note two important design choices here. First, iNat17 did not collect bounding boxes for plant categories because it is often unclear how to draw bounding boxes for plants. Consequently, iNatLoc500 does not contain any

¹https://github.com/visipedia/inat_loc/

plant categories. Second, we set very high quality standards for the bounding boxes. Five computer vision researchers manually reviewed $\sim 65k$ images to ensure the quality of the bounding boxes for D_f and D_{test} , of which only 51% met our quality standards. Explicit quality criteria and examples of removed images can be found in the supplementary material.

Dataset Properties

iNatLoc500 is fine-grained, large-scale, and visually diverse. Moreover, iNatLoc500 has a consistent label hierarchy which serves as a reliable proxy for label granularity. We now discuss the importance of each of these properties and contrast iNatLoc500 with existing WSOL datasets.

Fine-grained categories. Each category in iNatLoc500 corresponds to a different species, and the differences between species can be so subtle as to require expert-level knowledge [51]. While there are challenging images in ImageNet and OpenImages30k, most of the categories are coarse-grained i.e. relatively few pairs of categories are highly visually similar. For instance, the reptile categories in OpenImages30k (lizard, snake, frog, crocodile) are typically easy to distinguish. In iNatLoc-500 there are 107 reptile species, some of which are highly similar (e.g. Chihuahuan spotted whiptail vs. Common spotted whiptail).

Consistent label hierarchy. The label hierarchy for iNatLoc500 consists of the following seven tiers, ordered from coarsest to finest: kingdom, phylum, class, order, family, genus, and species. All of the species in iNatLoc500 are animals, so the "kingdom" tier only has one node (*Animalia*), which is the root node of the label hierarchy. Every species lies at the same distance from the root. The iNatLoc500 label hierarchy is *consistent* in the sense that all nodes at a given level of the hierarchy correspond to concepts with similar levels of specificity. This means that depth in the label hierarchy measures label granularity. The label hierarchy for CUB is also consistent. However, the taxonomies that underlie ImageNet and OpenImages30k are considerably more arbitrary. For instance, in OpenImages30k some categories are far from the root of the label hierarchy, e.g.

entity/vehicle/land_vehicle/car/limousine
entity/animal/mammal/carnivore/fox

while others are close to the root, e.g.

entity/bicycle_wheel
entity/human_ear

despite the fact that there is no obvious difference in concept specificity.

Unambiguous label semantics. The categories in iNatLoc500 are well-defined in the sense that (for most species) there is little room for debate about what "counts" as an instance of that species. While the distinctions between species can be quite subtle, each species is a well-defined category. CUB shares this advantage for the most part, but ImageNet and OpenImages30k do not. For instance, OpenImages30k contains the categories `wine` and `bottle`. To which category does a bottle of wine belong? (In fact, we find bottles of wine in both categories.) ImageNet is known to have similar issues with ambiguous and overlapping category definitions [3].

Visual diversity. Like ImageNet and OpenImages30k, iNatLoc500 has a category set which exhibits a high degree of visual diversity. CUB is much more homogeneous, consisting of only birds. Combined with its consistent label hierarchy, the visual diversity of iNatLoc500 enables future work on e.g. how localization ability generalizes across categories as a function of taxonomic distance.

Large scale. iNatLoc500 is a large-scale dataset, both in terms of the number of categories and the number of training images. CUB and OpenImages30k are considerably smaller on both counts. Large training sets are valuable because they simplify supervised learning. Large training sets also enable research on self-supervised representation learning, which has received little attention thus far in WSOL. We provide a summary of the key dataset statistics in Table 4.1.

4.6 Experiments

In this section we present WSOL results on iNatLoc500 as well as existing benchmark datasets. We also consider few-shot learning baselines based on segmentation and detection architectures. Finally, we use the unique properties of iNatLoc500 to study how label granularity affects localization performance and data efficiency. A summary of the different WSOL datasets can be found in Table 4.1.

Implementation Details

Performance metrics. All WSOL performance numbers in this paper are `MaxBoxAccV2`, which is defined in [11]. The only exceptions are the results for OpenImages30k in Table 4.2, which are given in `PxAP` as defined in [11].

Fixed-granularity training. In Sec. 4.6 we probe the effect of granularity on WSOL by training on "coarsened" versions of D_w . In the notation of Sec. 4.4, these can be written $D_w^{k \cdot \mathbf{1}}$ for $k = 1, 2, \dots$, where $\mathbf{1}$ denotes the "all ones" vector. This corresponds to merging all leaves with their parent k times. We then run the entire WSOL pipeline from scratch to compute $E(D_{\text{test}} | D_w^{k \cdot \mathbf{1}})$ for each k . To the best of our knowledge this is compatible with all existing WSOL methods.

Fixed-granularity CAM aggregation. We also consider a second method for using label hierarchy information to improve WSOL, inspired by [27]. Just like traditional CAM, the first step is to train an image classifier using the standard (most fine-grained) label set. However, instead of returning only the CAM for the species labeled in the input image, we return a CAM for each species in the same genus / family / ... / phylum and average them. This "aggregated" CAM is then evaluated as normal. We abbreviate this method as CAM-Agg.

Hyperparameter search for WSOL methods. Each time we train a WSOL method we re-tune the learning rate over the set $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ and choose the one that leads to the best MaxBoxAccV2 performance on the fully supervised validation set D_f . We then report the MaxBoxAccV2 performance for the selected model on D_{test} . We leave all other hyperparameters fixed. Full training details can be found in the supplementary material.

Non-WSOL methods. We provide results for the baselines proposed in [11] (Center, FSL-Seg), as well as a new few-shot detection baseline (FSL-Det). "Center" is a naive baseline that simply assumes a centered Gaussian activation map for all images. "FSL-Seg" is a supervised baseline that is trained on the D_f split of each dataset. The architecture is based on models for saliency mask prediction [33]. Finally, we introduce "FSL-Det", a few-shot detection baseline for WSOL that is also trained on D_f . It uses Faster-RCNN [39] with the same backbone as other methods (i.e. ImageNet-pretrained ResNet-50 [22]). Full implementation details can be found in the supplementary material.

Baseline Results

We follow [11] and evaluate six recent WSOL methods and two non-WSOL methods (Center and FSL-Seg) on iNatLoc500. The results can be found in Table 4.2. We focus our observations on ImageNet, CUB, and iNatLoc500 since OpenImages30k is evaluated using a different task and evaluation metric. We first note that our findings on iNatLoc500 reinforce the main results from [11], namely that (a) none

Table 4.2: Comparison of WSOL methods. Numbers are `MaxBoxAccV2` for ImageNet, CUB, and iNatLoc500 and `PxAP` for OpenImages30k. All results use an ImageNet-pretrained ResNet-50 [22] backbone with an input resolution of 224x224. WSOL numbers for ImageNet, CUB, and OpenImages30k are the updated results from [12]. WSOL numbers for iNatLoc500 are our own, as are the numbers for the baselines (Center, FSL-Seg, FSL-Det). FSL baselines use 10 images / class for ImageNet, 5 images / class for CUB, 25 images / class for OpenImages30k, and 25 images / class for iNatLoc500. We do not report FSL-Det for OpenImages30k because the evaluation protocol for that dataset requires segmentation masks.

Method	ImageNet	CUB	OpenImages30k	iNatLoc500
CAM [63]	63.7	63.0	58.5	60.2
HaS [45]	63.4	64.7	55.9	60.0
ACoL [61]	62.3	66.5	57.3	55.3
SPG [62]	63.3	60.4	56.7	60.7
ADL [10]	63.7	58.4	55.2	58.9
CutMix [58]	63.3	62.8	57.7	60.1
Center	53.4	56.8	46.0	42.8
FSL-Seg	68.7	89.4	75.2	78.6
FSL-Det	70.4	95.4	-	83.6

of the WSOL methods performs substantially better than CAM and (b) FSL-Seg significantly outperforms all WSOL methods. Second, if we consider the performance gap between CAM and the Center baseline, we see that simple centered boxes are not as successful on iNatLoc500 (-17.2 `MaxBoxAccV2`) as they are on CUB (-6.2 `MaxBoxAccV2`) and ImageNet (-10.3 `MaxBoxAccV2`). This indicates that iNatLoc500 is a more challenging dataset for benchmarking WSOL. Finally, we provide results for our few-shot detection baseline (FSL-Det). For ImageNet, CUB, and iNatLoc500 we find that FSL-Det is a stronger baseline than FSL-Seg. Like FSL-Seg, FSL-Det directly trains on the boxes in D_f , whereas the WSOL methods only use those boxes to tune their hyperparameters. However, FSL-Det sets a new ceiling for localization performance on these datasets, indicating that current WSOL methods have considerable room for improvement.

Label Granularity and Localization Performance

iNatLoc500 is equipped with a consistent label hierarchy which allows us to directly study the relationship between label granularity and localization performance. The traditional approach to WSOL on iNatLoc500 would begin by training a classifier on the *species-level* labels, i.e. the finest level in the label hierarchy. However, our hypothesis is that training at the most fine-grained level may not lead to the

best localization performance. To study this, we use the fixed-granularity training method discussed in Sec. 4.6. In particular, we "re-label" D_w at each level of the label hierarchy using successively coarser categories. We then use each of these re-labeled datasets to train and evaluate different WSOL methods. The results in Fig. 4.3(left) show that coarsening the labels of D_w can significantly boost WSOL performance (e.g. up to +5.1 MaxBoxAccV2 for CAM). The numerical values plotted in Fig. 4.3(left) can be found in the supplementary materials. Note that it would be difficult to draw similar conclusions by studying ImageNet or OpenImages30k because their label hierarchies do not measure how fine-grained different categories are; see Sec. 4.5 for a discussion. Our conceptually simple coarsening approach results in large performance improvements across five different WSOL methods, without any modifications to the model architectures or training losses.

Coarse training beyond iNatLoc500. Fig. 4.3(left) shows that coarse training significantly improves WSOL performance on iNatLoc500. We study the effect of coarse training on FGVC-Aircraft [31], CUB [52], and ImageNet [15] in the supplementary material. As expected, FGVC-Aircraft and CUB (which have consistent label hierarchies) both benefit from coarse training while ImageNet (which lacks a consistent label hierarchy) does not.

Localization performance vs. classification performance. In Fig. 4.3(right) we show the image classification performance for each WSOL method in Fig. 4.3(left) at each granularity level. We see that classification performance and WSOL performance are not necessarily correlated. WSOL performance increases before decreasing at the coarsest level of granularity. Classification performance increases with label coarsening, even at the coarsest level of granularity.

An alternative method for incorporating label granularity. We also present the performance of CAM-Agg, an alternative method for incorporating granularity information in WSOL (see Sec. 4.6). In our experiments, CAM-Agg underperforms vanilla CAM at every granularity level. As a point of comparison, [27] finds that CAM-Agg is better than CAM for CUB but worse than CAM for ImageNet. Our findings suggest that training the model with coarse categories leads to much better localization performance when compared to aggregating the localization outputs for multiple similar fine-grained categories.

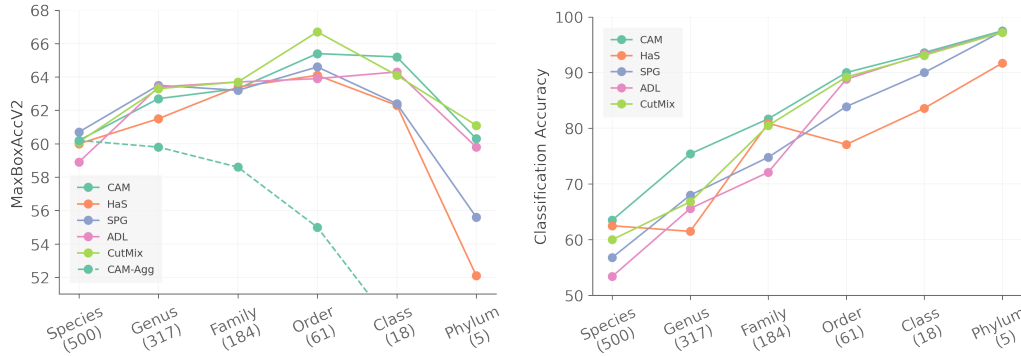


Figure 4.3: Effect of label granularity of D_w on WSOL performance (left) and classification accuracy (right) for iNatLoc500. The number of categories at each tier is given in parentheses. **(Left)** Localization performance suffers when the category labels are either too fine (e.g. Species) or too coarse (e.g. Phylum). The results on the very left of the plot are the same as those in Table 4.2. Note that ACoL is excluded due to poor performance — we suspect it requires more epochs of training than the standard protocol allows for iNatLoc500. We also show results for CAM-Agg (Sec. 4.6), an alternative method for aggregating hierarchy information in WSOL. **(Right)** Each WSOL method trains the image classifier in a different way, but classification accuracy generally increases as the labels become more coarse. Naturally it is easier to distinguish between coarser categories, but it is interesting to note that classification performance is excellent at the phylum level, despite poor localization performance.

Label Granularity and Data Efficiency

Most WSOL work makes D_w as large as possible by default, so there has been little attention paid to how the size of D_w trades off against localization performance. In this section we analyze the performance of CAM-based WSOL as a function of the size of D_w . We are particularly interested in how label granularity interacts with data efficiency. To study this question, we first pick a granularity level and generate subsampled versions of D_w by choosing, uniformly at random, 50, 100, or 200 images from each category. Note that the size of each subsampled version of D_w depends on the granularity level. For instance, if the categories are the 317 genera, then 50 images per category is $50 \times 317 = 15,850$, compared to $50 \times 61 = 3,050$ images if the categories are the 61 orders. We present WSOL results for four granularity levels in Fig. 4.4. We find that by training at a coarser level, we can obtain better performance with fewer labels. All of the square markers above the dashed line in Fig. 4.4 correspond to cases where we can achieve better performance than the standard species-level CAM approach using fewer labels. To take one example, by training at the family level we can match the performance of

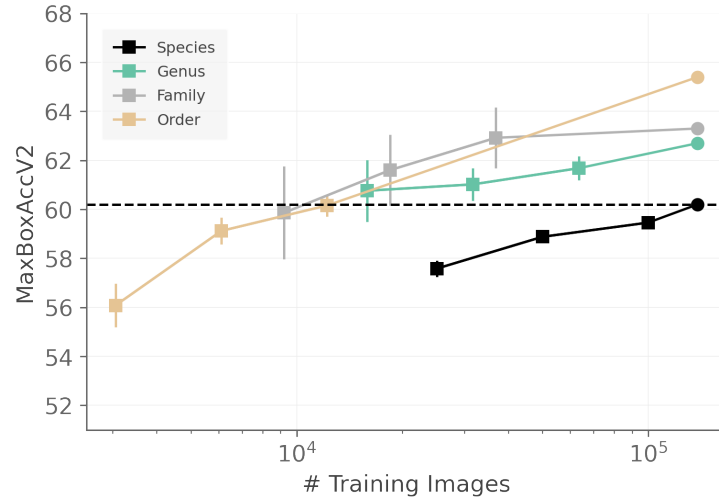


Figure 4.4: Effect of the number of training images (N_w) on CAM performance for iNatLoc500. The dashed line corresponds to the performance of species-level CAM with the entirety of D_w . Each color corresponds to a different label granularity for D_w . Circles at the right of the graph indicate performance using all of D_w . Squares represent subsampled datasets which use a fixed number of images per category: 50, 100, or 200. All squares have error bars indicating the standard deviation over 5 runs with different randomly sampled subsets of D_w .

the standard CAM approach by training with 50 images per family (9200 images), a training set reduction of $\sim 15\times$.

4.7 Discussion

Why does performance increase as we coarsen the labels? In Fig. 4.3(left) we see that five different WSOL algorithms perform better as we coarsen the labels in D_w , up until the coarsest level when performance drops. What accounts for this behavior? Our analysis of CAM in Fig. 4.5 provides some clues. Fig. 4.5(left) shows that the area of the predicted box tends to be larger than the area of the ground truth box, and that their ratio *decreases* towards unity as we coarsen the labels (black curve). That is, the predicted box size gets closer to the true box size as we coarsen the labels. This casts doubt on a common intuition (which as far as we know has not been empirically investigated before now) that WSOL methods predict smaller boxes for more fine-grained categories [27].

Why does performance drop at the coarsest level of granularity? In Fig. 4.5(left) we see that as we coarsen the labels the concentration of *activation* in the ground truth box *increases* before collapsing at the coarsest level (red curve). Fig. 4.5(right) shows that the activation maps become highly fragmented at coarser levels. Taken

together, these two findings suggest that at the coarsest level the activation maps tend to focus more on global image characteristics (e.g. land vs. water) than the properties of the foreground object. Note that these features are still useful for image classification, as is shown in Fig. 4.3(right).

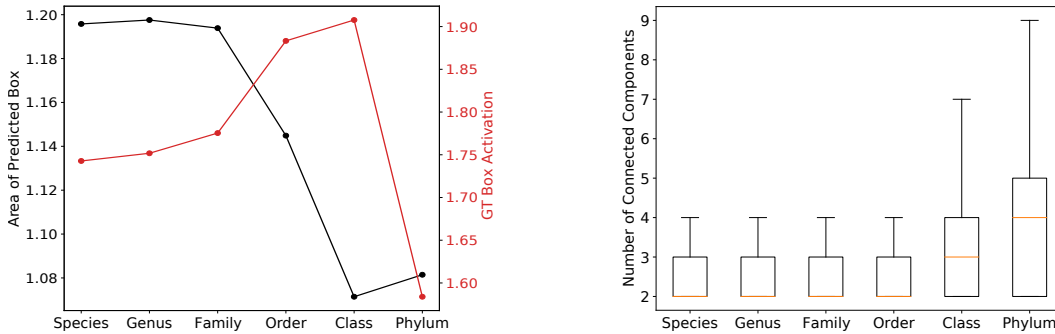


Figure 4.5: Analysis of CAM-based WSOL on the D_f split of iNatLoc500. **(Left)** *Black*: Ratio of the area of the predicted box to the area of the ground truth box. *Red*: Ratio of the activation inside the ground truth box to the activation of background pixels. Both curves show medians over the 12.5k images in D_f at each granularity level. **(Right)** Number of connected components in the binarized activation maps at each granularity level. Each box plot shows the distribution over the 12.5k images in D_f . See the supplementary material for full details on the construction of these plots.

Limitations. The iNatLoc500 dataset has several limitations. First, it contains only animal categories. These categories are highly diverse, but they are not representative of all visual domains. Second, it is possible that there are errors in the image-level labels provided by the iNaturalist community, though this is expected to be rare as each image has been labeled by multiple people [50]. Third, many real fine-grained problems have a long-tailed class distribution but, like other localization datasets, iNatLoc500 is approximately balanced (at the species level). Finally, there is a conceptual limitation in our experiments: the use of a single granularity level across the entire dataset. In fact, it is likely that different images are best treated at different granularity levels. Our work does not address this important topic which we leave for future work.

iNatLoc500 can be used to investigate numerous research agendas beyond traditional WSOL. For example, D_w was designed to be large enough for self-supervised learning, which has received surprisingly little attention in the WSOL community [8]. We are also interested in using iNatLoc500 to study whether self-supervised learning methods can be improved by using WSOL methods to select crops [38], especially in the context of fine-grained data [14]. For the object de-

tection community, the clean boxes in iNatLoc500 can (i) serve as a test set for object detectors trained on the noisy iNat17 boxes, (ii) be used to study the problem of learning multi-instance detectors from one box per image, and (iii) be used to analyze the role of label granularity in object detection. Finally, we have seen that hierarchical reasoning can significantly improve localization performance. In the future, we aim to explore methods for automatically determining the most appropriate level of coarseness required for generating representations that best encode object location.

4.8 Conclusion

We have shown that substantial improvements in WSOL performance can be achieved by modulating the granularity of the training labels, and that coarser-grained training leads to more data-efficient WSOL. We also presented iNatLoc500, a new large-scale fine-grained dataset for WSOL. Despite the gains in performance from coarse-level training, iNatLoc500 remains a challenging localization task which we hope will motivate additional progress in WSOL.

4.9 Acknowledgements

We thank the iNaturalist community for sharing images and species annotations. This work was supported by the Caltech Resnick Sustainability Institute, an NSF Graduate Research Fellowship (grant number DGE1745301), and the Pioneer Centre for AI (DNRF grant number P1).

References

- [1] Wonho Bae, Junhyug Noh, and Gunhee Kim. “Rethinking class activation mapping for weakly supervised object localization”. In: *ECCV*. 2020.
- [2] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. “Large-scale interactive object segmentation with human annotators”. In: *CVPR*. 2019.
- [3] Lucas Beyer et al. “Are we done with imagenet?” In: *arXiv:2006.07159* (2020).
- [4] Alsallakh Bilal et al. “Do convolutional neural networks learn class hierarchy?” In: *IEEE transactions on visualization and computer graphics* 24.1 (2017), pp. 152–162.
- [5] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. “Weakly supervised object detection with convex clustering”. In: *CVPR*. 2015.
- [6] Hakan Bilen and Andrea Vedaldi. “Weakly supervised deep detection networks”. In: *CVPR*. 2016.

- [7] Eran Borenstein and Shimon Ullman. “Class-specific, top-down segmentation”. In: *ECCV*. 2002.
- [8] Mathilde Caron et al. “Emerging properties in self-supervised vision transformers”. In: *ICCV*. 2021.
- [9] Aditya Chattopadhyay et al. “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks”. In: *WACV*. 2018.
- [10] Junsuk Choe and Hyunjung Shim. “Attention-based dropout layer for weakly supervised object localization”. In: *CVPR*. 2019.
- [11] Junsuk Choe et al. “Evaluating weakly supervised object localization methods right”. In: *CVPR*. 2020.
- [12] Junsuk Choe et al. “Evaluation for weakly supervised object localization: Protocol, metrics, and datasets”. In: *arXiv:2007.04178* (2020).
- [13] Youngduck Choi et al. “Ednet: A large-scale hierarchical dataset in education”. In: *International Conference on Artificial Intelligence in Education*. Springer. 2020, pp. 69–73.
- [14] Elijah Cole et al. “When Does Contrastive Visual Representation Learning Work?” In: *CVPR*. 2022.
- [15] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *CVPR*. 2009.
- [16] Mark Everingham et al. “The pascal visual object classes (voc) challenge”. In: *IJCV* (2010).
- [17] Robert Fergus, Pietro Perona, and Andrew Zisserman. “Object class recognition by unsupervised scale-invariant learning”. In: *CVPR*. 2003.
- [18] Carolina Galleguillos et al. “Weakly supervised object localization with stable segmentations”. In: *ECCV*. 2008.
- [19] Jan C van Gemert et al. “Nature conservation drones for automatic localization and counting of animals”. In: *ECCV*. 2014.
- [20] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. “Multi-fold mil training for weakly supervised object localization”. In: *CVPR*. 2014.
- [21] Sheng Guo et al. “The imaterialist fashion attribute dataset”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019.
- [22] Kaiming He et al. “Deep residual learning for image recognition”. In: *CVPR*. 2016.
- [23] *iNaturalist*. www.inaturalist.org, accessed Mar 7 2022.
- [24] Hyungsik Jung and Youngrook Oh. “Towards Better Explanations of Class Activation Mapping”. In: *ICCV*. 2021.

- [25] Muhammad Haris Khan et al. “Animalweb: A large-scale hierarchical dataset of annotated animal faces”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6939–6948.
- [26] Minsong Ki et al. “In-sample contrastive learning and consistent attention for weakly supervised object localization”. In: *ACCV*. 2020.
- [27] Jae Myung Kim et al. “Keep CALM and Improve Visual Feature Attribution”. In: *ICCV*. 2021.
- [28] Jeessoo Kim et al. “Normalization Matters in Weakly Supervised Object Localization”. In: *ICCV*. 2021.
- [29] Alina Kuznetsova et al. “The open images dataset v4”. In: *IJCV* (2020).
- [30] Zhuoling Li et al. “CLU-CNNs: Object detection for medical images”. In: *Neurocomputing* (2019).
- [31] S. Maji et al. *Fine-Grained Visual Classification of Aircraft*. Tech. rep. 2013. arXiv: 1306.5151 [cs-cv].
- [32] Minh Hoai Nguyen et al. “Weakly supervised discriminative localization and classification: a joint learning process”. In: *ICCV*. 2009.
- [33] Seong Joon Oh et al. “Exploiting saliency for object segmentation from image level labels”. In: *CVPR*. 2017.
- [34] Andreas Opelt and Axel Pinz. “Object localization with boosting and weak supervision for generic object recognition”. In: *Scandinavian Conference on Image Analysis*. 2005.
- [35] Maxime Oquab et al. “Is object localization for free?-weakly-supervised learning with convolutional neural networks”. In: *CVPR*. 2015.
- [36] Megha Pandey and Svetlana Lazebnik. “Scene recognition and weakly supervised object localization with deformable part-based models”. In: *ICCV*. 2011.
- [37] Dim P Papadopoulos et al. “Extreme clicking for efficient object annotation”. In: *ICCV*. 2017.
- [38] Xiangyu Peng et al. “Crafting better contrastive views for siamese representation learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16031–16040.
- [39] Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems*. 2015, pp. 91–99.
- [40] Joshua Robinson, Stefanie Jegelka, and Suvrit Sra. “Strength from weakness: Fast learning using weak supervision”. In: *ICML*. 2020.
- [41] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *IJCV* (2015).

- [42] Mert Bulent Sariyildiz et al. “Concept generalization in visual representation learning”. In: *ICCV*. 2021.
- [43] Ramprasaath R Selvaraju et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *ICCV*. 2017.
- [44] Dian Shao et al. “Finegym: A hierarchical video dataset for fine-grained action understanding”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2616–2625.
- [45] Krishna Kumar Singh and Yong Jae Lee. “Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization”. In: *ICCV*. 2017.
- [46] Fariborz Taherkhani et al. “A weakly supervised fine label classifier enhanced by coarse supervision”. In: *ICCV*. 2019.
- [47] Peng Tang et al. “Multiple instance detection network with online instance classifier refinement”. In: *CVPR*. 2017.
- [48] Hugo Touvron et al. “Grafit: Learning fine-grained image representations with coarse labels”. In: *ICCV*. 2021.
- [49] Jasper Uijlings, Stefan Popov, and Vittorio Ferrari. “Revisiting knowledge transfer for training object class detectors”. In: *CVPR*. 2018.
- [50] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. “Benchmarking representation learning for natural world image collections”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12884–12893. DOI: 10.48550/arXiv.2103.16483.
- [51] Grant Van Horn et al. “The iNaturalist species classification and detection dataset”. In: *CVPR*. 2018.
- [52] C. Wah et al. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.
- [53] Rui Wang, Dhruv Mahajan, and Vignesh Ramanathan. “What leads to generalization of object proposals?” In: *ECCV Workshops*. 2020.
- [54] Xiu-Shen Wei et al. “Fine-Grained Image Analysis with Deep Learning: A Survey”. In: *PAMI* (2021).
- [55] Bichen Wu et al. “Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving”. In: *CVPR Workshops*. 2017.
- [56] Yuanhong Xu et al. “Weakly Supervised Representation Learning With Coarse Labels”. In: *ICCV*. 2021.
- [57] Hao Yang, Hao Wu, and Hao Chen. “Detecting 11k classes: Large scale object detection without fine-grained bounding boxes”. In: *ICCV*. 2019.

- [58] Sangdoon Yun et al. “Cutmix: Regularization strategy to train strong classifiers with localizable features”. In: *ICCV*. 2019.
- [59] Matthew D Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *ECCV*. 2014.
- [60] Dingwen Zhang et al. “Weakly Supervised Object Localization and Detection: A Survey”. In: *PAMI* (2021).
- [61] Xiaolin Zhang et al. “Adversarial complementary learning for weakly supervised object localization”. In: *CVPR*. 2018.
- [62] Xiaolin Zhang et al. “Self-produced guidance for weakly-supervised object localization”. In: *ECCV*. 2018.
- [63] Bolei Zhou et al. “Learning deep features for discriminative localization”. In: *CVPR*. 2016.

MULTI-LABEL LEARNING FROM SINGLE POSITIVE LABELS

- [1] Elijah Cole, Oisín Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. “Multi-label learning from single positive labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 933–942. DOI: [10.48550/arXiv.2106.09708](https://doi.org/10.48550/arXiv.2106.09708).

5.1 Abstract

Predicting all applicable labels for a given image is known as multi-label classification. Compared to the standard multi-class case (where each image has only one label), it is considerably more challenging to annotate training data for multi-label classification. When the number of potential labels is large, human annotators find it difficult to mention all applicable labels for each training image. Furthermore, in some settings detection is intrinsically difficult, e.g. finding small object instances in high resolution images. As a result, multi-label training data is often plagued by false negatives. We consider the hardest version of this problem, where annotators provide only one relevant label for each image. As a result, training sets will have only one positive label per image and no confirmed negatives. We explore this special case of learning from missing labels across four different multi-label image classification datasets for both linear classifiers and end-to-end fine-tuned deep networks. We extend existing multi-label losses to this setting and propose novel variants that constrain the number of expected positive labels during training. Surprisingly, we show that in some cases it is possible to approach the performance of fully labeled classifiers despite training with significantly fewer confirmed labels.

5.2 Introduction

The majority of work in visual classification is focused on the *multi-class* setting, where each image is assumed to belong to one of L classes. However, the world is intrinsically *multi-label*: scenes contain multiple objects, CT scans reveal multiple health conditions, satellite images show multiple terrain types, etc. Unfortunately, it can be prohibitively expensive to obtain the large number of accurate multi-label annotations required to train deep neural networks [10]. Heuristics can be used to reduce the required annotation effort [34, 18], but this runs the risk of increasing

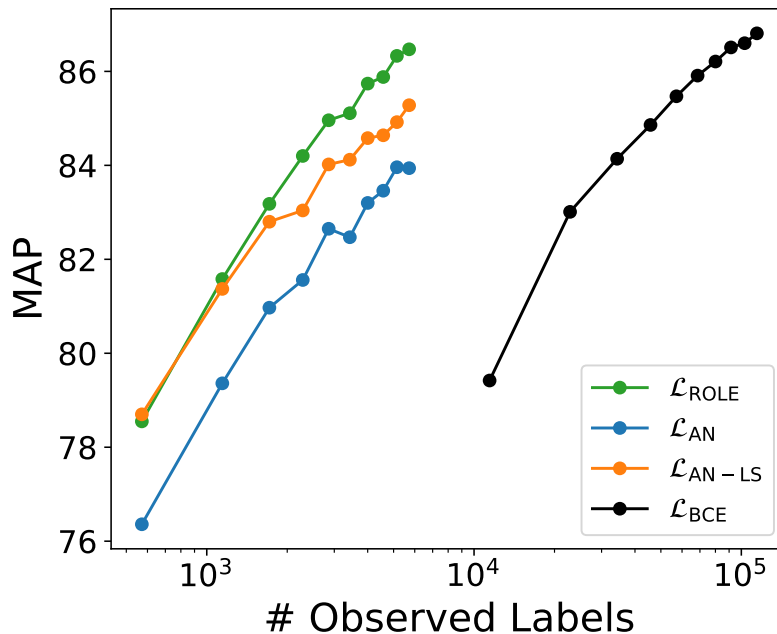


Figure 5.1: It is possible to approach the performance of full supervision (\mathcal{L}_{BCE}) using only one positive label per image. Here we show test MAP as a function of the number of training labels for PASCAL VOC 2012 [13]. Each curve is generated by randomly subsampling $m\%$ of the images from the training set for $m \in \{10, 20, \dots, 100\}$. The number of labels per image then determines the number of observed label on the horizontal axis: \mathcal{L}_{BCE} receives all 20 labels per image, while the other methods only receive one positive label per training image. Despite having a factor of 20 times fewer labels, our $\mathcal{L}_{\text{ROLE}}$ approach achieves comparable performance to the fully labeled case (\mathcal{L}_{BCE}).

error in the labels. Even without heuristics, false negatives are common because (i) rare classes are often missed by human annotators [60, 59] and (ii) detecting absence can be more difficult than detecting presence [60]. This may explain why even flagship multi-class datasets like ImageNet have been found to include images that actually belong to multiple classes [61]. Since it is generally infeasible to exhaustively annotate every image for all classes that could be present, there is a natural trade-off between *how many* images receive annotations and *how completely* each image is annotated. On one extreme, we could fully annotate images until the labeling budget is exhausted. In this paper we are interested in the other extreme, in which our dataset consists of many images, but each individual image has minimal supervision.

We explore the problem of *single positive multi-label learning*, where only a single

positive label (and no other true positives or true negative labels) is observed for each training image. This is a worthwhile problem for at least three reasons: First, an effective method for this setting could allow for significantly reduced annotations costs for future datasets. Second, multi-class datasets may have images that actually contain more than one class. For instance, the iNaturalist dataset has many images of insects on plants, but only one is annotated as the true class [53]. Finally, it is of scientific interest to understand how well multi-label classifiers can be made to perform at the minimal limit of supervision. This is particularly interesting because many standard approaches for dealing with missing labels, e.g. learning positive label correlations [6], performing label matrix completion [4], or learning to infer missing labels [54] break down in the single positive only setting.

We direct attention to this important but underexplored variant of multi-label learning. Our experiments show that training with a single positive label per image allows us to drastically reduce the amount of supervision required to train multi-label image classifiers, while only incurring a tolerable drop in classification performance (see Figure 5.1). We make three contributions: (i) A unified presentation and extension of existing multi-label approaches to the single positive multi-label learning setting; (ii) a novel single positive multi-label loss that estimates missing labels during training; and (iii) a detailed experimental evaluation that compares the performance of multiple different losses across four multi-label image classification datasets.

5.3 Related Work

Multi-label classification is an important and well studied problem [65, 62, 36] with applications in natural language processing [27, 28], audio classification [2, 5], information retrieval [46], and computer vision [66, 22, 15, 58, 57]. The conventional approach in vision is to train deep convolution neural networks with multiple output predictions — one for each concept/class of interest. When there are no missing labels (i.e. for each image we have complete observations of the presence and absence of each class), standard binary cross-entropy or softmax cross-entropy losses are typically used, e.g. [35, 40].

In practice, label information is often incomplete at training time because it can be extremely difficult to acquire exhaustive supervision [10]. Different approaches have been proposed to address the partially labeled setting including: assuming the missing labels are negative [51, 3, 39], ignoring missing labels [12], performing label matrix reconstruction [4, 63], learning label correlations [6, 12, 45, 25], learning

generative probabilistic models [31, 7], and training label cleaning networks [54]. It is worth noting that semi-supervised multi-label classification [37, 17, 56, 43] can be viewed as a special case of training with missing labels, where here we have entire images with no labels. The partially labelled setting is also related to methods that address label noise, e.g. [23, 24]. Label noise is also encountered in the related area of image tagging [50, 14], where only a small fraction of the potentially relevant tags are known for each image. We are interested in one special kind of label noise, where some unobserved labels are incorrectly treated as being absent. This "noise" is the result of a strong assumption, and is not label noise in the traditional sense. With the exception of the some simple approaches (e.g. assuming missing labels are negative [40]), most existing approaches assume that they have access to a subset of exhaustively labelled images, or at the very least, images with more than one confirmed positive or negative label.

We consider a setting where annotators are only asked to provide a single positive label for each training image and no additional negative or positive labels. This arises in multi-class image classification where multiple relevant objects may appear in each image but only a single class is annotated [49]. This same problem also occurs in non-vision domains such as species distribution modeling [44] where the training data are records of real-world (positive) observations for a given location, and there are no negatives. The single positive setting has advantages. When collecting multi-label annotations, it may be more efficient for a crowd worker to mark the presence of a specific class as opposed to confirming its absence.

Our setting is most closely related to positive-unlabeled (PU) learning [33]; see [1] for a recent survey focused on binary classification, which is the most commonly studied formulation of PU learning. In PU learning we only have access to a set of positive items and an additional set of unlabeled items, which may be either positive or negative. Compared to the classification setting, there are relatively few works that explore PU learning for multi-label tasks [51, 21, 30, 19], and to the best of our knowledge, there are no works that explicitly explore the single positive case in-depth. [47] and [11] address the setting where there is only a single label available for each item at training time. However, unlike in our setting, these labels can be positive *or* negative. Furthermore, when more than one positive label is available for each image, it is possible to infer class level co-occurrence information — something which is not directly possible with only single positive labels. In the multi-*class* setting, [26] proposes to learn from complementary labels, i.e. they assume access to

a single negative label per item that specifies that the item does not belong to a given class. Their solution falls under the "assume negative" set of approaches mentioned earlier, except that the positives and negative labels are reversed. Another related multi-class setting is set-valued classification, where each image has one label and the goal is to learn to predict a set of labels as a way to represent uncertainty [9]. In Section 5.5 we discuss several existing multi-label approaches in a unified context and adapt these methods for the single positive setting in Section 5.6.

5.4 Problem Statement

In the standard multi-*class* classification setting, each \mathbf{x} from the input space \mathcal{X} is assigned a single label from $\{1, \dots, L\}$, where L is the number of classes. In the multi-*label* classification setting, each \mathbf{x} is associated with a vector of labels \mathbf{y} from the label space $\mathcal{Y} = \{0, 1\}^L$, where an entry $y_i = 1$ if the i th class is relevant to \mathbf{x} and $y_i = 0$ if the i th class is not relevant.

The goal is to find a function $f : \mathcal{X} \rightarrow [0, 1]^L$ that predicts the applicable labels for each $\mathbf{x} \in \mathcal{X}$. The formal objective is to find an f that minimizes the risk

$$R(f) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim p(\mathbf{x}, \mathbf{y})} \overline{\mathcal{L}}(f(\mathbf{x}), \mathbf{y}), \quad (5.1)$$

where $\overline{\mathcal{L}} : [0, 1]^L \times \mathcal{Y} \rightarrow \mathbb{R}$ reflects some multi-label metric e.g. mean average precision or 0-1 error. In practice, we define f to be a neural network with parameters θ and we replace $\overline{\mathcal{L}}$ with a surrogate \mathcal{L} that is easier to optimize. Given an observed dataset $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$, we can use standard techniques to approximately solve

$$\hat{\theta}_{\text{full}} = \operatorname{argmin}_{\theta} \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f(\mathbf{x}_n; \theta), \mathbf{y}_n), \quad (5.2)$$

where $\mathcal{L} : [0, 1]^L \times \mathcal{Y} \rightarrow \mathbb{R}$ is a suitable multi-label loss function, e.g. binary cross-entropy or softmax cross-entropy. However, this formulation assumes that we have access to a *fully observed* label vector \mathbf{y}_n for each input \mathbf{x}_n . In this work we explore the setting where the true label vectors are not directly accessible. Instead, during training we observe $\mathbf{z}_n \in \mathcal{Z} = \{0, 1, \emptyset\}^L$, where $z_{ni} \in \{0, 1\}$ is interpreted as before, but $z_{ni} = \emptyset$ indicates that the i th label is unobserved for \mathbf{x}_n . That is, if $z_{ni} = \emptyset$ then the corresponding y_{ni} could be either 0 or 1. This is the *partially observed* setting, where we can use our training set $\{(\mathbf{x}_n, \mathbf{z}_n)\}_{n=1}^N$ to approximately solve

$$\hat{\theta}_{\text{partial}} = \operatorname{argmin}_{\theta} \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f(\mathbf{x}_n; \theta), \mathbf{z}_n), \quad (5.3)$$

where $\mathcal{L} : [0, 1]^L \times \mathcal{Z} \rightarrow \mathbb{R}$ is a multi-label loss function that can handle partially observed labels — see Section 5.5 for examples. Specifically, we focus on a particular instance of the partially observed setting which we call the *single positive only* case, where we observe one single positive label per training example and all the other labels are unknown. Formally, the single positive case is characterized by

$$\begin{aligned} z_{ni} &\in \{1, \emptyset\} \text{ for all } n \in \{1, \dots, N\}, i \in \{1, \dots, L\} \\ \sum_{i=1}^L \mathbb{1}_{[z_{ni}=1]} &= 1 \text{ for all } n \in \{1, \dots, N\}, \end{aligned} \quad (5.4)$$

where $\mathbb{1}_{[\cdot]}$ denotes the indicator function, i.e. $\mathbb{1}_{[z_{ni}=1]} = 1$ if $z_{ni} = 1$, and 0 otherwise. Intuitively we expect a lower risk for the function f learned from fully observed data, i.e. $R(f(\cdot; \hat{\theta}_{\text{full}})) \leq R(f(\cdot; \hat{\theta}_{\text{partial}}))$. The key question is: *how can we design a loss \mathcal{L} to minimize $R(f(\cdot; \hat{\theta}_{\text{partial}})) - R(f(\cdot; \hat{\theta}_{\text{full}}))$?*

5.5 Multi-Label Learning

In this section we compare and contrast three multi-label settings: fully observed labels, partially observed labels (i.e. some positives and some negatives are observed), and positive only labels (i.e. all observed labels are positive and there are no confirmed negatives). In the fully observed setting we cover the binary cross-entropy (BCE) loss. We then discuss how the standard BCE loss is modified to accommodate the partially observed and positive only settings. We focus on BCE because it is ubiquitous in multi-label classification, e.g. [54, 12], but one could carry out a similar exercise using other multi-label losses. We also compare the different variants in terms of the implicit assumptions each makes regarding unobserved labels.

First we introduce some additional notation. Let $\mathbf{f}_n = f(\mathbf{x}_n; \theta) \in [0, 1]^L$ be the vector of class probabilities predicted for \mathbf{x}_n by our multi-label classifier $f(\cdot; \theta)$, and let f_{ni} be the i th entry of \mathbf{f}_n . Note that since we are using the binary cross-entropy loss, the class probabilities f_{ni} do not sum to one over classes i .

Fully Observed Labels

The binary cross-entropy (BCE) loss is one of the simplest and most commonly used multi-label losses [42, 12]. For a fully observed data point $(\mathbf{x}_n, \mathbf{y}_n)$, the BCE loss is

$$\begin{aligned} \mathcal{L}_{\text{BCE}}(\mathbf{f}_n, \mathbf{y}_n) &= -\frac{1}{L} \sum_{i=1}^L [\mathbb{1}_{[y_{ni}=1]} \log(f_{ni}) \\ &\quad + \mathbb{1}_{[y_{ni}=0]} \log(1 - f_{ni})], \end{aligned} \quad (5.5)$$

where we have substituted $\mathbb{1}_{[y_{ni}=1]}$ for $P(y_i = 1|\mathbf{x}_n)$ and $\mathbb{1}_{[y_{ni}=0]}$ for $P(y_i = 0|\mathbf{x}_n)$. In the following sections, we present simple variants of \mathcal{L}_{BCE} that do not require fully observed data. The trade-off is that these variants make stronger implicit assumptions about the distribution $P(y_i|\mathbf{x}_n)$.

Partially Observed Labels

Suppose that we have a partially observed data point $(\mathbf{x}_n, \mathbf{z}_n)$. For observed labels we can simply let $P(y_i = 1|\mathbf{x}_n) = \mathbb{1}_{[z_{ni}=1]}$ and $P(y_i = 0|\mathbf{x}_n) = \mathbb{1}_{[z_{ni}=0]}$ just like we did for \mathcal{L}_{BCE} . However, it is not clear what to do if a label is unobserved (i.e. $z_{ni} = \emptyset$). One idea is to simply set the loss terms corresponding to unobserved labels to zero, resulting in the "ignore unobserved" (IU) loss

$$\begin{aligned} \mathcal{L}_{\text{IU}}(\mathbf{f}_n, \mathbf{z}_n) = & -\frac{1}{L} \sum_{i=1}^L [\mathbb{1}_{[z_{ni}=1]} \log(f_{ni}) \\ & + \mathbb{1}_{[z_{ni}=0]} \log(1 - f_{ni})]. \end{aligned} \quad (5.6)$$

This loss implicitly assumes that unobserved labels are perfectly predicted, i.e. $f_{ni} = P(y_i = 1|\mathbf{x}_n)$ if $z_{ni} = \emptyset$. If we additionally weight $\mathcal{L}_{\text{IU}}(\mathbf{f}_n, \mathbf{z}_n)$ by the number of observed labels in \mathbf{z}_n then we obtain the loss used in [12] (up to scaling).

The \mathcal{L}_{IU} loss allows for missing labels, but it requires both positive and negative labels. Our focus is the positive-only setting, in which these losses collapse to the trivial "always predict positive" solution due to the absence of any negative training examples. Though these losses are inapplicable in our setting, we discuss them to clarify the relationship between our work and [12]. In addition, we use variants of \mathcal{L}_{IU} as conceptual tools in our experiments. In particular, we use a version that "ignores unobserved negatives" (IUN), given by

$$\begin{aligned} \mathcal{L}_{\text{IUN}}(\mathbf{f}_n, \mathbf{z}_n, \mathbf{y}_n) = & -\frac{1}{L} \sum_{i=1}^L [\mathbb{1}_{[z_{ni}=1]} \log(f_{ni}) \\ & + \mathbb{1}_{[y_{ni}=0]} \log(1 - f_{ni})]. \end{aligned} \quad (5.7)$$

This is similar to \mathcal{L}_{IU} except with unrealistic access to all of the true negative labels. This hypothetical loss provides an intermediate step between the fully labeled setting and the positive only setting.

Positive Only Labels

Suppose that we have partially observed data $(\mathbf{x}_n, \mathbf{z}_n)$ and suppose that all of the observed labels are positive i.e. $z_{ni} \neq \emptyset \implies z_{ni} = 1$. We know what to do with

observed labels, i.e. we set $P(y_i = 1|\mathbf{x}_n) = \mathbb{1}_{[z_{ni}=1]}$. However, we cannot simply ignore the unobserved labels because that would lead to the degenerate "always predict positive" solution. The simplest approach is to assume unobserved labels are negative, i.e. $P(y_{ni} = 1|\mathbf{x}_n) = 0$ if $z_{ni} = \emptyset$. The resulting "assume negative" (AN) loss is given by

$$\mathcal{L}_{\text{AN}}(\mathbf{f}_n, \mathbf{z}_n) = -\frac{1}{L} \sum_{i=1}^L [\mathbb{1}_{[z_{ni}=1]} \log(f_{ni}) + \mathbb{1}_{[z_{ni} \neq 1]} \log(1 - f_{ni})]. \quad (5.8)$$

This is perhaps the most common approach to the positive only setting, and is explored as "noisy+" in [12], among others [29, 40, 32]. The drawback is that \mathcal{L}_{AN} will introduce some number of false negatives. Note that if the role of positive and negative labels are reversed, then this formulation is equivalent to complementary label learning [26].

5.6 Learning From Only Positive Labels

In typical multi-label datasets there are far more negative labels than positive labels. This means that in the single positive setting, \mathcal{L}_{AN} will actually get almost all of the unobserved labels correct. However, as we demonstrate in our experiments later, even these few false negatives can significantly reduce performance. An ideal solution to this problem would (i) reduce the damaging effects of false negatives while (ii) retaining as much of the simplicity of \mathcal{L}_{AN} as possible. With these goals in mind, we propose four ideas for mitigating the impact of false negatives: *weak negatives*, *label smoothing*, *expected positive regularization*, and *online label estimation*.

Weak Negatives

A simple way to reduce the impact of false negatives is to down-weight terms in the loss corresponding to negative labels. We introduce a weight parameter $\gamma \in [0, 1]$ and define the "weak assume negative" (WAN) loss as

$$\begin{aligned} \mathcal{L}_{\text{WAN}}(\mathbf{f}_n, \mathbf{z}_n) = & -\frac{1}{L} \sum_{i=1}^L [\mathbb{1}_{[z_{ni}=1]} \log(f_{ni}) \\ & + \mathbb{1}_{[z_{ni} \neq 1]} \gamma \log(1 - f_{ni})]. \end{aligned}$$

The "interesting" values of γ lie strictly between 0 and 1, since $\gamma = 1$ recovers the standard BCE loss and $\gamma = 0$ admits a trivial solution ("always predict positive"). In the single positive setting, if we choose $\gamma = \frac{1}{L-1}$ then the single positive has the

same influence on the loss as the $L - 1$ assumed negatives. This is similar to the loss used by [39], which uses single positive labels to learn spatio-temporal priors for image classification. Throughout this paper we use $\gamma = 1/(L - 1)$.

Connection to pseudo-negative sampling. $\mathcal{L}_{\text{WAN}(\gamma)}$ has a probabilistic interpretation based on sampling negatives at random. Consider the following procedure: each time $(\mathbf{x}_n, \mathbf{z}_n)$ occurs in a batch, choose one of the $L - 1$ unobserved labels uniformly at random and treat it as negative. We repeat this step each time the pair $(\mathbf{x}_n, \mathbf{z}_n)$ appears in a batch. Since there are typically many more negatives than positives for a given image, our randomly chosen *pseudo-negative* will be a true negative more often than not. Since we now have both positive and negative labels, we can use the \mathcal{L}_{IU} loss, resulting in

$$-\frac{1}{L} \sum_{i=1}^L [\mathbb{1}_{[z_{ni}=1]} \log(f_{ni}) + \mathbb{1}_{[z_{ni} \neq 1]} \eta_{ni} \log(1 - f_{ni})],$$

where η_{ni} is a random variable which is 1 if z_{ni} is chosen as the pseudo-negative and 0 otherwise. If we take the expectation with respect to the pseudo-negative sampling then we recover \mathcal{L}_{WAN} with $\gamma = \frac{1}{L-1}$. Though the two losses are equivalent in expectation, they may differ significantly in practice.

Label Smoothing

Label smoothing was proposed in [52] as a way to reduce overfitting when training multi-class classifiers with the categorical cross-entropy loss. Label smoothing has since been shown to mitigate the effects of label noise in the multi-class setting [41]. If we reframe \mathcal{L}_{AN} as \mathcal{L}_{BCE} with some "noisy" labels (i.e. those labels incorrectly assumed to be negative), then it is natural to ask whether label smoothing could help to reduce the impact of those incorrect labels.

In a multi-class context, the target distribution \mathbf{y}_n is a delta distribution supported on the correct class label. Label smoothing replaces \mathbf{y}_n with $(1 - \epsilon)\mathbf{y}_n + \epsilon\mathbf{u}$ where $\mathbf{u} = [1/L, \dots, 1/L]$ is the discrete uniform distribution with support size L and $\epsilon \in (0, 1)$ is a hyperparameter. It is possible to generalize traditional multi-class label smoothing to the binary cross-entropy loss, by simply applying label smoothing independently to each of the L binary target distributions $(\mathbb{1}_{[z_{ni} \neq 1]}, \mathbb{1}_{[z_{ni}=1]})$. We refer to the combination of the "assume negative" loss from Eqn. 5.8 with label

smoothing as

$$\begin{aligned} \mathcal{L}_{\text{AN-LS}}(\mathbf{f}_n, \mathbf{z}_n) = & -\frac{1}{L} \sum_{i=1}^L [\mathbb{1}_{[z_{ni}=1]}^{\frac{\epsilon}{2}} \log(f_{ni}) \\ & + \mathbb{1}_{[z_{ni} \neq 1]}^{\frac{\epsilon}{2}} \log(1 - f_{ni})], \end{aligned} \quad (5.9)$$

where ϵ is the label smoothing parameter and $\mathbb{1}_{[Q]}^\alpha = (1 - \alpha) \mathbb{1}_{[Q]} + \alpha \mathbb{1}_{[-Q]}$ for any logical proposition Q . Throughout this paper we use $\epsilon = 0.1$.

Expected Positive Regularization

Another way to avoid the label noise introduced by assuming unobserved labels are negative is to apply a loss to only the observed labels as in [12]. However, in the positive-only case the loss would be

$$\mathcal{L}_{\text{BCE}}^+(\mathbf{f}_n, \mathbf{z}_n) = - \sum_{i=1}^L \mathbb{1}_{[z_{ni}=1]} \log(f_{ni}),$$

which has a trivial solution, i.e. predict that every label is positive. We propose to build some domain knowledge into the loss to avoid this problem. Let us assume we have access to a scalar k , which is defined as the expected number of positive labels per image:

$$k = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim p_{\text{data}}(\mathbf{x}, \mathbf{y})} \sum_{i=1}^L \mathbb{1}_{[y_i=1]}.$$

We can estimate k from data or treat it as a hyperparameter.

Suppose we draw a batch of images with indices $B \subset \{1, \dots, N\}$. We define $\mathbf{F}_B = [f_{ni}]_{n \in B, i \in \{1, \dots, L\}}$ to be the matrix of predictions $f_{ni} \in [0, 1]$ for every image in the batch and category in the dataset. We can use the batch predictions \mathbf{F}_B to compute

$$\hat{k}(\mathbf{F}_B) = \frac{\sum_{n \in B} \sum_{i=1}^L \mathbf{f}_{ni}}{|B|}.$$

Ideally we would make perfect predictions, i.e. $\mathbf{F}_B = \mathbf{Y}_B$ where $\mathbf{Y}_B = [y_{ni}]_{n \in B, i \in \{1, \dots, L\}}$ is the matrix of true labels. A necessary condition for $\mathbf{F}_B = \mathbf{Y}_B$ is $\mathbb{E}[\hat{k}(\mathbf{F}_B)] = \mathbb{E}[\hat{k}(\mathbf{Y}_B)]$, where the expectation is taken over batch sampling. Since $\mathbb{E}[\hat{k}(\mathbf{Y}_B)] = k$ by the definition of k , it makes sense to introduce a regularization term $R_k(\mathbf{F}_B)$ that encourages $\hat{k}(\mathbf{F}_B)$ to be close to k . We can use this regularizer to implicitly penalize negatives and avoid the trivial "always predict positive" solution, leading to the loss

$$\mathcal{L}_{\text{EPR}}(\mathbf{F}_B, \mathbf{Z}_B) = \frac{1}{|B|} \sum_{n \in B} \mathcal{L}_{\text{BCE}}^+(\mathbf{f}_n, \mathbf{z}_n) + \lambda R_k(\mathbf{F}_B),$$

where λ is a hyperparameter. Regularizing at the batch level (instead of the image level) respects the fact that some images will have more than k positive labels and some will have fewer.

How should we define $R_k(\mathbf{F}_B)$? Since the number of classes L can vary widely depending on the dataset, we propose to work with the normalized deviation $(\hat{k}(\mathbf{F}_B) - k)/L \in [-1, 1]$. Penalizing this relative deviation makes sense in contexts where, e.g., an absolute deviation of 1 matters more if $L = 10$ than it does if $L = 100$. We can then define a variety of regularizers with any standard functional form. We use the squared error, leading to

$$R_k(\mathbf{F}_B) = \left(\frac{\hat{k}(\mathbf{F}_B) - k}{L} \right)^2. \quad (5.10)$$

Online Estimation of Unobserved Labels

While the idea behind \mathcal{L}_{EPR} seems reasonable, we find that it does not work well in our experiments (see Section 5.7). In this section we combine \mathcal{L}_{EPR} with a second module which maintains online estimates of the unobserved labels throughout training. The resulting method is similar to an expectation-maximization algorithm which jointly trains the image classifier and estimates the labels subject to constraints imposed by \mathcal{L}_{EPR} . We refer to this technique as *regularized online label estimation* (ROLE).

To make this more precise we will need some additional notation. We write the estimated labels as $\tilde{\mathbf{Y}} \in [0, 1]^{N \times L}$ in analogy with the matrix of true labels $\mathbf{Y} \in \{0, 1\}^{N \times L}$ and the matrix of classifier predictions $\mathbf{F} \in [0, 1]^{N \times L}$. We carry through the derived notation: $\tilde{\mathbf{Y}}_B \in [0, 1]^{|B| \times L}$ for a batch B , $\tilde{\mathbf{y}}_n \in [0, 1]^L$ for a row, and $\tilde{y}_{ni} \in [0, 1]$ for a single entry. Finally, we make the (non-restrictive) assumption that $\tilde{\mathbf{y}}_n = g(\mathbf{x}_n; \phi)$ where the *label estimator* $g : \mathcal{X} \rightarrow [0, 1]^L$ is some function with parameters ϕ . We discuss our implementation of g later.

With this notation, our goal is to jointly train the label estimator $g(\cdot; \phi)$ and the image classifier $f(\cdot; \theta)$. We first consider the intermediate loss

$$\begin{aligned} \mathcal{L}'(\mathbf{F}_B | \tilde{\mathbf{Y}}_B) &= \frac{1}{|B|} \sum_{n \in B} \mathcal{L}_{\text{BCE}}(\mathbf{f}_n, \text{sg}(\tilde{\mathbf{y}}_n)) \\ &+ \mathcal{L}_{\text{EPR}}(\mathbf{F}_B, \mathbf{Z}_B), \end{aligned} \quad (5.11)$$

where sg is the stop-gradient function which prevents its argument from backpropagating gradients [16] and we have suppressed the dependence on \mathbf{Z}_B on the left-hand

side because \mathbf{Z} is fixed throughout training. The \mathcal{L}_{BCE} term encourages the image classifier predictions \mathbf{F}_B to match the estimated labels $\tilde{\mathbf{Y}}_B$, while the \mathcal{L}_{EPR} term pushes \mathbf{F}_B to correctly predict known positives and respect the expected number of positives per image. We can use this loss to update θ while assuming that ϕ is fixed. By switching the arguments in Eqn. 5.11 we obtain an analogous loss which allows us to update ϕ while assuming θ is fixed. Then our final loss is simply

$$\mathcal{L}_{\text{ROLE}}(\mathbf{F}_B, \tilde{\mathbf{Y}}_B) = \frac{\mathcal{L}'(\mathbf{F}_B|\tilde{\mathbf{Y}}_B) + \mathcal{L}'(\tilde{\mathbf{Y}}_B|\mathbf{F}_B)}{2}$$

through which we can update \mathbf{F}_B and $\tilde{\mathbf{Y}}_B$ simultaneously.

We now give some intuition for why this might work. We start with an informal proposition: all else being equal, a convolutional network will more readily train on informative labels than on uninformative labels. Concretely, it has been observed that convolutional neural networks can be trained to accurately predict completely random labels, but the same network will fit to the correct labels much faster [64]. How does this relate to our context? $\mathcal{L}_{\text{ROLE}}$ allows the labels to be set arbitrarily, as long as they are consistent with the known labels and the expected number of positive labels. Since it is easier to train image classifiers on informative labels than uninformative ones, we hypothesize that *correct labels are a "good choice" from the algorithm's perspective*. While it is possible to learn to predict labels unrelated to the image content, in many cases it may be easier to predict the correct ones.

5.7 Experiments

Here we present multi-label image classification results on four standard benchmark datasets: PASCAL VOC 2012 (VOC12) [13], MS-COCO 2014 (COCO) [34], NUS-WIDE (NUS) [8], and CUB-200-2011 (CUB) [55]. For each dataset we present results for both (i) linear classification on fixed features and (ii) end-to-end fine-tuning.

Implementation Details

Data preparation. Our goal is to evaluate the performance of different single positive multi-label learning losses. To do this, we begin with fully labeled multi-label image datasets and corrupt them by discarding annotations. Specifically, we simulate single positive training data by randomly selecting one positive label to keep for each training example. This is performed once for each dataset and the same label set is used for all comparisons on that dataset, i.e. every time an image appears in a batch it has the same single positive label. For each dataset, we withhold 20% of

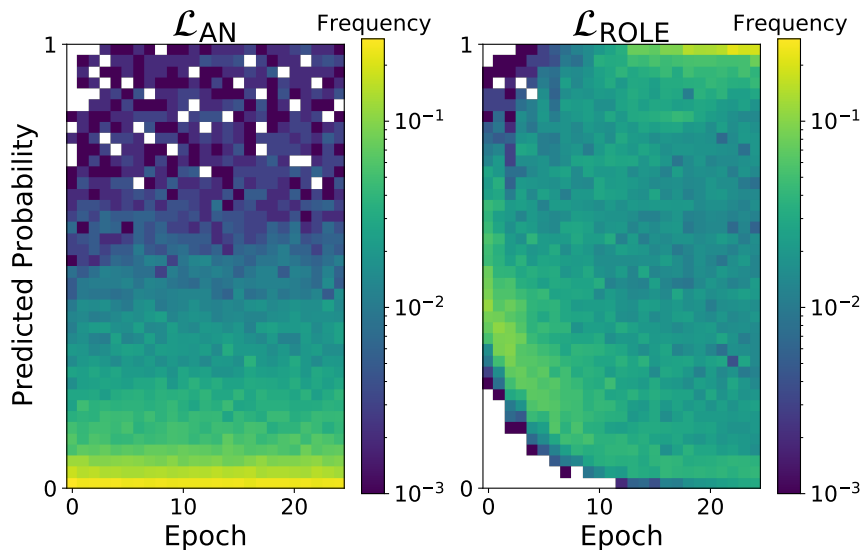


Figure 5.2: Distribution of predicted probabilities for *unobserved* positives when training with a single positive per image for COCO. Each column represents a normalized histogram and white pixels indicate a frequency of zero. Training with $\mathcal{L}_{\text{ROLE}}$ (right) results in the recovery of a significant number of the unlabeled positives as evident by the majority of the probability correctly being concentrated at 1.0 (top right) by the end of training. \mathcal{L}_{AN} (left) does not exhibit the same behavior.

the training set for validation. The validation and test sets are always fully labeled. VOC12 contains 5,717 training images and 20 classes, and we report results on the official validation set (5,823 images). COCO consists of 82,081 training images and 80 classes, and we also report results on the official validation set (40,137 images). The complete NUS dataset is not available online so we re-scraped it from Flickr. As a result, it was not possible to obtain all of the original images. In total, we collected 126,034 and 84,226 images from the official training and test sets respectively, consisting of 81 classes. In accordance with standard practice [15, 12], we merged the training and test sets and randomly selected 150,000 images for training and used the remaining 60,260 for testing. CUB is divided into 5,994 training images and 5,794 test images. Each CUB image is associated with a vector indicating the presence or absence of 312 binary attributes. Note that subsets of these attributes are known to be mutually exclusive, but we do not make use of that information. We provide additional statistics on the datasets in the supplementary material.

Hyperparameters. For each method, we conducted a hyperparameter search and selected the hyperparameters with the best mean average precision (MAP) on the validation set. We considered learning rates in $\{1e-2, 1e-3, 1e-4, 1e-5\}$ and

Loss	Labels Per Image	Linear				Fine-Tuned			
		VOC12	COCO	NUS	CUB	VOC12	COCO	NUS	CUB
\mathcal{L}_{BCE}	All Pos. & All Neg.	86.7	70.0	50.7	29.1	89.1	75.8	52.6	32.1
$\mathcal{L}_{\text{BCE-LS}}$	All Pos. & All Neg.	87.6	70.2	51.7	29.3	90.0	76.8	53.5	32.6
\mathcal{L}_{IUN}	1 Pos. & All Neg.	86.4	67.0	49.0	19.4	87.1	70.5	46.9	21.3
\mathcal{L}_{IU}	1 Pos. & 1 Neg.	82.6	60.8	43.6	16.1	83.2	59.7	42.9	17.9
\mathcal{L}_{AN}	1 Pos. & 0 Neg.	84.2	62.3	46.2	17.2	85.1	64.1	42.0	19.1
$\mathcal{L}_{\text{AN-LS}}$	1 Pos. & 0 Neg.	<u>85.3</u>	<u>64.8</u>	<u>48.5</u>	15.4	86.7	66.9	44.9	17.9
\mathcal{L}_{WAN}	1 Pos. & 0 Neg.	84.1	63.1	45.8	<u>17.9</u>	86.5	64.8	46.3	20.3
\mathcal{L}_{EPR}	1 Pos. & 0 Neg.	83.8	62.6	46.4	18.0	85.5	63.3	46.0	<u>20.0</u>
$\mathcal{L}_{\text{ROLE}}$	1 Pos. & 0 Neg.	86.5	66.3	49.5	16.2	<u>87.9</u>	66.3	43.1	15.0
$\mathcal{L}_{\text{AN-LS}} + \text{LinearInit.}$	1 Pos. & 0 Neg.	-	-	-	-	86.5	69.2	<u>50.5</u>	16.6
$\mathcal{L}_{\text{ROLE}} + \text{LinearInit.}$	1 Pos. & 0 Neg.	-	-	-	-	88.2	<u>69.0</u>	51.0	16.8

Table 5.1: Multi-label test set mean average precision (MAP) for different multi-label losses on four different image classification datasets. We present results for two scenarios: (i) training a linear classifier on fixed features and (ii) fine-tuning the entire network end-to-end. In all cases the backbone network is an ImageNet pre-trained ResNet-50. All methods below the break use only one positive per image (i.e. 1 Pos. & 0 Neg.), while methods above the break use additional supervision. In each column we bold the best performing single positive method and underline the second-best. For each method and we select the hyperparameters that perform the best on the held-out validation set. For losses labeled with "LinearInit." we freeze the weights of the backbone network for the initial epochs of training and then fine-tune the entire network end-to-end for the remaining epochs. Note that this linear initialization phase is identical to the training protocol for the "Linear" results.

Loss	VOC12	COCO	NUS	CUB
\mathcal{L}_{AN}	85.8	63.8	49.3	16.8
$\mathcal{L}_{\text{AN-LS}}$	86.9	65.4	49.7	17.4
$\mathcal{L}_{\text{ROLE}}$	90.3	69.5	56.0	19.6

Table 5.2: Training set MAP for multi-label predictions evaluated with respect to the *full* ground truth labels. These values measure how well each method recovers the true training labels despite being trained with one positive label per image. Note that all results are for the linear case. Hyperparameters and stopping epoch are selected using the validation set as before.

batch sizes in $\{8, 16\}$. We train for 25 epochs in the linear case and 10 epochs in the fine-tuned case. The rows tagged with "+LinearInit" are fine-tuned for 5 epochs starting from the best weights found during linear training. For $\mathcal{L}_{\text{ROLE}}$ we set the learning rate for the label estimate parameters ϕ to be $10\times$ larger than the learning rate for the image classifier parameters θ . For \mathcal{L}_{EPR} and $\mathcal{L}_{\text{ROLE}}$ we compute k based on the fully labeled training set - we give these values and study the effect of mis-specifying k in the supplementary material. All experiments are based on a ResNet-50 [20] pre-trained on ImageNet [49].

Implementation of g for $\mathcal{L}_{\text{ROLE}}$. We let $\phi \in [0, 1]^{N \times L}$ and define $\tilde{\mathbf{Y}}$ by $\tilde{y}_{ni} = \sigma(\phi_{ni})$ where $\sigma : \mathbb{R} \rightarrow (0, 1)$ is the sigmoid function. As a result, g is a simple "look-up" operation given by $g(\mathbf{x}_n; \phi) = \tilde{\mathbf{y}}_n$. We initialize ϕ_{ni} from the uniform distribution on $[\sigma^{-1}(0.4), \sigma^{-1}(0.6)]$ if $z_{ni} = 0$ or we initialize $\phi_{ni} = \sigma^{-1}(0.995)$ if $z_{ni} = 1$. Note that this does not apply to " $\mathcal{L}_{\text{ROLE}}+\text{LinearInit.}$ " which starts from the ϕ parameters found during linear training.

Single Positive Classification Results

In Table 5.1 we evaluate the different training losses outlined earlier in the paper in the single positive case (i.e. "1 Pos. & 0 Neg.") and compare their performance to other labeling regimes (e.g. fully labeled, "All Pos. & All Neg."). We also compare against intermediate variants such as \mathcal{L}_{IUN} , which has access to one positive label per image and all the negatives i.e. more labels than the single positive case, but fewer than the fully labeled case. We find that $\mathcal{L}_{\text{ROLE}}$ is the strongest method in the linear case, often approaching (and sometimes surpassing) the performance of \mathcal{L}_{IUN} , which has access to many more labels at training time. In the fine-tuned case, we see that better initialization provides substantial benefits to both $\mathcal{L}_{\text{ROLE}}$ and \mathcal{L}_{AN} (see rows with "+LinearInit."). However, $\mathcal{L}_{\text{AN-LS}}$ is also very effective, especially in light of its simplicity.

Single positive training performs surprisingly well. One way to better understand the overall performance is by comparing different losses in terms of the number of training labels used. In Figure 5.1 we observe that in the linear case, $\mathcal{L}_{\text{ROLE}}$ achieves test MAP comparable to the fully labelled loss (\mathcal{L}_{BCE}) on VOC12, despite using 20 times fewer labels.

The choice of single positive loss matters. While we have discussed the shortcomings of the assume negative baseline \mathcal{L}_{AN} , we observe that it performs reasonably well. However, we note that the gap between \mathcal{L}_{AN} and the fully supervised \mathcal{L}_{BCE} is substantially wider in the end-to-end fine-tuned case. Presumably this is due to the fact that the false negative labels can do much more damage when they are able to corrupt the backbone feature extractor. This result adds to a broader conversation (which has mostly been focused on the multi-class setting) about whether, and to what extent, deep learning is robust to label noise [48]. Our multi-label label smoothing variant $\mathcal{L}_{\text{AN-LS}}$ and our $\mathcal{L}_{\text{ROLE}}$ loss perform much better in most cases, indicating that the widely used \mathcal{L}_{AN} baseline is a lower bound on performance. We also note that although \mathcal{L}_{EPR} typically performs worse than \mathcal{L}_{AN} , it seems to work

quite well for CUB. CUB is unusual among our datasets because the average number of positive labels per image is over 30 (more than $10\times$ higher than VOC12, COCO, and NUS). We suspect that the relatively mild loss applied to unobserved labels under \mathcal{L}_{EPR} may be beneficial when there are so many unobserved positives.

Label smoothing is a strong baseline. [38] showed that label smoothing mitigates the damaging effects of label noise in the multi-class setting. We extend these results to the multi-label setting. We see in Table 5.1 that $\mathcal{L}_{\text{AN-LS}}$ (i.e. assume negative with label smoothing) outperforms the basic assume negative \mathcal{L}_{AN} loss in nearly every case. It is also worth noting that label smoothing provides a larger benefit in the single positive case ($\mathcal{L}_{\text{AN-LS}}$ vs. \mathcal{L}_{AN}) than it does in the fully labeled case ($\mathcal{L}_{\text{BCE-LS}}$ vs. \mathcal{L}_{BCE}). We therefore recommend $\mathcal{L}_{\text{AN-LS}}$ as a strong and simple baseline for the single positive multi-label setting. However, training with our $\mathcal{L}_{\text{ROLE}}$ loss still performs best in most settings. $\mathcal{L}_{\text{ROLE}}$ requires more parameters to be estimated at training time, but incurs no additional computational overhead at inference time. In Table 5.2 we present MAP scores computed on the fully observed training set for losses trained with only a single positive per image. Interestingly, we observe that $\mathcal{L}_{\text{ROLE}}$ does a better job at recovering the full unobserved label matrix when compared to $\mathcal{L}_{\text{AN-LS}}$. This is illustrated qualitatively in Figure 5.2, which shows that $\mathcal{L}_{\text{ROLE}}$ can successfully recover many of the unobserved positive labels during training. However, as seen in Table 5.1, this better recovery of the clean training labels does not necessarily translate to comparable gains on the test set.

Initialization matters. $\mathcal{L}_{\text{ROLE}}$ is very effective in the linear setting i.e. when training a randomly initialized linear classifier and label estimator on frozen backbone features. However, we find that starting from a randomly initialized classifier and label estimator in the end-to-end setting results in an inferior model. This is perhaps not too surprising given the additional degrees of freedom afforded by end-to-end fine-tuning. However, as a simple remedy we recommend starting with a frozen backbone for the first few epochs of end-to-end training, which is denoted in Table 5.1 as $\mathcal{L}_{\text{ROLE}} + \text{LinearInit}$. We observe that this procedure also provides substantial benefits for $\mathcal{L}_{\text{AN-LS}}$, the label smoothed version of the assume negative training loss.

5.8 Limitations

When creating our simulated training annotations, our single positive label generation process assumes that for a given image any positive label that is present is

equally likely to be annotated. This is in line with similar assumptions made in other related work e.g. [12]. However, in practice this is an oversimplification, as human annotators are likely to have biases related to the object categories they annotate. Depending on the specific dataset, this could be manifested as a preference for annotating familiar object categories, or it could be based on factors related to the saliency of the object instance in the image e.g. smaller objects may be less likely to be annotated compared to larger ones. In this work we focus on better understanding the potential of single positive training, and leave modeling annotation biases to future work.

Our $\mathcal{L}_{\text{ROLE}}$ loss requires the online estimation of an $N \times L$ label matrix. As presented, we store the full label matrix in memory. For a dataset like ImageNet this would require 4GB of memory, but would become infeasible for larger datasets or larger numbers of labels. Possible alternative implementations of the label estimator g (which would still be fully compatible with our loss) include learning a factorized estimate of the matrix or using a small neural network to approximate it.

5.9 Conclusion

We have investigated an underexplored variant of partially observed multi-label classification — that of single positive training. Perhaps surprisingly, we have showed that in this supervision deprived setting it is possible to achieve classification results that are competitive with full label supervision using an order of magnitude fewer labels. This opens up future avenues of work related to efficient crowdsourcing of annotations for large-scale multi-label datasets. In future work we intend to further explore the connections to semi-supervised multi-label classification along with applications in self-supervised representation learning where the problem of how to address false negative labels often occurs. In addition, many of the ideas discussed are applicable to the more general "partially observed multi-label" case (i.e. not just positive labels), and we plan to consider extensions to that setting also.

5.10 Acknowledgements

This project was supported in part by an NSF Graduate Research Fellowship (Grant No. DGE1745301) and the Microsoft AI for Earth program. We would also like to thank Jennifer J. Sun, Matteo Ruggero Ronchi, and Joseph Marino for helpful feedback.

References

- [1] Jessa Bekker and Jesse Davis. “Learning from positive and unlabeled data: a survey”. In: *Machine Learning* (2020).
- [2] Forrest Briggs et al. “Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach”. In: *The Journal of the Acoustical Society of America* (2012).
- [3] Serhat Selcuk Bucak, Rong Jin, and Anil K Jain. “Multi-label learning with incomplete class assignments”. In: *CVPR*. 2011.
- [4] Ricardo S Cabral et al. “Matrix completion for multi-label image classification”. In: *NeurIPS*. 2011.
- [5] Emre Cakir et al. “Polyphonic sound event detection using multi label deep neural networks”. In: *IJCNN*. 2015.
- [6] Zhao-Min Chen et al. “Multi-label image recognition with graph convolutional networks”. In: *CVPR*. 2019.
- [7] Hong-Min Chu, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. “Deep Generative Models for Weakly-Supervised Multi-Label Classification”. In: *ECCV*. 2018.
- [8] Tat-Seng Chua et al. “NUS-WIDE: a real-world web image database from National University of Singapore”. In: *International Conference on Image and Video Retrieval*. 2009.
- [9] Evgenii Chzhen et al. “Set-valued classification—overview via a unified framework”. In: *arXiv:2102.12318* (2021).
- [10] Jia Deng et al. “Scalable multi-label annotation”. In: *CHI*. 2014.
- [11] Junhong Duan, Xiaoyu Li, and Dejun Mu. “Learning Multi Labels from Single Label - An Extreme Weak Label Learning Algorithm”. In: *Wuhan University Journal of Natural Sciences* (2019).
- [12] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. “Learning a Deep ConvNet for Multi-label Classification with Partial Labels”. In: *CVPR* (2019).
- [13] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*.
- [14] Jianlong Fu and Yong Rui. “Advances in deep learning approaches for image tagging”. In: *APSIPA Transactions on Signal and Information Processing 6* (2017).
- [15] Yunchao Gong et al. “Deep Convolutional Ranking for Multilabel Image Annotation”. In: *ICLR*. 2014.
- [16] Jean-Bastien Grill et al. “Bootstrap Your Own Latent-A New Approach to Self-Supervised Learning”. In: *NeurIPS* (2020).

- [17] Yuhong Guo and Dale Schuurmans. “Semi-supervised multi-label classification”. In: *ECML*. 2012.
- [18] Agrim Gupta, Piotr Dollar, and Ross Girshick. “Lvis: A dataset for large vocabulary instance segmentation”. In: *CVPR*. 2019.
- [19] Yufei Han et al. “Multi-label Learning with Highly Incomplete Data via Collaborative Embedding”. In: *KDD*. 2018.
- [20] Kaiming He et al. “Deep residual learning for image recognition”. In: *CVPR*. 2016.
- [21] Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit S Dhillon. “PU Learning for Matrix Completion”. In: *ICML*. 2015.
- [22] Daniel J Hsu et al. “Multi-label prediction via compressed sensing”. In: *NeurIPS*. 2009.
- [23] Mengying Hu et al. “Multi-label Learning from Noisy Labels with Non-linear Feature Transformation”. In: *ACCV* (2018).
- [24] Mengying Hu et al. “Weakly Supervised Image Classification through Noise Regularization”. In: *CVPR* (2019).
- [25] Dat Huynh and Ehsan Elhamifar. “Interactive multi-label CNN learning with partial labels”. In: *CVPR*. 2020.
- [26] Takashi Ishida et al. “Learning from complementary labels”. In: *NeurIPS*. 2017.
- [27] Rie Johnson and Tong Zhang. “Effective Use of Word Order for Text Categorization with Convolutional Neural Networks”. In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015.
- [28] Armand Joulin et al. “Bag of Tricks for Efficient Text Classification”. In: *Conference of the European Chapter of the Association for Computational Linguistics*. 2017.
- [29] Armand Joulin et al. “Learning Visual Features from Large Weakly Supervised Data”. In: *ECCV* (2016).
- [30] Atsushi Kanehira and Tatsuya Harada. “Multi-label Ranking from Positive and Unlabeled Data”. In: *CVPR* (2016).
- [31] Ashish Kapoor, Raajay Viswanathan, and Prateek Jain. “Multilabel classification using bayesian compressed sensing”. In: *NeurIPS*. 2012.
- [32] Kaustav Kundu and Joseph Tighe. “Exploiting weakly supervised visual patterns to learn from partial annotations”. In: *NeurIPS* (2020).
- [33] Xiaoli Li and Bing Liu. “Learning to classify texts using positive and unlabeled data”. In: *IJCAI*. 2003.

- [34] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *ECCV*. 2014.
- [35] Jingzhou Liu et al. “Deep Learning for Extreme Multi-label Text Classification”. In: *SIGIR*. 2017.
- [36] Weiwei Liu et al. “The Emerging Trends of Multi-Label Learning”. In: *arXiv* (2020).
- [37] Yi Liu, Rong Jin, and Liu Yang. “Semi-supervised multi-label learning by constrained non-negative matrix factorization”. In: *AAAI*. 2006.
- [38] Michal Lukasik et al. “Does label smoothing mitigate label noise?” In: *ICML*. 2020.
- [39] Oisín Mac Aodha, Elijah Cole, and Pietro Perona. “Presence-Only Geographical Priors for Fine-Grained Image Classification”. In: *ICCV* (2019).
- [40] Dhruv Mahajan et al. “Exploring the limits of weakly supervised pretraining”. In: *ECCV*. 2018.
- [41] Rafael Muller, Simon Kornblith, and Geoffrey Hinton. “When Does Label Smoothing Help?” In: *NeurIPS*. 2019.
- [42] Jinseok Nam et al. “Large-Scale Multi-label Text Classification - Revisiting Neural Networks”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2014).
- [43] Xuesong Niu et al. “Multi-label Co-regularization for Semi-supervised Facial Action Unit Recognition”. In: *NeurIPS*. 2019.
- [44] Steven J Phillips, Miroslav Dudík, and Robert E Schapire. “A maximum entropy approach to species distribution modeling”. In: *ICML*. 2004.
- [45] Luis Pineda et al. “Elucidating image-to-set prediction: An analysis of models, losses and datasets”. In: *arXiv:1904.05709* (2019).
- [46] Yashoteja Prabhu and Manik Varma. “Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning”. In: *SIGKDD*. 2014.
- [47] Shuang Qiu et al. “Nonconvex One-bit Single-label Multi-label Learning”. In: *arXiv:1703.06104* (2017).
- [48] David Rolnick et al. “Deep learning is robust to massive label noise”. In: *arXiv:1705.10694* (2017).
- [49] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *IJCV* (2015).
- [50] Jitao Sang, Changsheng Xu, and Jing Liu. “User-aware image tag refinement via ternary semantic analysis”. In: *IEEE Transactions on Multimedia* (2012).
- [51] Yu-Yin Sun, Yin Zhang, and Zhi-Hua Zhou. “Multi-Label Learning with Weak Label”. In: *AAAI* (2010).

- [52] Christian Szegedy et al. “Rethinking the inception architecture for computer vision”. In: *CVPR*. 2016.
- [53] Grant Van Horn et al. “The iNaturalist species classification and detection dataset”. In: *CVPR*. 2018.
- [54] Andreas Veit et al. “Learning From Noisy Large-Scale Datasets With Minimal Supervision”. In: *CVPR (2017)*.
- [55] C. Wah et al. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.
- [56] Bo Wang, Zhuowen Tu, and John K Tsotsos. “Dynamic label propagation for semi-supervised multi-class multi-label classification”. In: *ICCV*. 2013.
- [57] Jiang Wang et al. “CNN-RNN: A Unified Framework for Multi-Label Image Classification”. In: *CVPR*. 2016.
- [58] Yunchao Wei et al. “HCP: A flexible CNN framework for multi-label image classification”. In: *PAMI (2015)*.
- [59] Jeremy M Wolfe. “Visual search”. In: *Current biology (2010)*.
- [60] Jeremy M Wolfe, Todd S Horowitz, and Naomi M Kenner. “Rare items often missed in visual searches”. In: *Nature (2005)*.
- [61] Baoyuan Wu et al. “Tencent ml-images: A large-scale multi-label image database for visual representation learning”. In: *IEEE Access (2019)*.
- [62] Donna Xu et al. “Survey on multi-output learning”. In: *IEEE transactions on neural networks and learning systems (2019)*.
- [63] Miao Xu, Rong Jin, and Zhi-Hua Zhou. “Speedup matrix completion with side information: Application to multi-label learning”. In: *NeurIPS*. 2013.
- [64] Chiyuan Zhang et al. “Understanding Deep Learning Requires Rethinking Generalization”. In: *ICLR*. 2017.
- [65] Min-Ling Zhang and Zhi-Hua Zhou. “A review on multi-label learning algorithms”. In: *TKDE (2013)*.
- [66] Min-Ling Zhang and Zhi-Hua Zhou. “ML-KNN: A lazy learning approach to multi-label learning”. In: *Pattern recognition (2007)*.

Part III

Spatial Representation Learning

*Chapter 6*SPECIES DISTRIBUTION MODELING FOR MACHINE
LEARNING PRACTITIONERS: A REVIEW

- [1] Sara Beery*, Elijah Cole*, Joseph Parker, Pietro Perona, and Kevin Winner. “Species distribution modeling for machine learning practitioners: A review”. In: *ACM SIGCAS conference on computing and sustainable societies*. 2021, pp. 329–348. DOI: [10.48550/arXiv.2107.10400](https://doi.org/10.48550/arXiv.2107.10400).

6.1 Abstract

Conservation science depends on an accurate understanding of what’s happening in a given ecosystem. How many species live there? What is the makeup of the population? How is that changing over time? Species Distribution Modeling (SDM) seeks to predict the spatial (and sometimes temporal) patterns of *species occurrence*, i.e. where a species is likely to be found. The last few years have seen a surge of interest in applying powerful machine learning tools to challenging problems in ecology [2, 89, 55]. Despite its considerable importance, SDM has received relatively little attention from the computer science community. Our goal in this work is to provide computer scientists with the necessary background to read the SDM literature and develop ecologically useful ML-based SDM algorithms. In particular, we introduce key SDM concepts and terminology, review standard models, discuss data availability, and highlight technical challenges and pitfalls.

6.2 Introduction

Ecological research helps us to understand ecosystems and how they respond to climate change, human activity, and conservation policies. Much of this work starts by deploying networks of sensors (often cameras or microphones) to monitor the organisms living in a fixed study area. Ecologists must then invest significant effort to filter, label, and analyze this data. This step is often a bottleneck for ecological research. For example, it can take years for scientists to process and interpret a single season of data from a network of camera traps. In another case, building real-time estimates of salmonid escapement requires teams of field ecologists working in shifts to watch streams of sonar data 24 hours a day. The challenge is even greater for taxa that are studied by trapping specimens, such as beetles and other insects.

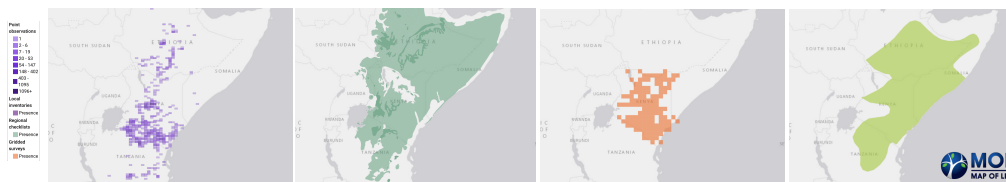


Figure 6.1: Species distribution models describe the relationship between environmental conditions and (actual or potential) species presence. However, the link between the environment and species distribution data can be complex, particularly since distributional data comes in many different forms. Above are four different sources of distribution data for the *Von Der Decken's Hornbill* [113]: (from left to right) raw point observations, regional checklists, gridded ecological surveys, and data-driven expert range maps. All images are from Map of Life [91].

Data collection method	Example	Observation type
Community science observations	iNaturalist	Presence-only
Community science checklists	eBird	Presence-absence
Static sensors	Camera traps	Presence-absence
Sample collection	Insect trapping	Presence-absence
Expert field surveys	Line transects	Presence-absence
Historic records, natural history collections	Herbarium sheets	Presence-only

Table 6.1: Sources of species observation data. Each of these examples represents a method of collecting or accessing observations of different species. One important distinction is whether the observations are *presence-only* or *presence-absence*. Presence-only data consists of locations where a species has been sighted. Presence-absence data also includes locations where a species was checked for but not observed.

Entomologists can collect thousands of beetles in a few days, but it may require months or years for a suitable expert to exhaustively identify all of the specimens to the species level.

Machine learning methods can significantly accelerate the processing and analysis of large repositories of raw data [90, 201, 106, 18, 177], which can increase the speed and geographic scope of ecological analysis. For instance, ongoing collaborations between machine learning researchers and ecologists have led to tremendous progress in automating species identification from images in community science data [190, 109] and camera trap data [13, 201]. However, unfamiliar ecological concepts and terminology can present a barrier to entry for many computer scientists who might otherwise be interested in contributing to ecological problems. This is particularly true for more involved ecological problems which may not fit neatly into existing machine learning paradigms.

One such area is **species distribution modeling** (SDM): using species observations and environmental data to estimate the geographic range of a species.¹ This problem has received significant attention from ecologists and statisticians, and there has been increasing interest in machine learning methods due to the large amounts of available data and the highly complex relationships between species and their environments. This document is meant to serve as an easy entry point for computer scientists interested in SDM. In particular, we aim to highlight the exciting technical challenges posed by SDM while also emphasizing the needs of end-users to encourage ecologically meaningful progress. Our hope is that this document can serve as a quick resource for computer science researchers interested in getting started working on conservation and sustainability applications.

The rest of this work is organized as follows. In Section 6.3 we discuss different ways to represent the distribution of a species. We discuss species distribution modeling in Section 6.4 and we consider other related ecological modeling problems in Section 6.5. In Section 6.6 we point out pitfalls and challenges in SDM. Finally, we provide pointers to available data (Section 6.7) and discuss open problems (Section 6.8).

6.3 Representing the distribution of species

The distribution of a species is typically represented as a *map* which indicates the spatial extent of the species. These maps can be created in a variety of ways, ranging from highly labor-intensive expert range maps to fully automatic species distribution models. We show four examples in Fig. 6.1. In this section we give a high-level overview of three important sources of maps: raw species observation data, predictions from statistical models, and expert knowledge.

Raw species observation data.

Any representation of the distribution of a species begins with some sort of *species observation data*. In general, species observation data consists of records indicating whether a species is present or absent at certain locations. Species observation data can take many forms; see Table 6.1 for examples. Species observation data falls into two general categories: **presence-only** data reports known sightings, or occurrences, of a species, while **presence-absence** data also provides information on where a species did not occur. Data collection strategies define whether absence data will be available. For instance, iNaturalist collects opportunistic imagery of species from

¹We will use the term "species distribution modeling" throughout this document, though sometimes the closely related term "ecological niche modeling" would be more appropriate [142].

community scientists, which produces presence-only species observations. On the other hand, eBird uses species *checklists* where *all* bird species seen and/or heard within a time span at a given location are reported. Since exhaustive reporting is expected from observers, any bird species not reported is assumed to be absent. In this sense, checklists are treated as presence-absence data.

One of the simplest ways to convey the distribution of a species is to simply show all of the locations where the species is known to be present or absent on a map. However, this sort of highly simplified "species distribution" is not able to make any predictions about whether a species might be present or absent at locations which have not been sampled.

Statistical models.

To create species distributions that can extrapolate beyond sampled locations, we can pair species observations with collections of environmental characteristics (altitude, land cover, humidity, temperature, etc.) and fit statistical models that use the environmental characteristics to predict species presence or absence. These models can make predictions at any place and time for which these environmental characteristics are known. Species distribution models fall into this category, and are our focus throughout this document.

Expert range maps.

Species range maps have traditionally been heavily influenced by the individual scientists who study those species. These maps are often based on a complex combination of heterogeneous information sources, including personal observations, understanding of the species' habitat preferences, local knowledge/reports, etc. From our discussions with practitioners, we find that these *expert range maps* (ERMs) are often the most trusted source of distribution information. Perhaps the most widely-known expert range maps are those published by IUCN [70] as part of their *Red List* of vulnerable and endangered species. An example of the IUCN range map for the *caracal* can be seen in Fig. 6.2. Studies have shown both agreement [3] and disagreement [88, 66] between ERMs and species observation data. Expert range maps have also been found to be highly scale-dependent, tending to overestimate the occupancy area of individual species and ranges < 200km [87]. It is important to note that ERMs come in many forms, from hand-drawn maps to data-driven maps that are slightly refined by experts. In the latter case, ERMs are partially based on species observation data, so the two cannot be treated as indepen-



Figure 6.2: The International Union for Conservation of Nature (IUCN) publishes expert range maps for many species, particularly those on their "Red List of Threatened Species" [195]. Here we show the IUCN Range Map for the *Caracal caracal* [7].

dent sources. As we will discuss in more detail in Section 6.4, the lack of a solid "ground truth" information about the true underlying distribution of species across space and time makes it difficult to analyze the accuracy of any species distribution model, including those drawn by experts.

6.4 Species Distribution Models

The terminology in this area can be confusing, so we will start with a definition and a few clarifications.

Intuitive definition. A species distribution model is a function that uses the characteristics of a location to predict whether or not a species is present at that location. This can be understood as a supervised learning problem. The input is a vector of environmental characteristics for a location and the output is species presence or absence. In principle one could use almost any classification or regression technique as the basis for an SDM.

Formal definition. The key components of a simple species distribution modeling pipeline are: (1) species observation data, (2) a method for encoding locations, and (3) a function which maps location encodings to predictions. Formally, we define these components as follows:

1. A dataset of species observations. This is a collection of records indicating

that a species is present or absent at given location and time. We write this as $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathcal{X}$ is a spatiotemporal location and $y_i \in \{0, 1\}$ indicates presence (1) or absence (0). The spatiotemporal domain \mathcal{X} is typically something like $\mathcal{X} = [0, 180) \times [0, 360) \times [0, 1)$ which encodes global longitude and latitude as well as the time of year.

2. A location representation $h : \mathcal{X} \rightarrow \mathcal{Z} \subset \mathbb{R}^k$. This is typically a simple "look-up" operation, where $\mathbf{x} \in \mathcal{X}$ is cross-referenced with k pre-defined geospatial data layers to produce a vector of location features $h(\mathbf{x}) \in \mathbb{R}^k$. That is, $h(\mathbf{x})$ is a representation of the location $\mathbf{x} \in \mathcal{X}$ in some environmental feature space.
3. A model $f_\theta : \mathcal{Z} \rightarrow [0, 1]$ where θ is a parameter vector. The goal is to find parameters θ of f so that $f_\theta(h(\mathbf{x})) = 1$ when the species is present and $f_\theta(h(\mathbf{x})) = 0$ otherwise. This is usually framed as a supervised learning problem on the dataset $\{(h(\mathbf{x}_i), y_i)\}_{i=1}^N$.

Note that this is a streamlined formalization meant to capture the essence of SDM. While there are many variants in practice, almost any species distribution modeling will include these core concepts.

What does an SDM actually predict? An SDM takes as input a vector of environmental features and predicts a numerical score (usually between 0 and 1) for a location. An important distinction to note regarding SDMs is *geographic space* vs. *environmental space*, elucidated in Fig. 6.3. This score is often interpreted as a prediction of habitat suitability. Typically the score *may not* be interpreted as the probability a species is present. Note that here we are only considering presence vs. absence; predicting species *abundance* is a more challenging problem, which we discuss in Section 6.5.

How is an SDM used? The most common end product is a map of the SDM predictions, which is produced by simply visualizing the SDM predictions across an area of interest. Binary predictions can be obtained by applying a threshold to the continuous predictions of the SDM.

A brief history of species distribution modeling

Early predecessors for SDM include qualitative works that link patterns within taxonomic groups to environmental or geographic factors, such as Joseph Grinnel's 1904 study of the distribution of the chestnut-backed chickadee [69], among others [124, 160, 198, 110].

Modern SDMs are primarily statistical models fit to observed data. Early quantitative approaches used multiple linear regression and linear discriminant function analyses to associate species and habitat [26, 168]. The application of generalized linear models (GLMs) [128, 5] provided more flexibility by allowing non-normal error distributions, additive terms, and nonlinear relationships. The explosive proliferation of large "presence-only" datasets (see Table 6.1) in recent years has led to the development of new modeling approaches to SDMs such as the popular "Maximum Entropy Modeling" (MaxEnt) approach [144] with roots in point process modeling [152].

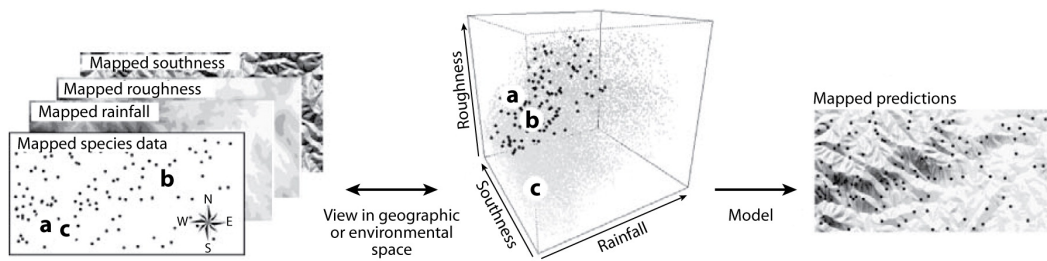
The first modern SDM computing package, BIOCLIM, was introduced in 1984 on the CSIRO network [25, 20]. This package took observation information, such as the species observed, location, elevation, and time, and used them to determine what environmental variables correlated with that species' occurrence. These variables were then used to map possible distributions of the species under consideration. Climate interpolation techniques developed for BIOCLIM are the basis of the existing WorldClim database [53] and are still widely used in SDMs today. Many different implementations of various SDM methods are now publicly available. We would like to highlight Wallace [98], which is a well-documented R implementation of historic and modern techniques.

As earth observation technology has improved, the scope of what is possible to include as an environmental covariate in a model has vastly increased. Improvements in weather monitoring systems gave access to high-temporal-frequency temperature, wind, and precipitation measurements. Recently, ecologists have turned to remote sensing imagery to estimate high-spatial-coverage ecological variables such as soil composition or density of sequestered carbon, as well as mapping land cover type across regions [80]. Modern SDM methods pair these covariate estimates with increasingly accurate global elevation maps, and selected high-quality but sparse in-situ measurements [150, 103].

Several excellent, detailed reviews of SDMs have been published within the ecology community [47, 75, 168, 74, 153, 163]. We direct the reader to the excellent summary by Elith and Leathwick [47].

Covariates for species distribution modeling

In this section we discuss several environmental characteristics (often called *covariates*) that can be used for species distribution modeling. Here we are focused



R Elith J, Leathwick JR. 2009. Annu. Rev. Ecol. Evol. Syst. 40:677–97

Figure 6.3: **Geographic vs. environmental space.** Observation data can be associated with a geographical location, or mapped into a feature space based on environmental covariates. Most SDMs operate under the assumption that with the right set of *environmental variables* and an appropriate model, one could use environmental characteristics to map species distribution. Figure reproduced with permission, originally published in [47].

on describing the different categories of covariates; details on specific covariate datasets are available in Section 6.7. Some of the covariates we discuss are widely used in the species distribution modeling literature, while others are more recent or speculative. It is also important to keep in mind that many covariates are themselves based on sophisticated predictive models due to the cost of densely sampling any property of the earth’s surface.

Climatic variables.

Temperature and precipitation are critical characteristics of an ecosystem. Perhaps the most commonly used climate dataset for SDM is the WorldClim bioclimatic variables [53] dataset, which provides 19 climate-related variables averaged over the period from 1970 to 2000 at a spatial resolution of around 1km². We show a few examples of variables from this dataset in the top row of Fig. 6.5.

Pedologic (soil) variables.

Soil characteristics are intimately related to the plant life in an area, which naturally influences the entire ecosystem. One example of a comprehensive pedologic dataset is SoilGrids250m [83], which consists of soil properties like pH, density, and organic carbon content at a 250m² resolution globally. We show a few examples of variables from this dataset in the bottom row of Fig. 6.5.

Vegetation indices.

A *vegetation index* (VI) is a number used to measure something about the plant life in an area, and is typically computed from remote sensing data like satellite imagery. Many different VIs have been proposed. A review paper published in 1995 discussed 40 different vegetation indices that had been developed by different researchers [9]. One of the most popular examples is the *normalized difference vegetation index* (NDVI). If a remote sensing image includes the red and near-infrared (NIR) bands, then the corresponding NDVI image can be computed by applying the formula

$$\text{NDVI} = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}} \quad (6.1)$$

independently at each pixel. NDVI is meant to indicate the presence of live green plants. From a computer vision perspective, these VIs are essentially hand-designed features for remote sensing data.

Land use / land cover.

The term *land cover* refers to the physical terrain at a location, while the closely related term *land use* tends to emphasize the function of a location. For instance, an area with the land cover label "dense urban" may have a land use label like "school" or "hospital." We provide an example in Fig. 6.4, which shows RGB imagery and land cover from two different sources for the same 1km² area. It is not obvious what the best label set would be for species prediction, but practically speaking many of the available land use / land cover datasets are focused on relatively coarse categories related to agriculture, natural resources, or urban development. For instance, the U.S. National Land Cover Database assigns one of 20 land cover classes to every 30m² patch of land in the United States at a temporal resolution of 2-3 years [85]. The classes cover various general habitat types (water, snow, developed land, forests...) but are not tuned for species prediction in particular.

Measures of human influence.

Humans have had a profound impact on the natural world, so it is reasonable to include measures of human influence as environmental characteristics. For instance, the Human Influence Index [159] uses eight factors (human population density, railroads, roads, navigable rivers, coastlines, nighttime lights, urban footprint, and land cover) to compute a score that is meant to quantify how much an environment has been reshaped by humans.

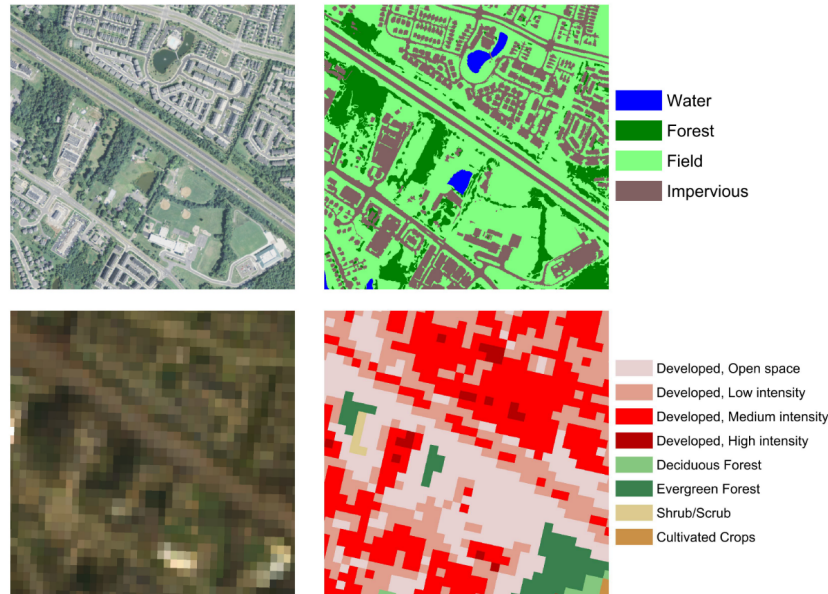


Figure 6.4: RGB imagery (left column) and land cover maps (right column) from two different remote sensing sources covering the same 1km^2 area, from [156]. RGB imagery is manually or semi-automatically annotated to produce the land cover labels. As this example demonstrates, the set of land cover labels can vary depending on the organization doing the labeling. Figure reproduced with permission, originally published in [156].

Remote sensing imagery.

Imagery collected by satellites, planes, or drones can provide substantial information about an environment. To start with, we note that vegetation indices, land cover, land use, and many measures of human influence are all derived from some form of overhead imagery like that in Fig. 6.4. In addition, there may be more abstract patterns that can be extracted using modern computer vision techniques like convolutional neural networks. Research on the use of raw overhead imagery (instead of derived products) for SDM is in its early stages [175, 31, 38].

Properties of species distribution models

In this section we describe important properties that can be used to categorize species distribution models. Any particular species distribution model may or may not have any of these properties. The categories we describe are in general nested or overlapping, not mutually exclusive.

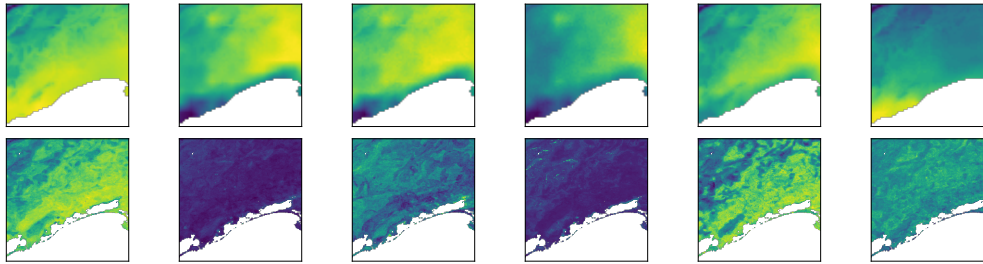


Figure 6.5: Visualizations of some of the bioclimatic variables (top row: `bio_1` - `bio_6` from left to right) and pedologic variables (bottom row: `orcdrc`, `phihox`, `cecsol`, `bdticm`, `clyppt`, `sltppt` from left to right) provided for the GeoLifeCLEF 2020 competition [31]. The area shown in each image is approximately 64 km² centered in Montpellier, France. While we visualize each environmental variable as a 2D raster, most species distribution modeling methods are only compatible with relatively low-dimensional vectors of environmental variables (not "stacks" of 2D patches). As is typical in a collection of covariates, we see that the pedologic variables have a different resolution than the bioclimatic variables.

Presence only vs. presence-absence models.

Species observation datasets may be either presence-absence or presence-only. While presence-only data is easier to collect, there are limitations on what can be estimated from such data [79]. Typically a species distribution model is designed to handle either presence-absence or presence-only data, though there is growing interest in developing methods that can use both [65, 138, 58].

Single vs. multi-species models.

Many SDMs are designed to model the distribution of a single species. This is in contrast to *multi-species* models which are meant to capture information about several species. Many of the earlier models are single-species models [144, 47], though interest in multi-species models has grown over time [86, 78, 131].

Multi-species models: stacked vs. joint

Multi-species SDMs can be classified as either *stacked* or *joint*. In a *stacked* model, a single-species SDM is fit for each species and the resulting maps are "stacked" on top of one another to provide a multi-species map. This approach is simple, but it cannot take advantage of patterns in how species co-occur. This is the motivation for *joint* SDMs, in which the estimated distribution of each species also depends on occurrence data for other species. Recent work has begun to systematically compare

the results from stacked and joint species distribution models for different species and regions [82, 131, 207].

Spatially explicit models.

Typically species distribution models use environmental characteristics to make predictions about the presence or absence of species. Such models represent a location in terms of these environmental features, so two different locations with the same environmental characteristics will lead to the same predictions, even though the two locations may be far apart. Models that mitigate this concern by incorporating geographical location information directly are referred to as *spatially explicit* [40] models.

Occupancy models.

It is easier to confirm that a species is present than it is to confirm that a species is absent. One confident observation of a species suffices to confirm its presence at a given location. However, failing to observe a species at a location does not suffice to prove absence, since the species could have been present but not observed. *Occupancy models* are meant to account for imperfect detection by modeling the probability that a species is present but unobserved at a given location conditional on the sampling effort that has been invested [111, 8].

Understanding uncertainty and error.

Species distribution models attempt to capture the behavior of a complex system from data, which is a challenging and error-prone process. [157] describes 11 sources of uncertainty and error in species distribution models, and groups them into two clusters: (i) uncertainty in the observation data itself and (ii) uncertainty due to arbitrary modeling choices. [41] studies the effect of making different reasonable modeling choices on final projections of species distribution under different future climate scenarios. Similarly, [172] considers the uncertainty introduced by the arbitrary choice of covariates while [167] analyzes the effect of uncertainty in the values of the covariates themselves. [126] focuses on the effect of uncertainty in the location of species observations. [11] reviews sources of uncertainty for different types of species distribution models, as well as best practices for minimizing uncertainty and methods for incorporating uncertainty directly into the model.

Algorithms for species distribution modeling

In this section we provide a high-level overview of the space of algorithms commonly used for species distribution modeling in the ecological community. This section draws heavily from the organization of [131], which is an excellent comparative study of different species distribution modeling techniques. We discuss several commonly used models, and note that the different methods can have very different properties, assumptions, and use cases. Unlike some classes of algorithms, different species distribution modeling methods are generally not readily interchangeable.

Presence-only methods.

Perhaps the most popular approach for presence-only SDM is *MaxEnt* [144]. We follow the description given in [49]. The basic idea is to estimate the probability of observing a given species as a function of the environmental covariates. The estimate is chosen to be (i) consistent with the available species observation data and (ii) as close as possible (in KL divergence) to the marginal distribution of the covariates. Criterion (ii) is necessary because there are typically many distributions that satisfy criterion (i). Another simple approach for presence-only SDM is to introduce artificial negative observations called *pseudonegatives* or *pseudoabsences* based on some combination of domain knowledge and data. Once pseudonegatives have been generated, they are combined with the presence-only data and traditional presence/absence methods are applied.

Traditional statistical methods.

Perhaps the most common methods in species distribution modeling are workhorse methods drawn from the statistics literature such as generalized linear models [61, 59, 196, 192, 135]. Important special cases include logistic regression [140] and generalized additive models [205]. Some species distribution modeling algorithms are better thought of as general frameworks whose particular realization depends on the available data sources and modeling goals. As an example, the Hierarchical Modeling of Species Communities (HMSC) framework [135] minimally requires species occurrence data with corresponding environmental features. The species occurrences are related to environmental features by a generalized linear model. However, the framework can be extended to incorporate information on species traits, evolutionary history, etc.

Machine learning methods.

The relationship between species and their environment is complex and may not satisfy traditional statistical assumptions such as linear dependence on covariates or i.i.d. sampling. For this reason, machine learning approaches have also enjoyed considerable popularity in the species distribution modeling literature. Examples include boosted regression trees [48], random forests [33], and support vector machines [43]. In addition, neural networks have been used for species distribution modeling since well before the deep learning era [22, 136, 206, 181]. Interest in joint species distribution modeling with neural networks has only grown as deep learning has come to maturity [78]. Convolutional neural networks in particular have created a new opportunity: the ability to extract features from spatial arrays of environmental features [28, 35] instead of using hand-selected environmental feature vectors.

The challenge of evaluation

How can we tell whether a species distribution model is performing well or not? The typical approach in machine learning is to use the model to make predictions on a held-out set of data and compute an appropriate performance metric by comparing the model predictions to ground-truth labels. But what is "ground truth" for a species distribution model?

Notions of Ground Truth

We describe several common approaches to the challenging problem of how to evaluate SDMs in practice. For further detail, [121] provides an excellent discussion of different metrics for evaluating SDMs and the extent to which they are ecologically meaningful.

Compare against presence-absence data. Ideally, for each location, an expert observer would determine whether each species of interest is present or absent at that location. Conducting this kind of survey for a single species in a limited area is expensive, and the survey would need to be repeated periodically to monitor change over time. These exhaustive surveys quickly become extraordinarily expensive as we expand the number of species of interest or the geographic extent of the survey. Even if the resources were available, the observations would have some degree of noise - in particular, confirming that a species is absent from an area can typically only be done up to some degree of certainty. (See the discussion of occupancy

modeling in Section 6.4.) For most species and most locations on earth, this sort of ideal ground truth data is just not available. However, this kind of evaluation is possible for select species and locations at sparse time points. For instance, [50] includes presence-absence data for 226 species from 6 parts of the world collected at various time points.

Compare against presence-only data. Unfortunately, presence-absence data is often unavailable. We describe a few simple methods for comparing predictions against presence-only data along with their shortcomings.

- False negative rate: how often are locations which are known to be positive predicted to be negative? The false negative rate measures whether the model is consistent with the observed positives, but does not assess the model's behavior at other points.
- Top- k classification accuracy: how often is the observed species among the k most likely species under the model? However, there is not an obvious way to choose k . Moreover, for any fixed k it is likely that some locations will have more than k species while others will have fewer.
- Adaptive top- k classification accuracy: this is a variant of the top- k classification accuracy that assumes that the number of species is k on average, while allowing some locations to have more than k species while others may have fewer. See [31] for details. Like standard top- k classification accuracy, choosing k may be difficult.

Note that adaptive top- k and top- k are both metrics for multi-species models, while the false negative rate can be computed for single species models as well.

Compare against community science data. Community science projects like iNaturalist and eBird are generating species observation data at an extraordinary rate and frequency. iNaturalist alone generates millions of species observations per month [1]. However, the data produced by such projects can vary in terms of how easy it is to use and interpret depending on the sampling protocol [102]. For instance, iNaturalist accepts presence-only observations, which allows the user base to scale broadly but limits the utility of the data for ground truthing. iNaturalist data tells us where different species have been observed by humans, but not where those species are either absent or present without human observation. eBird uses a more rigorous sampling protocol that records both presences and absences, but their

observations are limited to birds. The quality of these reports depends on the skill of the user at identifying all bird species they see or hear. Citizen science data has been found to produce results similar to those from (coarse) professional surveys under the right circumstances [84, 184, 102].

Compare against expert range maps. Another possibility is to compare the model predictions against one or more range maps that are hand-drawn by experts (see Section 6.3). However, this raises the question: how do we validate *those* range maps? A hand-drawn map may be biased by an individual's experience or by the data sources the expert prefers. In addition, it can be difficult to find a suitable expert to generate a map for every species of interest. Another challenging question relates to temporal progression: is each expert updating their maps according to the latest data? If so, when was that data collected? The IUCN has a published set of standards for creating species range maps [70], but not all creators of maps match these standards.

In addition, there is the methodological question of how one should evaluate a model against an expert range map, which is explored in [112]. Approaches range from very qualitative (ask an expert whether the map looks reasonable to them) to very quantitative (compute a well-defined error metric between the SDM predictions and the expert range map). Important to note here, expert range maps are most often categorical, with hard boundaries drawn representing temporal categories like "breeding", "non-breeding", "year-round", etc. On the other hand, SDM predictions are often real-valued on $[0, 1]$ over both space and time. While continuous predictions can be converted to binary maps by applying a threshold, it can be unclear how to choose this threshold if a robust validation method is not available.

Evaluation on downstream tasks. Instead of evaluating whether a species distribution model produces a faithful map of species presence, we may instead check whether it is useful for some other downstream task. For example, [109] builds a simple SDM and demonstrates that it improves accuracy on an image-based species classification task. However, it is certainly possible for an SDM to be useful whether or not it accurately reflects the true species distribution.

Evaluation pitfalls

Even when suitable ground truth data is available, there are some pitfalls that can hinder meaningful evaluation. In this section we discuss some of these pitfalls and

make specific recommendations to the machine learning community for handling them.

Performance overestimation due to spatial autocorrelation. In the machine learning community it is common to sample a test set uniformly at random from the available data. However, this strategy can lead to overestimation of algorithm performance for spatial prediction tasks since it is possible to obtain high performance on a uniformly sampled test set by simple interpolation [154]. This effect is called *spatial autocorrelation*. Similar concerns are relevant for evaluating camera trap image classifiers [15]. For ecological tasks, it is important to evaluate models as they are intended to be used. In many cases, the more ecologically meaningful question is whether the model generalizes to novel locations, unseen in the training set. In these cases it is important to create a test set by holding out spatial areas. In other cases, the ecologist seeks to build a model that will perform accurately in the future at their set of monitoring sites. In these cases, instead of holding out data in space, we can split the data to hold out a test set based on time. A randomly sampled test set is not a good proxy for the use case of either scenario.

Hyperparameter selection. The performance of an algorithm typically depends on several hyperparameters. In the machine learning community these are set using cross-validation on held-out data. However, selecting and obtaining a useful validation set can be particularly challenging in SDM due to the data collection challenges described elsewhere. Recent work has also studied the sensitivity of SDMs to hyperparameters [76] and developed techniques for hyperparameter selection in the presence of spatial autocorrelation [162].

Spatial quantization. A natural first step when working with spatially distributed species observations is to define a spatial quantization scheme. By "binning" observations in this way, we can associate many species observations with a single vector of covariates. Additionally, spatially quantized data can be more natural from the perspective of many machine learning algorithms since the domain becomes discrete. However, the choice of quantization scheme (grid cell size) is difficult to motivate in a rigorous way. This is a problem because different quantization choices can result in vastly different outcomes: this is known as the *modifiable areal unit problem* [134]. It is possible to cross-validate the quantization parameters, but only in those limited cases where there is enough high-quality data for this to be a reliable procedure. Furthermore, that process may be computationally expensive.

The long tail. Many real-world datasets exhibit a *long tail*: a few classes represent a

large proportion of the observations, while many classes have very few observations [188, 15]. Species observation data is no exception; for example, in the Snapshot Serengeti camera trap dataset [169] there are fewer than 10 images of gorillas out of millions of images collected over 11 years. This presents at least two problems: the first problem is that standard training procedures will typically result in a model that perform well on the common classes and poorly on the rare classes; the second is that many evaluation metrics are averaged over all examples in the dataset, which means that the metric can be very high despite poor performance on almost all species. It is much more informative to study the performance on each class or on groups of classes (e.g. common classes vs. rare classes). One common solution is to compute metrics separately for each class and then average over all classes to help avoid bias towards common classes in evaluation.

Model trust

Once a model has been built, the previously discussed challenges of model evaluation make it difficult to determine where, how much, and for how long a model is sufficiently accurate to be used. The accuracy needed may also vary by use case and subject species. In our discussions with ecologists, we find that this leads to a lack of trust in SDMs. What verification and quality control is needed to ensure a model is still valid over time? This is an open question, and an important one to answer if our models are to be used in the real world.

6.5 Other types of ecological models

Species distribution modeling is only one of many ways that ecologists seek to describe and understand the natural world. To give readers a sense of how SDM fits into the broader scope of ecological modeling, we provide a high-level overview of other common modeling tasks.

Mechanistic models

Mechanistic models make assumptions about how species depend on the environment or on other species. One example is to use an understanding of a plant's biology to predict the viable temperature range where the plant can grow [170]. Such models are useful but difficult to scale, as they require species-specific expert knowledge. Our focus in this work is on *correlative* species distribution models, which do not require mechanistic knowledge.

Abundance modeling

Abundance modeling goes beyond species presence or absence, aiming to characterize the absolute or relative number of individuals at a given location. We define abundance and related concepts in Section 6.5.

Population estimation

Population estimation is concerned with counting the total number of individuals of a species, typically within some defined area [161]. Population size is most frequently estimated using *capture-recapture models*, which require the ability to distinguish between individuals of the same species. Traditionally this individual re-identification was based on physical tags or collars [67], but some recent efforts have relied on the less invasive method of identifying visually distinctive features, such as stripe patterns or the contour of an ear [18].

Density estimation

Density estimation seeks to model *spatial abundance*, the abundance of a species per unit area, to understand where a species is densely versus sparsely populated [158, 193].

Data collection procedures for abundance

As mentioned above, capture-recapture requires an individual to be re-identifiable. In the absence of the ability to re-identify individuals, several other data collection procedures are used. One that is frequently used for insects and fish populations is the *harvest method*, where individuals are collected in traps which are open for a set amount of time and then counted [148, 164]. Sampling strategies for other taxa include:

- **Quadrat sampling.** A *quadrat* is a fixed-size area where species are to be sampled. Within the quadrat, the observer exhaustively determines the occurrence and relative abundance of the species of interest. Quadrat sampling is most commonly used for stationary species like plants. The observer will sample quadrats throughout the region of interest to derive sample variance and conduct further statistical analysis [77].
- **Line intercept sampling.** A *line intercept* or *line transect* is a straight line that is marked along the ground or the tree canopy, and is primarily used for

stationary species [81]. The observer proceeds along the line and records all of the specimens intercepted by the line. Each transect is regarded as one sample unit, similar to a single quadrat.

- **Cue counting.** Cue counting is based on observing cues or signals that a species is nearby, such as whale or bird calls. It is used primarily for species that are underwater or similarly difficult to sight [114].
- **Distance sampling.** *Distance sampling* refers to a class of methods which estimate the density of a population using measured distances to individuals in the population [23]. Distance sampling can be added to line transects in order to incorporate specimens that are off the transect line but still visible. Appropriately calibrated camera traps can also benefit from distance sampling [158].
- **Environmental DNA (eDNA) sampling.** Samples of water or excrement collected in the field can be sequenced to provide species identifications. The ratios of environmental DNA for each species can be used to estimate abundance [108, 187].

Each of these procedures produces different types of data, and each method comes with its own innate collection biases. These biases can add to the challenge of evaluating ecological models, as discussed in Section 6.4.

Biodiversity measurement and prediction

While it is important to understand the distribution of particular species, in many cases the ultimate goal is to understand the health of an ecosystem at a higher level. *Biodiversity* is a common surrogate for ecosystem health, and there are many different ways to measure it [199, 95, 96]. In this section we define and discuss several biodiversity metrics and related concepts. Note that some sources give different definitions than those presented here, so caution is warranted.

We now define some preliminary notation. We let R denote an arbitrary spatial unit such as a country. Many biodiversity metrics are computed based on a *partition* of R into N sub-units, which we denote by $\{R_i\}_{i=1}^N$. The choice of partition can have a significant impact on the value of some metrics, but for the purposes of this section we simply assume a partition has been provided.

Species richness. The species richness of R is the number of unique species in R , which we write as $S(R)$.

Absolute abundance. The absolute abundance of species k in R is the number of individuals in R who belong to species k . We write this as $A_k(R)$.

Relative abundance. The relative abundance of species k in R is the fraction of individuals in R who belong to species k , which is

$$p_k(R) = \frac{A_k(R)}{\sum_{j=1}^{S(R)} A_j(R)}. \quad (6.2)$$

Since $\sum_{j=1}^{S(R)} p_j(R) = 1$ and $p_j(R) \geq 0$ for all $j \in \{1, \dots, S(R)\}$, the vector of relative abundances $\mathbf{p}(R) = (p_1(R), \dots, p_{S(R)}(R))$ forms a discrete probability distribution. The species richness can then be alternately defined as the support of this distribution, given by

$$S(R) = |\{j \in \{1, \dots, S(R)\} : p_j(R) > 0\}|. \quad (6.3)$$

Of course we can replace p_j with A_j everywhere and get an identical quantity.

Shannon index. The Shannon index of R is the entropy of the probability distribution $\mathbf{p}(R)$, so

$$H(\mathbf{p}(R)) = - \sum_{j=1}^{S(R)} p_j(R) \log p_j(R). \quad (6.4)$$

The Shannon index quantifies the uncertainty involved in guessing the species of an individual chosen at random from R . Sometimes H is instead written as H' , and sometimes the argument is written as R instead of $\mathbf{p}(R)$.

Simpson index. The Simpson index of R is the probability that two individuals drawn at random from the dataset (with replacement) are the same species, and is given by

$$\lambda(R) = \sum_{i=1}^{S(R)} p_i^2. \quad (6.5)$$

Alpha diversity. The alpha diversity of R is the average species richness across the sub-units $\{R_i\}_{i=1}^N$, given by

$$\alpha(R) = \frac{1}{N} \sum_{i=1}^N S(R_i). \quad (6.6)$$

Gamma diversity. The gamma diversity of R is defined as

$$\gamma(R, q) = \left(\sum_{j=1}^{S(R)} p_j^q \right)^{1/(1-q)}, \quad (6.7)$$

where $q \in [0, 1) \cup (1, \infty)$ is a weighting parameter [95]. Note that gamma diversity is also commonly denoted by ${}^\gamma D_q(R)$. There are several interesting special cases:

- If $q = 0$ then gamma diversity reduces to species richness i.e. $\gamma(R, 0) = S(R)$.
- Gamma diversity is also related to the Shannon index, since $\lim_{q \rightarrow 1} \gamma(R, q) = \exp H(\mathbf{p}(R))$ [95].
- If $q = 2$ then gamma diversity reduces to the inverse of the Simpson index i.e. $\gamma(R, 2) = 1/\lambda(R)$.

Beta diversity. The beta diversity of R is meant to measure the extent to which sub-units R_i are ecologically differentiated. This can be interpreted as a measure of the variability of biodiversity across sub-regions or habitats within a larger area. It is defined as

$$\beta(R, q) = \frac{\gamma(R, q)}{\alpha(R)}, \quad (6.8)$$

where q is the same weighting parameter we say in the definition of gamma diversity [183, 95]. Beta diversity quantifies how many sub-units there would be if the total species diversity of the region γ and the mean species diversity per sub-unit α remained the same, but the sub-units had no species in common.

6.6 Common challenges and risks

Differences in tools

R is the dominant coding language in ecology and statistics, but Python is dominant in machine learning. This language barrier limits code sharing, which in turn limits algorithm sharing. It is also important to note that some machine learning models are extremely computationally demanding to train, and some ecologists may not have access to the necessary computational resources.

Differences in ideas and terminology

Differences in concepts and terminology can make it difficult for machine learning practitioners to find and read relevant work from the ecology community (and vice-versa). However, there is a growing body of interdisciplinary work which brings

ecologists and computer scientists together [2, 89, 133]. It is important for computer scientists working in this area to establish ties with ecologists who can help them understand how to make ecologically meaningful progress.

Combining data sources

Species observation data is collected according to many different protocols, which means that effectively combining different data sources can be nontrivial [137, 101, 119, 63]. For instance, observations collected in a well-designed scientific survey have significantly different collection biases from observations collected via iNaturalist. Handling these biases in a robust, systematic way can be quite challenging, particularly for large collections of data encompassing thousands of different projects, each with their own sampling strategies. In many cases, understanding the protocols used for a specific data collection project within a larger repository requires one to delve into the literature for that project. However, for many projects there do not exist accessible, standardized definitions or quantitative analysis of bias.

Black boxes, uncertainty, and interpretability

Machine learning models are frequently "black boxes", meaning that it is difficult to understand how a prediction is being made. Ecologists are accustomed to models that are simpler to inspect and analyze, where they can confidently determine what factors are most important and what the effect of different factors might be. Because the results of ecological models are used to drive policy, being able to interpret how a model is making predictions and avoid inaccuracies due to overfitting is important. This is closely related to trust (or lack thereof) in model outputs and the need for uncertainty quantification, particularly in scenarios where models are being asked to generalize to new locations or forward in time.

Norms surrounding data sharing and open sourcing in ecology

Computer science has benefited from strong community norms promoting public data and open-sourced code. One consequence of this shift is that it is easy for computer scientists to take data for granted and to be frustrated when a scientist is unwilling to share their data publicly. However, it is important to remember that in some fields data can be extremely expensive to collect and curate. The cost of the hardware, travel to the study site, and the time needed to place the sensors and maintain the sensor network quickly adds up. Add to this the number of hours it takes for an expert to process and label the data so that it is ready for analysis, and it is

easy to see why a researcher would want to publish several papers on their hard-won data before sharing it publicly. On the other hand, public datasets like those hosted on LILA.science [106] have clear benefits for the community such as promoting reproducible research. Properly attributing data to the researchers who collected it (e.g. through the use of "DOIs for datasets" [155]) could encourage more open data sharing in ecology. Data sharing norms are changing and many researchers are now happy to share their data and are pushing for more open data practices [149, 151], but it is important to be aware of this cultural difference between computer science and other fields.

Model handoffs, deployment, and accessibility

Once a machine learning method has been rigorously evaluated and found to be helpful, it is important to ensure these techniques are accessible to those who can put them to good use. In computer science, we have a culture of "open code, open data" which means that for most papers, all of the data and code is publicly available. However, ecologists may be less familiar with machine learning packages like PyTorch and TensorFlow, and may not have access to the computational resources required to train models on their data. If a method is to have real impact for the ecology community, it is important to provide models and code in a format that is accessible to end-users and well-documented. If the model is meant to become an integral part of an ecology workflow, plans for model maintenance and upkeep should be discussed.

Sensitive species

It is common for ecologists to obfuscate geolocation information before publishing any data containing rare or protected species to avoid poaching or stress from ecotourism. However, it is unclear whether obfuscation of GPS signal is sufficient to obscure the location of a photograph. It may be that a better solution is to remove any photos containing sensitive species, or to restrict sensitive access to a list of verified members of the research community. Second, the obfuscation distance of GPS location in published datasets might have a large effect on the accuracy of an SDM or other ecological model, particularly when both the training and validation data have been obfuscated. This obfuscation will further effect the reproducibility of a study, as results with or without obfuscation might be quite different.

6.7 What data is available and accessible?

There is an increasing number of publicly available ecological datasets that can be used for model training and evaluation. In this section we provide a few useful data sources as a starting point. We make a distinction between "analysis-ready" datasets which package species observations and covariates together and other data sources which can be combined to produce analysis-ready datasets.

Traditional analysis-ready datasets for multi-species distribution modeling

- The comprehensive SDM comparison in [131] uses five presence-absence datasets covering different species and parts of the world. Each dataset has a different set of covariates (min 6, max 38) and a different set of species (min 50, max 242). The datasets are available for download on Zenodo [129].
- The recently released benchmark dataset [50] covers 226 species from 6 regions. Each region has a different set of covariates (min 11, max 13) and a different set of species (min 32, max 50).

Note that many "traditional" SDM datasets may not be large enough to train some of the more data-hungry machine learning methods.

Large-scale analysis-ready datasets for multi-species distribution modeling

- The GeoLifeCLEF datasets combine 2D patches of covariates with species observations from community science programs. The GeoLifeCLEF 2020 dataset [31] consists of 1.9M observations of 31k plant and animal species from France and the US, each of which is paired with high-resolution 2D covariates (satellite imagery, land cover, and altitude) in addition to traditional covariates. Previous editions of the GeoLifeCLEF dataset [36, 21] are also available, and are suitable for large-scale plant-focused species distribution modeling in France using traditional covariates. Note that all of the GeoLifeCLEF datasets are based on presence-only observations, so performance is typically evaluated using information retrieval metrics such as top- k accuracy.
- The eBird Reference Dataset (ERD) [123] is built around checklists collected by eBird community members. In particular, it is limited to checklists for which the observer (i) asserts that they reported everything they saw and (ii) quantified their sampling effort. This allows unobserved species to be interpreted as absences if sufficient sampling effort has been expended. The

resulting presence/absence data is combined with land cover and climate variables. Unfortunately, the ERD does not appear to be maintained or publicly available as of November 2020.

Sources for species observation data

- The Global Biodiversity Information Facility (GBIF) [177] aggregates and organizes species observation data from over 1700 institutions around the world. We discuss a few specific contributors below.
- iNaturalist [90] is a community science project that has produced over 70 million point observations of species across the entire taxonomic tree. The data can be noisy as it is collected and labeled by non-experts.
- eBird [45] is a community science project hosted by the Cornell Lab of Ornithology which has produced more than 77 million birding checklists. These checklists provide both presence and absence, but absences can be noisy as it is possible the birder did not observe every species that was present at a given location.
- Movebank [122] is a database of animal tracking data hosted by the Max Planck Institute of Animal Behavior. It contains GPS tracking data for individual animals, covering 900 taxa and including 2.2 billion unique location readings.

Sources for covariates

Earth observation datasets and their derived products can be freely obtained from many sources, including the NASA Open Data Portal [127], the USGS Land Processes Distributed Access Data Archive [185], ESA Earth Online [51], and Google Earth Engine [64]. Also see the detailed discussion of covariates in Section 6.4.

Sources for training species identification models

Species observation data can be produced by classifying the species found in geolocated images. Those who are interested in the species classification problem may be interested in the datasets below.

- The iNaturalist species classification datasets [190, 189] are curated species classification datasets built from research-grade observations in iNaturalist.

- LILA.science [106, 15, 132] hosts a number of biology-focused image classification datasets, including camera trap datasets covering diverse species and locations.
- The Fine-Grained Visual Categorization (FGVC) workshop [55] at CVPR hosts a number of competitions each year [55, 16, 14, 12, 17, 190, 176, 174, 125] which focus on species classification and related biodiversity tasks.

6.8 Open Problems

There are many open problems in SDM that may benefit from machine learning tools. In this section we discuss a few of these problems which we find particularly interesting.

Scaling up, geospatially and taxonomically

One of the main challenges in modern SDMs is scale. This includes scaling up SDMs to efficiently handle large geographic regions [179, 99, 92], many-species communities [130, 202, 145, 180], and large volumes of training data [117, 203, 180]. One particularly interesting question is whether jointly modeling many species could lead to SDMs which are significantly better than those based on modeling species independently.

Incorporating ecological theory and expert knowledge

There is a considerably amount of domain knowledge and ecological theory which would ideally be incorporated into SDMs [73]. This might include knowledge about species dispersal [60, 10, 120, 37], spatial patterns of community composition [34, 29, 94], and constraints on species ranges (e.g. cliffs, water) [57, 52, 120, 32]. Another area of significant interest is to factor in cross-species biological processes such as niche exclusion/competition [200, 146], predator/prey dynamics [182, 42, 146], phylogenetic niche evolution [141, 62, 27], or models linked across functional traits [147, 30, 194]. These types of "domain-aware" algorithms are an active research area in the machine learning community [19, 71, 171, 39].

Fusing data

A third open area of investigation centers on how to best incorporate and utilize data collected at different spatiotemporal scales or in heterogeneous formats. This includes combining presence-only, presence-absence, abundance, and individual data such as GPS telemetry data [93, 143, 139, 54]. It also includes multi-scale

or cross-scale modeling [186, 173], such as microclimate niche vs. macroscale niche [104], individual niche variance vs. species level niche variance[54], and cross-scale ecological processes[72, 115]. Finally, it may also include models of temporal ecological processes, such as seasonal range shifts and migrations [178, 166].

Evaluation

How should we compare competing models and decide which models to trust? Naturally, fair head-to-head evaluation of different models will be important [4, 46, 131]. Future large-scale evaluations may require accounting for biases in species observation data [191, 197, 107, 56], especially that which comes from community science projects. However, it is important to keep in mind that there is no single metric which makes one SDM better than another. It may be important to understand how a model's predictions change under novel climate scenarios [57, 24, 6, 105] or different conservation policies [165, 116, 44] or how well-calibrated the SDM predictions are [4, 68]. One promising avenue is to study models in increasing realistic simulation environments [204, 97, 118], which allows for more comprehensive analysis. Many of these topics are directly related to active areas of machine learning research, such as generalization, domain adaptation, and overcoming dataset bias and imbalance [100].

6.9 Conclusion

We have sought to introduce machine learning researchers to a challenging and important real-world problem domain. We have discussed common terminology and highlighted common pitfalls and challenges. To lower the initial overhead, we have inventoried some available datasets and common methods. We hope that this document is useful for any computer scientist interested in bringing machine learning expertise to species distribution modeling.

6.10 Acknowledgements

Our research for this paper included informational interviews with Meredith Palmer, Michael Tabak, Corrie Moreau, and Carrie Seltzer. Their insights into the unique challenges of species distribution modeling was invaluable. This work was supported in part by the Caltech Resnick Sustainability Institute and NSFGRFP Grant No. 1745301. The views expressed in this work are those of the authors and do not necessarily reflect the views of the NSF.

References

- [1] *50 million observations on iNaturalist!* <https://www.inaturalist.org/blog/40699-50-million-observations-on-inaturalist>.
- [2] *AI for Animal Re-Identification Workshop at WACV 2020*. <https://sites.google.com/corp/view/wacv2020animalreid/>.
- [3] Bader H Alhajeri and Yoan Fourcade. “High correlation between species-level environmental data estimates extracted from IUCN expert range maps and from GBIF occurrence data”. In: *Journal of Biogeography* 46.7 (2019), pp. 1329–1341.
- [4] Miguel B. Araújo and Antoine Guisan. “Five (or so) Challenges for Species Distribution Modelling”. en. In: *Journal of Biogeography* 33.10 (2006), pp. 1677–1688. ISSN: 1365-2699. DOI: 10.1111/j.1365-2699.2006.01584.x.
- [5] Michael Phillip Austin. “Continuum concept, ordination methods, and niche theory”. In: *Annual review of ecology and systematics* 16.1 (1985), pp. 39–61.
- [6] Mike P. Austin and Kimberly P. Van Niel. “Improving Species Distribution Models for Climate Change Studies: Variable Selection and Scale”. en. In: *Journal of Biogeography* 38.1 (2011), pp. 1–8. ISSN: 1365-2699. DOI: 10.1111/j.1365-2699.2010.02416.x.
- [7] Laila Bahaa-El-Din et al. *Caracal aurata*. *The IUCN Red List of Threatened Species 2015*. Apr. 2015.
- [8] Larissa L Bailey, Darryl I MacKenzie, and James D Nichols. “Advances and applications of occupancy models”. In: *Methods in Ecology and Evolution* 5.12 (2014), pp. 1269–1279.
- [9] A Bannari et al. “A review of vegetation indices”. In: *Remote sensing reviews* 13.1-2 (1995), pp. 95–120.
- [10] Narayani Barve et al. “The Crucial Role of the Accessible Area in Ecological Niche Modeling and Species Distribution Modeling”. en. In: *Ecological Modelling* 222.11 (June 2011), pp. 1810–1819. ISSN: 0304-3800. DOI: 10.1016/j.ecolmodel.2011.02.011.
- [11] Colin M Beale and Jack J Lennon. “Incorporating uncertainty in predictive species distribution modelling”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 367.1586 (2012), pp. 247–258.
- [12] Sara Beery, Elijah Cole, and Arvi Gjoka. “The iWildCam 2020 Competition Dataset”. In: *CVPR Workshop on Fine-Grained Visual Categorization* (2020). DOI: 10.48550/arXiv.2004.10340.

- [13] Sara Beery and Dan Morris. “Efficient Pipeline for Automating Species ID in new Camera Trap Projects”. In: *Biodiversity Information Science and Standards* 3 (2019), e37222.
- [14] Sara Beery, Dan Morris, and Pietro Perona. “The iWildCam 2019 Challenge Dataset”. In: *The Sixth Fine-Grained Visual Categorization Workshop at CVPR* (2019).
- [15] Sara Beery, Grant Van Horn, and Pietro Perona. “Recognition in terra incognita”. In: *ECCV*. 2018.
- [16] Sara Beery et al. “The iWildCam 2018 Challenge Dataset”. In: *The Fifth Fine-Grained Visual Categorization Workshop at CVPR* (2019).
- [17] Sara Beery et al. “The iWildCam 2021 Competition Dataset”. In: *The Eighth Fine-Grained Visual Categorization Workshop at CVPR* (2021).
- [18] Tanya Y Berger-Wolf et al. “Wildbook: Crowdsourcing, computer vision, and data science for conservation”. In: *arXiv preprint arXiv:1710.08880* (2017).
- [19] Christopher M Bishop. “Model-based machine learning”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1984 (2013), p. 20120222.
- [20] Trevor H Booth et al. “BIOCLIM: the first species distribution modelling package, its early applications and relevance to most current MAXENT studies”. In: *Diversity and Distributions* 20.1 (2014), pp. 1–9.
- [21] Christophe Botella et al. “Overview of GeoLifeCLEF 2019: plant species prediction using environment and animal occurrences”. In: 2019.
- [22] David S Broomhead and David Lowe. *Radial basis functions, multi-variable functional interpolation and adaptive networks*. Tech. rep. Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.
- [23] Stephen T Buckland et al. “Distance sampling”. In: *Encyclopedia of biostatistics* 2 (2005).
- [24] Laëtitia Buisson et al. “Uncertainty in Ensemble Forecasting of Species Distribution”. en. In: *Global Change Biology* 16.4 (2010), pp. 1145–1157. ISSN: 1365-2486. DOI: 10.1111/j.1365-2486.2009.02000.x.
- [25] J_R Busby. “BIOCLIM—a bioclimate analysis and prediction system”. In: *Plant protection quarterly* 61 (1991), pp. 8–9.
- [26] David E Capen. *The use of multivariate statistics in studies of wildlife habitat*. Vol. 87. Rocky Mountain Forest and Range Experiment Station, Forest Service, US . . . , 1981.
- [27] Daniel S. Chapman et al. “Mechanistic Species Distribution Modeling Reveals a Niche Shift during Invasion”. en. In: *Ecology* 98.6 (2017), pp. 1671–1680. ISSN: 1939-9170. DOI: 10.1002/ecy.1835.

- [28] Di Chen et al. “Deep multi-species embedding”. In: *arXiv preprint arXiv:1609.09353* (2016).
- [29] Di Chen et al. “Deep multi-species embedding”. In: *IJCAI*. 2017.
- [30] James S. Clark et al. “Generalized Joint Attribute Modeling for Biodiversity Analysis: Median-Zero, Multivariate, Multifarious Data”. en. In: *Ecological Monographs* 87.1 (2017), pp. 34–56. ISSN: 1557-7015. DOI: 10.1002/ecm.1241.
- [31] Elijah Cole, Benjamin Deneu, Titouan Lorieul, Maximilien Servajean, Christophe Botella, Dan Morris, Nebojsa Jojic, Pierre Bonnet, and Alexis Joly. “The geolifeclef 2020 dataset”. In: *arXiv preprint arXiv:2004.04192* (2020). DOI: 10.48550/arXiv.2004.04192.
- [32] Jacob C. Cooper and Jorge Soberón. “Creating Individual Accessible Area Hypotheses Improves Stacked Species Distribution Model Performance”. en. In: *Global Ecology and Biogeography* 27.1 (2018), pp. 156–165. ISSN: 1466-8238. DOI: 10.1111/geb.12678.
- [33] D Richard Cutler et al. “Random forests for classification in ecology”. In: *Ecology* 88.11 (2007), pp. 2783–2792.
- [34] Manuela D’Amen, Jean-Nicolas Pradervand, and Antoine Guisan. “Predicting Richness and Composition in Mountain Insect Communities at High Resolution: A New Test of the SESAM Framework”. en. In: *Global Ecology and Biogeography* 24.12 (2015), pp. 1443–1453. ISSN: 1466-8238. DOI: 10.1111/geb.12357.
- [35] Benjamin Deneu et al. “Evaluation of deep species distribution models using environment and co-occurrences”. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer. 2019, pp. 213–225.
- [36] Benjamin Deneu et al. “Location-based species recommendation using co-occurrences and environment-GeoLifeCLEF 2018 challenge”. In: 2018.
- [37] Michele Di Musciano et al. “Dispersal Ability of Threatened Species Affects Future Distributions”. en. In: *Plant Ecology* 221.4 (Apr. 2020), pp. 265–281. ISSN: 1573-5052. DOI: 10.1007/s11258-020-01009-0.
- [38] Solomon Z Dobrowski et al. “Mapping mountain vegetation using species distribution modeling, image-based texture analysis, and object-based classification”. In: *Applied Vegetation Science* 11.4 (2008), pp. 499–508.
- [39] Bradley B Doll, Dylan A Simon, and Nathaniel D Daw. “The ubiquity of model-based reinforcement learning”. In: *Current opinion in neurobiology* 22.6 (2012), pp. 1075–1081.
- [40] Sami Domisch et al. “Spatially explicit species distribution models: A missed opportunity in conservation planning?” In: *Diversity and Distributions* 25.5 (2019), pp. 758–769.

- [41] Carsten F Dormann et al. “Components of uncertainty in species distribution analysis: a case study of the great grey shrike”. In: *Ecology* 89.12 (2008), pp. 3371–3386.
- [42] Carsten F. Dormann et al. “Biotic Interactions in Species Distribution Modelling: 10 Questions to Guide Interpretation and Avoid False Conclusions”. en. In: *Global Ecology and Biogeography* 27.9 (2018), pp. 1004–1016. ISSN: 1466-8238. DOI: 10.1111/geb.12759.
- [43] John M Drake, Christophe Randin, and Antoine Guisan. “Modelling ecological niches with support vector machines”. In: *Journal of applied ecology* 43.3 (2006), pp. 424–432.
- [44] Sally Eaton et al. “Adding Small Species to the Big Picture: Species Distribution Modelling in an Age of Landscape Scale Conservation”. en. In: *Biological Conservation* 217 (Jan. 2018), pp. 251–258. ISSN: 0006-3207. DOI: 10.1016/j.biocon.2017.11.012.
- [45] *eBird*. <https://ebird.org/home>.
- [46] Jane Elith and Catherine H. Graham. “Do They? How Do They? Why Do They Differ? On Finding Reasons for Differing Performances of Species Distribution Models”. In: *Ecography* 32.1 (2009), pp. 66–77. ISSN: 0906-7590.
- [47] Jane Elith and John R Leathwick. “Species distribution models: ecological explanation and prediction across space and time”. In: *Annual review of ecology, evolution, and systematics* 40 (2009), pp. 677–697.
- [48] Jane Elith, John R Leathwick, and Trevor Hastie. “A working guide to boosted regression trees”. In: *Journal of Animal Ecology* 77.4 (2008), pp. 802–813.
- [49] Jane Elith et al. “A statistical explanation of MaxEnt for ecologists”. In: *Diversity and distributions* 17.1 (2011), pp. 43–57.
- [50] Jane Elith et al. “Presence-only and Presence-absence Data for Comparing Species Distribution Modeling Methods”. In: *Biodiversity Informatics* 15.2 (2020), pp. 69–80.
- [51] *ESA Earth Online*. <https://www.earth.esa.int/>.
- [52] Robert M. Ewers, Charles J. Marsh, and Oliver R. Wearn. “Making Statistics Biologically Relevant in Fragmented Landscapes”. en. In: *Trends in Ecology & Evolution* 25.12 (Dec. 2010), pp. 699–704. ISSN: 0169-5347. DOI: 10.1016/j.tree.2010.09.008.
- [53] Stephen E Fick and Robert J Hijmans. “WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas”. In: *International journal of climatology* 37.12 (2017), pp. 4302–4315.

- [54] John R. Fieberg et al. “Used-Habitat Calibration Plots: A New Procedure for Validating Species Distribution, Resource Selection, and Step-Selection Models”. en. In: *Ecography* 41.5 (2018), pp. 737–752. ISSN: 1600-0587. DOI: 10.1111/ecog.03123.
- [55] *Fine Grained Visual Categorization Workshop (FGVC) at CVPR*. <http://www.fgvc.org/>.
- [56] William Fithian et al. “Bias Correction in Species Distribution Models: Pooling Survey and Collection Data for Multiple Species”. en. In: *Methods in Ecology and Evolution* 6.4 (2015), pp. 424–438. ISSN: 2041-210X. DOI: 10.1111/2041-210X.12242.
- [57] Matthew C. Fitzpatrick and William W. Hargrove. “The Projection of Species Distribution Models and the Problem of Non-Analog Climate”. en. In: *Biodiversity and Conservation* 18.8 (Apr. 2009), p. 2255. ISSN: 1572-9710. DOI: 10.1007/s10531-009-9584-8.
- [58] Robert J Fletcher Jr et al. “A practical guide for combining data to model species distributions”. In: *Ecology* 100.6 (2019), e02710.
- [59] Scott D Foster and Piers K Dunstan. “The analysis of biodiversity using rank abundance distributions”. In: *Biometrics* 66.1 (2010), pp. 186–195.
- [60] Janet Franklin. “Moving beyond Static Species Distribution Models in Support of Conservation Biogeography”. en. In: *Diversity and Distributions* 16.3 (2010), pp. 321–330. ISSN: 1472-4642. DOI: 10.1111/j.1472-4642.2010.00641.x.
- [61] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. “Regularization paths for generalized linear models via coordinate descent”. In: *Journal of statistical software* 33.1 (2010), p. 1.
- [62] Daniel G. Gavin et al. “Climate Refugia: Joint Inference from Fossil Records, Species Distribution Models and Phylogeography”. en. In: *New Phytologist* 204.1 (2014), pp. 37–54. ISSN: 1469-8137. DOI: 10.1111/nph.12929.
- [63] Alan E. Gelfand and Shinichiro Shirota. “Preferential sampling for presence/absence data and for fusion of presence/absence data with presence-only data”. In: *Ecological Monographs* (2019).
- [64] *Google Earth Engine*. <https://earthengine.google.com>.
- [65] Andrew M Gormley et al. “Using presence-only and presence–absence data to estimate the current and potential distributions of established invasive species”. In: *Journal of Applied Ecology* 48.1 (2011), pp. 25–34.
- [66] Catherine H Graham and Robert J Hijmans. “A comparison of methods for mapping species ranges and species richness”. In: *Global Ecology and biogeography* 15.6 (2006), pp. 578–587.

- [67] Annegret Grimm, Bernd Gruber, and Klaus Henle. “Reliability of different mark-recapture methods for population size estimation tested against reference population sizes constructed from field data”. In: *PLoS One* 9.6 (2014), e98840.
- [68] Liam Grimmitt, Rachel Whitsed, and Ana Horta. “Presence-Only Species Distribution Models Are Sensitive to Sample Prevalence: Evaluating Models Using Spatial Prediction Stability and Accuracy Metrics”. en. In: *Ecological Modelling* 431 (Sept. 2020), p. 109194. ISSN: 0304-3800. DOI: 10.1016/j.ecolmodel.2020.109194.
- [69] Joseph Grinnell. “The origin and distribution of the chest-nut-backed chickadee”. In: *The Auk* 21.3 (1904), pp. 364–382.
- [70] Red List Technical Working Group et al. *Mapping Standards and Data Quality for the IUCN Red List Categories and Criteria*. 2018.
- [71] Shixiang Gu et al. “Continuous deep q-learning with model-based acceleration”. In: *International Conference on Machine Learning*. PMLR. 2016, pp. 2829–2838.
- [72] Gurutzeta Guillera-Arroita et al. “Is My Species Distribution Model Fit for Purpose? Matching Data and Models to Applications”. en. In: *Global Ecology and Biogeography* 24.3 (2015), pp. 276–292. ISSN: 1466-8238. DOI: 10.1111/geb.12268.
- [73] Antoine Guisan and Wilfried Thuiller. “Predicting Species Distribution: Offering More than Simple Habitat Models”. en. In: *Ecology Letters* 8.9 (2005), pp. 993–1009. ISSN: 1461-0248. DOI: 10.1111/j.1461-0248.2005.00792.x.
- [74] Antoine Guisan and Wilfried Thuiller. “Predicting species distribution: offering more than simple habitat models”. In: *Ecology letters* 8.9 (2005), pp. 993–1009.
- [75] Antoine Guisan and Niklaus E Zimmermann. “Predictive habitat distribution models in ecology”. In: *Ecological modelling* 135.2-3 (2000), pp. 147–186.
- [76] W Hallgren et al. “Species distribution models can be highly sensitive to algorithm configuration”. In: *Ecological Modelling* 408 (2019), p. 108719.
- [77] Thomas A Hanley. “A comparison of the line interception and quadrat estimation methods of determining shrub canopy coverage.” In: *Range-land Ecology & Management/Journal of Range Management Archives* 31.1 (1978), pp. 60–62.
- [78] David J. Harris. “Generating realistic assemblages with a joint species distribution model”. In: *Methods in Ecology and Evolution* (2015).
- [79] Trevor Hastie and Will Fithian. “Inference from presence-only data; the ongoing controversy”. In: *Ecography* 36.8 (2013), pp. 864–867.

- [80] Kate S He et al. “Will remote sensing shape the next generation of species distribution models?” In: *Remote Sensing in Ecology and Conservation* 1.1 (2015), pp. 4–18.
- [81] Harold F Heady and RW Gibbens. “A comparison of the charting, line intercept, and line point methods of sampling shrub types of vegetation.” In: *Rangeland Ecology & Management/Journal of Range Management Archives* 12.4 (1959), pp. 180–188.
- [82] Emilie B Henderson et al. “Species distribution modelling for plant communities: stacked single species or multivariate modelling approaches?” In: *Applied vegetation science* 17.3 (2014), pp. 516–527.
- [83] Tomislav Hengl et al. “SoilGrids250m: Global gridded soil information based on machine learning”. In: *PLoS one* 12.2 (2017), e0169748.
- [84] Motoki Higa et al. “Mapping large-scale bird distributions using occupancy models and citizen science data with spatially biased sampling effort”. In: *Diversity and Distributions* (2014).
- [85] Collin Homer et al. “Completion of the 2011 National Land Cover Database for the conterminous United States—representing a decade of land cover change information”. In: *Photogrammetric Engineering & Remote Sensing* 81.5 (2015), pp. 345–354.
- [86] Francis KC Hui et al. “To mix or not to mix: comparing the predictive performance of mixture models vs. separate species distribution models”. In: *Ecology* 94.9 (2013), pp. 1913–1919.
- [87] Allen H Hurlbert and Walter Jetz. “Species richness, hotspots, and the scale dependence of range maps in ecology and conservation”. In: *Proceedings of the National Academy of Sciences* 104.33 (2007), pp. 13384–13389.
- [88] Allen H Hurlbert and Ethan P White. “Disparity between range map-and survey-based analyses of species richness: patterns, processes and implications”. In: *Ecology Letters* 8.3 (2005), pp. 319–327.
- [89] *ICCV 2019 Workshop and Challenge on Computer Vision for Wildlife Conservation (CVWC)*. <https://cvwc2019.github.io/>.
- [90] *iNaturalist*. <https://www.inaturalist.org/>.
- [91] Walter Jetz, Jana M McPherson, and Robert P Guralnick. “Integrating biodiversity distribution knowledge: toward a global map of life”. In: *Trends in ecology & evolution* 27.3 (2012), pp. 151–159.
- [92] Walter Jetz et al. “Essential Biodiversity Variables for Mapping and Monitoring Species Populations”. In: *Nature Ecology & Evolution* 3 (Mar. 2019). doi: 10.1038/s41559-019-0826-1.

- [93] Chris J. Johnson and Michael P. Gillingham. “Sensitivity of Species-Distribution Models to Error, Bias, and Model Design: An Application to Resource Selection Functions for Woodland Caribou”. en. In: *Ecological Modelling* 213.2 (May 2008), pp. 143–155. ISSN: 0304-3800. DOI: 10.1016/j.ecolmodel.2007.11.013.
- [94] Maxwell B. Joseph. “Neural Hierarchical Models of Ecological Populations”. en. In: *Ecology Letters* 23.4 (2020), pp. 734–747. ISSN: 1461-0248. DOI: 10.1111/ele.13462.
- [95] Lou Jost. “Entropy and diversity”. In: *Oikos* 113.2 (2006), pp. 363–375.
- [96] Lou Jost. “Partitioning diversity into independent alpha and beta components”. In: *Ecology* 88.10 (2007), pp. 2427–2439.
- [97] Paulo De Marco Júnior and Caroline Corrêa Nóbrega. “Evaluating Collinearity Effects on Species Distribution Models: An Approach Based on Virtual Species Simulation”. en. In: *PLOS ONE* 13.9 (Sept. 2018), e0202403. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0202403.
- [98] Jamie M Kass et al. “Wallace: A flexible platform for reproducible modeling of species niches and distributions built for community expansion”. In: *Methods in Ecology and Evolution* 9.4 (2018), pp. 1151–1156.
- [99] W. Daniel Kissling et al. “Towards Global Data Products of Essential Biodiversity Variables on Species Traits”. en. In: *Nature Ecology & Evolution* 2.10 (Oct. 2018), pp. 1531–1540. ISSN: 2397-334X. DOI: 10.1038/s41559-018-0667-3.
- [100] Pang Wei Koh et al. “WILDS: A Benchmark of in-the-Wild Distribution Shifts”. In: *arXiv preprint arXiv:2012.07421* (2020).
- [101] Vira Koshkina et al. “Integrated species distribution models: combining presence-background data and site-occupancy data with imperfect detection”. In: *Methods in Ecology and Evolution* (2017).
- [102] Margaret Kosmala et al. “Assessing data quality in citizen science”. In: *Frontiers in Ecology and the Environment* 14.10 (2016), pp. 551–560.
- [103] Thierry Lassueur, Stéphane Joost, and Christophe F Randin. “Very high resolution digital elevation models: Do they improve models of plant species distribution?” In: *Ecological Modelling* 198.1-2 (2006), pp. 139–153.
- [104] Jonas J. Lembrechts, Ivan Nijs, and Jonathan Lenoir. “Incorporating Microclimate into Species Distribution Models”. en. In: *Ecography* 42.7 (2019), pp. 1267–1279. ISSN: 1600-0587. DOI: 10.1111/ecog.03947.
- [105] Wanwan Liang et al. “The Effect of Pseudo-Absence Selection Method on Transferability of Species Distribution Models in the Context of Non-Adaptive Niche Shift”. en. In: *Ecological Modelling* 388 (Nov. 2018), pp. 1–9. ISSN: 0304-3800. DOI: 10.1016/j.ecolmodel.2018.09.018.

- [106] *LILA.science*. <http://lila.science/>. Accessed: 2019-10-22.
- [107] Canran Liu, Matt White, and Graeme Newell. “Selecting Thresholds for the Prediction of Species Occurrence with Presence-Only Data”. en. In: *Journal of Biogeography* 40.4 (2013), pp. 778–789. ISSN: 1365-2699. DOI: 10.1111/jbi.12058.
- [108] David M Lodge et al. “Conservation in a cup of water: estimating biodiversity and population abundance from environmental DNA”. In: *Molecular ecology* 21.11 (2012), pp. 2555–2558.
- [109] Oisín Mac Aodha, Elijah Cole, and Pietro Perona. “Presence-only geographical priors for fine-grained image classification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9596–9606. DOI: 10.48550/arXiv.1906.05272.
- [110] Robert H MacArthur. “Population ecology of some warblers of northeastern coniferous forests”. In: *Ecology* 39.4 (1958), pp. 599–619.
- [111] Darryl I MacKenzie et al. “Estimating site occupancy rates when detection probabilities are less than one”. In: *Ecology* (2002).
- [112] Kumar Mainali et al. “Matching expert range maps with species distribution model predictions”. In: *Conservation Biology* 34.5 (2020), pp. 1292–1304. DOI: 10.1111/cobi.13492.
- [113] *Map of Life: Von Der Decken’s Hornbill*. https://mol.org/species/map/Tockus_deckeni.
- [114] Tiago A Marques et al. “Estimating North Pacific right whale *Eubalaena japonica* density using passive acoustic cue counting”. In: *Endangered Species Research* 13.3 (2011), pp. 163–172.
- [115] Jason Matthiopoulos, John Fieberg, and Geert Aarts. *Species-Habitat Associations: Spatial Data, Predictive Models, and Ecological Insights*. en. University of Minnesota Libraries Publishing, Dec. 2020. DOI: 10.24926/2020.081320.
- [116] William J. McSHEA. “What Are the Roles of Species Distribution Models in Conservation Planning?” en. In: *Environmental Conservation* 41.2 (June 2014), pp. 93–96. ISSN: 0376-8929, 1469-4387. DOI: 10.1017/S0376892913000581.
- [117] Cory Merow, Matthew J. Smith, and John A. Silander. “A Practical Guide to MaxEnt for Modeling Species’ Distributions: What It Does, and Why Inputs and Settings Matter”. en. In: *Ecography* 36.10 (2013), pp. 1058–1069. ISSN: 1600-0587. DOI: 10.1111/j.1600-0587.2013.07872.x.
- [118] Christine N. Meynard, Boris Leroy, and David M. Kaplan. “Testing Methods in Species Distribution Modelling Using Virtual Species: What Have We Learnt and What Are We Missing?” en. In: *Ecography* 42.12 (2019), pp. 2021–2036. ISSN: 1600-0587. DOI: 10.1111/ecog.04385.

- [119] David A. W. Miller et al. “The recent past and promising future for data integration methods to estimate species’ distributions”. In: *Methods in Ecology and Evolution* (2019).
- [120] Jennifer A Miller and Paul Holloway. “Incorporating Movement in Species Distribution Models”. en. In: *Progress in Physical Geography: Earth and Environment* 39.6 (Dec. 2015), pp. 837–849. ISSN: 0309-1333. DOI: 10.1177/0309133315580890.
- [121] Ans M Mouton, Bernard De Baets, and Peter LM Goethals. “Ecological relevance of performance criteria for species distribution models”. In: *Ecological modelling* 221.16 (2010), pp. 1995–2002.
- [122] Movebank. <https://www.movebank.org/cms/movebank-main>.
- [123] M Arthur Munson et al. “The eBird reference dataset”. In: *Cornell Lab of Ornithology and National Audubon Society, Ithaca, NY [En linea]: http://www.avianknowledge.net/content*. Acceso: Julio (2011).
- [124] Andrew Murray. *The geographical distribution of mammals*. 1866.
- [125] Ernest Mwebaze et al. *iCassava 2019 Fine-Grained Visual Categorization Challenge*. 2019. arXiv: 1908.02900 [cs.CV].
- [126] Babak Naimi et al. “Where is positional uncertainty a problem for species distribution modelling?” In: *Ecography* 37.2 (2014), pp. 191–203.
- [127] NASA Open Data Portal. <https://www.data.nasa.gov/>.
- [128] John Ashworth Nelder and Robert WM Wedderburn. “Generalized linear models”. In: *Journal of the Royal Statistical Society: Series A (General)* 135.3 (1972), pp. 370–384.
- [129] Anna Norberg. *aminorberg/SDM-comparison: Norberg et al. (2019)*. Version publication. Apr. 2019. DOI: 10.5281/zenodo.2637812. URL: <https://doi.org/10.5281/zenodo.2637812>.
- [130] Anna Norberg et al. “A Comprehensive Evaluation of Predictive Performance of 33 Species Distribution Models at Species and Community Levels”. en. In: *Ecological Monographs* 89.3 (2019), e01370. ISSN: 1557-7015. DOI: 10.1002/ecm.1370.
- [131] Anna Norberg et al. “A comprehensive evaluation of predictive performance of 33 species distribution models at species and community levels”. In: *Ecological Monographs* 89.3 (2019), e01370.
- [132] Mohammad Sadegh Norouzzadeh et al. “Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning”. In: *Proceedings of the National Academy of Sciences* 115.25 (2018), E5716–E5725.

- [133] *OOS 64 - Deep Learning for Image Analysis in Ecology, Session at ESA 2020*. <https://eco.confex.com/eco/2020/meetingapp.cgi/Session/17295>.
- [134] Stan Openshaw. *The Modifiable Areal Unit Problem*. Norwick, 1984.
- [135] Otso Ovaskainen et al. “How to make more out of community data? A conceptual framework and its implementation as models and software”. In: *Ecology Letters* 20.5 (2017), pp. 561–576.
- [136] Stacy L. Özesmi and Uygur Özesmi. “An artificial neural network approach to spatial habitat modelling with interspecific interaction”. In: *Ecological Modelling* (1999).
- [137] Krishna Pacifici et al. “Integrating multiple data sources in species distribution modeling: a framework for data fusion”. In: *Ecology* (2016).
- [138] Krishna Pacifici et al. “Integrating multiple data sources in species distribution modeling: a framework for data fusion”. In: *Ecology* 98.3 (2017), pp. 840–850.
- [139] Krishna Pacifici et al. “Integrating Multiple Data Sources in Species Distribution Modeling: A Framework for Data Fusion*”. en. In: *Ecology* 98.3 (2017), pp. 840–850. ISSN: 1939-9170. DOI: 10.1002/ecy.1710.
- [140] Jennie Pearce and Simon Ferrier. “An evaluation of alternative algorithms for fitting species distribution models using logistic regression”. In: *Ecological modelling* 128.2-3 (2000), pp. 127–147.
- [141] Peter B. Pearman et al. “Niche Dynamics in Space and Time”. en. In: *Trends in Ecology & Evolution* 23.3 (Mar. 2008), pp. 149–158. ISSN: 0169-5347. DOI: 10.1016/j.tree.2007.11.005.
- [142] A Townsend Peterson and Jorge Soberón. “Species distribution modeling and ecological niche modeling: getting the concepts right”. In: *Natureza & Conservação* 10.2 (2012), pp. 102–107.
- [143] Steven Phillips and Jane Elith. “Logistic Methods for Resource Selection Functions and Presence-Only Species Distribution Models”. en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 25.1 (Aug. 2011). ISSN: 2374-3468.
- [144] Steven J Phillips, Robert P Anderson, and Robert E Schapire. “Maximum entropy modeling of species geographic distributions”. In: *Ecological modelling* 190.3-4 (2006), pp. 231–259.
- [145] Maximilian Pichler and Florian Hartig. “A New Method for Faster and More Accurate Inference of Species Associations from Big Community Data”. In: *arXiv:2003.05331 [q-bio, stat]* (Oct. 2020). arXiv: 2003.05331 [q-bio, stat].

- [146] Giovanni Poggiato et al. “On the Interpretations of Joint Modeling in Community Ecology”. en. In: *Trends in Ecology & Evolution* (Feb. 2021). ISSN: 0169-5347. DOI: 10.1016/j.tree.2021.01.002.
- [147] Laura J. Pollock, William K. Morris, and Peter A. Vesk. “The Role of Functional Traits in Species Distributions Revealed through a Hierarchical Model”. en. In: *Ecography* 35.8 (2012), pp. 716–725. ISSN: 1600-0587. DOI: 10.1111/j.1600-0587.2011.07085.x.
- [148] Kevin L Pope, Steve E Lochmann, and Michael K Young. “Methods for assessing fish populations”. In: *In: Hubert, Wayne A; Quist, Michael C., eds. Inland Fisheries Management in North America, 3rd edition. Bethesda, MD: American Fisheries Society: 325-351.* (2010), pp. 325–351.
- [149] Stephen M. Powers and Stephanie E. Hampton. “Open science, reproducibility, and transparency in ecology”. In: *Ecological Applications* (2011).
- [150] Jean-Nicolas Pradervand et al. “Very high resolution environmental predictors in species distribution models: moving beyond topography?” In: *Progress in Physical Geography* 38.1 (2014), pp. 79–96.
- [151] O. J. Reichman, Matthew B. Jones, and Mark P. Schildhauer. “Challenges and Opportunities of Open Data in Ecology”. In: *Science* (2011).
- [152] Ian W Renner and David I Warton. “Equivalence of MAXENT and Poisson point process models for species distribution modeling in ecology”. In: *Biometrics* 69.1 (2013), pp. 274–281.
- [153] Corinne L Richards, Bryan C Carstens, and L Lacey Knowles. “Distribution modelling and statistical phylogeography: an integrative framework for generating and testing alternative biogeographical hypotheses”. In: *Journal of Biogeography* 34.11 (2007), pp. 1833–1845.
- [154] David R. Roberts et al. “Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure”. In: *Ecography* (2016), pp. 913–929.
- [155] Tim Robertson et al. “Training machines to identify species using gbif-mediated datasets”. In: *Biodiversity Information Science and Standards* (2019).
- [156] Caleb Robinson et al. “Large scale high-resolution land cover mapping with multi-resolution data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12726–12735.
- [157] Duccio Rocchini et al. “Accounting for uncertainty when mapping species distributions: the need for maps of ignorance”. In: *Progress in Physical Geography* 35.2 (2011), pp. 211–226.
- [158] J Marcus Rowcliffe et al. “Estimating animal density using camera traps without the need for individual recognition”. In: *Journal of Applied Ecology* (2008), pp. 1228–1236.

- [159] Eric W Sanderson et al. “The human footprint and the last of the wild: the human footprint is a global map of human influence on the land surface, which suggests that human beings are stewards of nature, whether we like it or not”. In: *BioScience* 52.10 (2002), pp. 891–904.
- [160] Andreas Franz Wilhelm Schimper. *Plant-geography Upon a Physiological Basis...* Clarendon Press, 1903.
- [161] Zoe Emily Schnabel. “The estimation of the total fish population of a lake”. In: *The American Mathematical Monthly* 45.6 (1938), pp. 348–352.
- [162] Patrick Schratz et al. “Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data”. In: *Ecological Modelling* 406 (2019), pp. 109–120.
- [163] Boris Schroeder. “Challenges of species distribution modeling belowground”. In: *Journal of Plant Nutrition and Soil Science* 171.3 (2008), pp. 325–337.
- [164] Sebastian Seibold et al. “Arthropod decline in grasslands and forests is associated with landscape-level drivers”. In: *Nature* 574.7780 (2019), pp. 671–674.
- [165] Steve J. Sinclair, Matthew D. White, and Graeme R. Newell. “How Useful Are Species Distribution Models for Managing Biodiversity under Future Climates?” In: *Ecology and Society* 15.1 (2010). ISSN: 1708-3087.
- [166] Andrea Soriano-Redondo et al. “Understanding Species Distribution in Dynamic Populations: A New Approach Using Spatio-Temporal Point Process Models”. en. In: *Ecography* 42.6 (2019), pp. 1092–1102. ISSN: 1600-0587. DOI: [10.1111/ecog.03771](https://doi.org/10.1111/ecog.03771).
- [167] Jakub Stoklosa et al. “A climate of uncertainty: accounting for error in climate variables for species distribution models”. In: *Methods in Ecology and Evolution* 6.4 (2015), pp. 412–423.
- [168] DF Stuffer. “Linking populations and habitats: where have we been? Where are we going?” In: *Predicting species occurrences: Issues of accuracy and scale* (2002).
- [169] Alexandra Swanson et al. “Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna”. In: *Scientific data* 2.1 (2015), pp. 1–14.
- [170] FC Sweeney and JM Hopkinson. “Vegetative growth of nineteen tropical and sub-tropical pasture grasses and legumes in relation to temperature”. In: *Tropical Grasslands* 9.3 (1975), pp. 209–217.
- [171] Renee Swischuk et al. “Projection-based model reduction: Formulations for physics-based machine learning”. In: *Computers & Fluids* 179 (2019), pp. 704–717.

- [172] Nicholas W Synes and Patrick E Osborne. “Choice of predictor variables as a source of uncertainty in continental-scale species distribution modelling under climate change”. In: *Global Ecology and Biogeography* 20.6 (2011), pp. 904–914.
- [173] Matthew V. Talluto et al. “Cross-Scale Integration of Knowledge for Predicting Species Ranges: A Metamodelling Framework”. en. In: *Global Ecology and Biogeography* 25.2 (2016), pp. 238–249. ISSN: 1466-8238. DOI: 10.1111/geb.12395.
- [174] Kiat Chuan Tan et al. *The Herbarium Challenge 2019 Dataset*. 2019. arXiv: 1906.05372 [cs.CV].
- [175] Luming Tang et al. “Multi-entity dependence learning with rich context via conditional variational auto-encoder”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.
- [176] Ranjita Thapa et al. *The Plant Pathology 2020 challenge dataset to classify foliar disease of apples*. 2020. arXiv: 2004.11958 [cs.CV].
- [177] *The Global Biodiversity Information Facility*. <https://www.gbif.org/>.
- [178] James T. Thorson et al. “Joint Dynamic Species Distribution Models: A Tool for Community Ordination and Spatio-Temporal Monitoring”. en. In: *Global Ecology and Biogeography* 25.9 (2016), pp. 1144–1158. ISSN: 1466-8238. DOI: 10.1111/geb.12464.
- [179] Wilfried Thuiller et al. “Predicting Global Change Impacts on Plant Species’ Distributions: Future Challenges”. en. In: *Perspectives in Plant Ecology, Evolution and Systematics*. Space Matters - Novel Developments in Plant Ecology through Spatial Modelling 9.3 (Mar. 2008), pp. 137–152. ISSN: 1433-8319. DOI: 10.1016/j.ppees.2007.09.004.
- [180] Gleb Tikhonov et al. “Computationally Efficient Joint Species Distribution Modeling of Big Spatial Data”. en. In: *Ecology* 101.2 (2020), e02929. ISSN: 1939-9170. DOI: 10.1002/ecy.2929.
- [181] Tina Tirelli and Daniela Pessani. “Use of decision tree and artificial neural network approaches to model presence/absence of *Telestes muticellus* in piedmont (North-Western Italy)”. In: *River research and applications* 25.8 (2009), pp. 1001–1012.
- [182] Anne M. Trainor et al. “Enhancing Species Distribution Modeling by Characterizing Predator–Prey Interactions”. en. In: *Ecological Applications* 24.1 (2014), pp. 204–216. ISSN: 1939-5582. DOI: 10.1890/13-0336.1.
- [183] Hanna Tuomisto. “A diversity of beta diversities: straightening up a concept gone awry. Part 1. Defining beta diversity as a function of alpha and gamma diversity”. In: *Ecography* 33.1 (2010), pp. 2–22.
- [184] Courtney A. Tye et al. “Evaluating citizen vs. professional data for modelling distributions of a rare squirrel”. In: *Journal of Applied Ecology* (2016).

- [185] *USGS LPDAAC*. <https://lpdaac.usgs.gov>.
- [186] Tomáš Václavík, John A. Kupfer, and Ross K. Meentemeyer. “Accounting for Multi-Scale Spatial Autocorrelation Improves Performance of Invasive Species Distribution Modelling (iSDM)”. en. In: *Journal of Biogeography* 39.1 (2012), pp. 42–55. ISSN: 1365-2699. DOI: 10.1111/j.1365-2699.2011.02589.x.
- [187] Alice Valentini et al. “Next-generation monitoring of aquatic biodiversity using environmental DNA metabarcoding”. In: *Molecular ecology* 25.4 (2016), pp. 929–942.
- [188] Grant Van Horn and Pietro Perona. “The devil is in the tails: Fine-grained classification in the wild”. In: *arXiv preprint arXiv:1709.01450* (2017).
- [189] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisín Mac Aodha. “Benchmarking representation learning for natural world image collections”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12884–12893. DOI: 10.48550/arXiv.2103.16483.
- [190] Grant Van Horn et al. “The iNaturalist species classification and detection dataset”. In: *CVPR*. 2018.
- [191] Jeremy VanDerWal et al. “Selecting Pseudo-Absence Data for Presence-Only Distribution Modeling: How Far Should You Stray from What You Know?” en. In: *Ecological Modelling* 220.4 (Feb. 2009), pp. 589–594. ISSN: 0304-3800. DOI: 10.1016/j.ecolmodel.2008.11.010.
- [192] William N Venables and Brian D Ripley. *Modern applied statistics with S-PLUS*. Springer Science & Business Media, 2013.
- [193] WCEP Verberk. “Explaining general patterns in species abundance and distributions”. In: *Nature Education Knowledge* 3.10 (2011), p. 38.
- [194] Peter A. Vesik et al. “Transferability of Trait-Based Species Distribution Models”. en. In: *Ecography* 44.1 (2021), pp. 134–147. ISSN: 1600-0587. DOI: 10.1111/ecog.05179.
- [195] Jean-Christophe Vié et al. “The IUCN Red List: a key conservation tool”. In: *Wildlife in a changing world—An analysis of the 2008 IUCN Red List of Threatened Species* (2009), p. 1.
- [196] YI Wang et al. “mvabund—an R package for model-based analysis of multivariate abundance data”. In: *Methods in Ecology and Evolution* 3.3 (2012), pp. 471–474.
- [197] Gill Ward et al. “Presence-Only Data and the EM Algorithm”. en. In: *Biometrics* 65.2 (2009), pp. 554–563. ISSN: 1541-0420. DOI: 10.1111/j.1541-0420.2008.01116.x.

- [198] Robert H Whittaker. “Vegetation of the great smoky mountains”. In: *Ecological Monographs* 26.1 (1956), pp. 2–80.
- [199] Robert Harding Whittaker. “Vegetation of the Siskiyou mountains, Oregon and California”. In: *Ecological monographs* 30.3 (1960), pp. 279–338.
- [200] John J. Wiens. “The Niche, Biogeography and Species Interactions”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 366.1576 (Aug. 2011), pp. 2336–2350. DOI: 10.1098/rstb.2011.0059.
- [201] *Wildlife Insights*. <https://www.wildlifeinsights.org/home>.
- [202] David P. Wilkinson et al. “A Comparison of Joint Species Distribution Models for Presence–Absence Data”. In: *Methods in Ecology and Evolution* 10.2 (Feb. 2019), pp. 198–211. ISSN: 2041-210X. DOI: 10.1111/2041-210X.13106.
- [203] David Peter Wilkinson. “A Comparison of the Inferential, Computational, and Predictive Performance of Joint Species Distribution Models”. en. In: (2019).
- [204] Mary S. Wisz and Antoine Guisan. “Do Pseudo-Absence Selection Strategies Influence Species Distribution Models and Their Predictions? An Information-Theoretic Approach Based on Simulated Data”. In: *BMC Ecology* 9.1 (Apr. 2009), p. 8. ISSN: 1472-6785. DOI: 10.1186/1472-6785-9-8.
- [205] Simon N Wood. “Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.1 (2011), pp. 3–36.
- [206] Peggy PW Yen, Falk Huettmann, and Fred Cooke. “A large-scale model for the at-sea distribution and abundance of Marbled Murrelets (*Brachyramphus marmoratus*) during the breeding season in coastal British Columbia, Canada”. In: *Ecological Modelling* 171.4 (2004), pp. 395–413.
- [207] Damaris Zurell et al. “Testing species assemblage predictions from stacked and joint species distribution models”. In: *Journal of Biogeography* 47.1 (2020), pp. 101–113.

PRESENCE-ONLY GEOGRAPHICAL PRIORS FOR FINE-GRAINED IMAGE CLASSIFICATION

- [1] Oisín Mac Aodha, Elijah Cole, and Pietro Perona. “Presence-only geographical priors for fine-grained image classification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9596–9606. DOI: [10.48550/arXiv.1906.05272](https://doi.org/10.48550/arXiv.1906.05272).

7.1 Abstract

Appearance information alone is often not sufficient to accurately differentiate between fine-grained visual categories. Human experts make use of additional cues such as where, and when, a given image was taken in order to inform their final decision. This contextual information is readily available in many online image collections but has been underutilized by existing image classifiers that focus solely on making predictions based on the image contents. We propose an efficient spatio-temporal prior that, when conditioned on a geographical location and time, estimates the probability that a given object category occurs at that location. Our prior is trained from presence-only observation data and jointly models object categories, their spatio-temporal distributions, and photographer biases. Experiments performed on multiple challenging image classification datasets show that combining our prior with the predictions from image classifiers results in a large improvement in final classification performance.

7.2 Introduction

Correctly classifying objects into different fine-grained visual categories is a challenging problem. In contrast to generic object recognition, it can require knowledge of subtle features that are essential for differentiating between visually similar categories. However, without having access to additional information that may not be present in an image, many categories can be visually indistinguishable. For example, the two toad species in Fig. 7.1 are similar in appearance but tend to be found in very different locations in Europe. Knowing *where* a given image was taken can provide a strong prior for *what* objects it may contain.

Most images that are captured and shared online today also come with additional

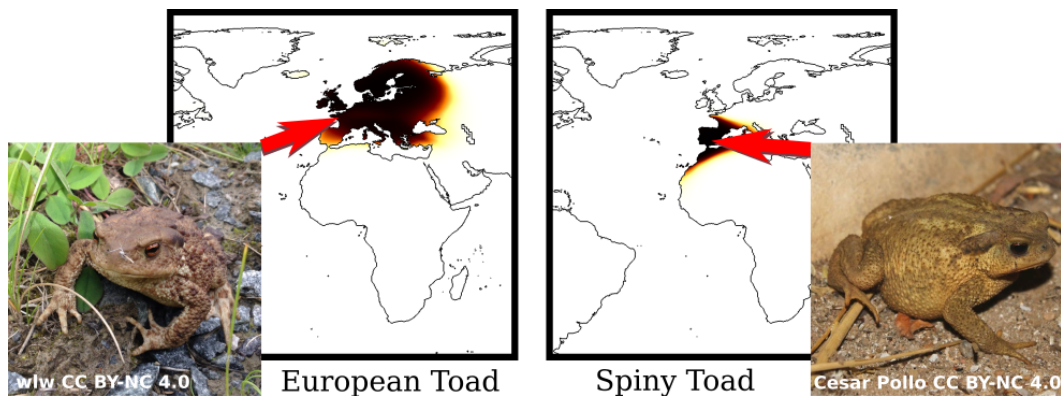


Figure 7.1: Differentiating between two visually similar categories such as the European (left) and Spiny (right) Toad can be challenging without additional context. To address this problem, we propose a spatio-temporal prior that encodes where, and when, a given category is likely to occur. For a known test location our prior predicts how likely it is for each category to be present. Darker colors indicate locations that are more likely to contain the object of interest.

metadata in the form of *where* they were taken, *when* they were taken, and *who* captured them. This information not only offers the possibility of helping to resolve ambiguous cases for image classification, but can also enable us to generate predictions of where, and when, different objects are likely to be observed.

Existing work that uses location information to improve classification performance either discretizes the input data into spatio-temporal volumes [3], store the entire training set in memory at inference time [64], or jointly train deep images classifiers along with corresponding location information [55]. Methods that discretize or store the raw training data do not scale well in terms of memory, and jointly training image classifiers with location information necessitates that location information is present at test time - which may not always be the case. We take inspiration from species distribution modeling (SDM) [18], and instead model a separate geographical prior that can be combined with the predictions of *any* image classifier. However, unlike many approaches to SDM that assume they have access to presence and absence information at training time (e.g. [56]), we make a more general assumption that only presence information is available i.e. we know where the categories have been observed, but have *no* explicit data regarding where they are not found.

In this work we make the following contributions: (1) an efficient spatio-temporal prior that jointly models the relationship between location, time of year, photographer, and the presence of multiple different object categories; (2) a novel presence-only training loss to capture these relationships; and (3) experiments that show

that combining the probabilistic predictions of image classifiers with our prior significantly improves the test time performance on challenging fine-grained image datasets.

7.3 Related Work

Here we discuss work related to spatio-temporal models that encode the location of a set of discrete object categories. We do not address methods that explore other uses of location information such as inferring where an image was taken given only the raw pixels [28, 60], or methods that use location to disambiguate visually similar places for image localization [59, 68].

Fine-Grained Image Classification

Correctly determining which one of multiple possible fine-grained categories is present in an image requires understanding the relationship between subtle visual features and the corresponding image-level category label e.g. [61, 35, 66, 58]. Existing approaches have investigated the modeling of parts [41, 70, 7, 71, 31], higher order feature interactions [40, 21], attention mechanisms [65, 72, 62], noisy web data [37], novel training losses [12], and pairwise category information [15]. Orthogonal to those works, we propose a spatio-temporal prior that can be combined with the probabilistic predictions of any image classifier to improve the final classification performance.

Location and Classification

A small number of approaches have explored the use of location information to improve image classification at test time. Berg et al. [3] proposed a spatio-temporal prior that when combined with the output of an image classifier increased the accuracy of bird species classification. Their approach discretized location and time into spatio-temporal cubes and used an adaptive kernel density estimator to represent the distribution of each species independently. Also in the context of predicting the presence of different biological species, Wittich et al. [64] evaluated different nearest neighbor based lookup strategies for retrieving the most relevant instances from a training set of geo-tagged observations. These approaches are inefficient in terms their memory requirements as they necessitate storing either the entire training set or a discretized version of it. Existing repositories of citizen science data (e.g. [53, 33, 23]) can contain on the order of tens of millions of observations making them prohibitively large to store and retrieve on mobile devices. Choosing the correct

discretization is challenging [48], and incorrect choices can significantly affect the final performance [38, 46]. A key benefit of our approach is that discretization is not required.

Tang et al. [55] explored different feature encodings for incorporating location information directly into deep neural networks at training time. This included raw location features (i.e. longitude and latitude), demographic information collected via a census, user provided hash-tags, and geographical map features (e.g. land use estimates). The disadvantage of their method is that it assumes that location information is present at test time and that all the required features can be computed for a given test location. Furthermore, they cannot use location information that does not have an associated image. They also need to retrain their entire model if new location data is collected. We instead propose an efficient spatio-temporal prior that jointly models the spatial distribution of multiple object categories that can be trained independently of the image classifier. Parallel to our work, [11] builds on [55] by exploring different ways to integrate location information into deep image classifiers.

Spatio-Temporal Distribution Modelling

Our goal is to estimate the spatio-temporal distribution of a set of object categories. Related to this, there is a rich literature exploring models for estimating the distribution of biological specimens across geographic space and time [30]. This is referred to as species distribution modelling or environmental niche modelling. Broadly, these methods can be divided into two groups: those that use *presence-absence* data and those that use *presence-only* data [27].

Making a presence-absence observation at a given location requires that every species from a predefined set of interest be confirmed as either present or absent for that sampling event. In practice, this kind of data is onerous to collect because it requires intense survey effort to confirm that a species is absent with a high degree of certainty [44]. However, once this data is collected it can be combined with standard supervised classification approaches such as logistic regression [27], probit regression [52], Gaussian processes [25], decision trees [18], and neural networks [67, 49, 45], among others [16, 47]. Presence-absence data is also compatible with traditional multi-label learning [34, 6, 69, 10, 63]. Recently deep models have been applied to this problem in order to jointly model the location preferences of different species [26, 9, 20, 56, 4] and human sampling biases [8].

In contrast, a presence-only (i.e. incidental) observation may be recorded wherever an object of interest is encountered - *without* requiring any absences to be verified. While presence-only data can be much easier to collect, the lack of absence information makes it more difficult to model. This limitation is typically dealt with in one of three different ways. The first approach is to generate ‘pseudo-negatives’ and then apply one of the presence-absence approaches from above. As no true negative information is available, these approaches randomly sample a set of locations and make the assumption that these locations are absences, e.g. [17, 51, 1]. The second commonly used approach is to train a highly regularized model directly on the presence-only data, e.g. by fitting a maximum entropy distribution [50] or a low-rank model [19], forcing the model to explain data where it has been observed and to be uncertain elsewhere. Finally, and most related to our work, there are approaches that use additional information such as the detectability of a given species and a photographer’s propensity to image them e.g. [43, 22].

Unlike many of the classic approaches for spatio-temporal distribution modelling, in this work we jointly learn a continuous spatio-temporal prior for each category of interest using a neural network to amortize the computation. In contrast to previous deep distribution models e.g. [26, 9, 56], we do not require presence-absence data or additional environmental features as input. We instead exploit the structure that exists in online image repositories, such as those collected by citizen scientists, to jointly model objects, their locations, and photographer biases.

7.4 Methods

Here we outline our spatio-temporal prior, which models the geographical and temporal distribution of a set of object categories and photographers. During training we assume that we have access to a set of tuples $\mathcal{D} = \{(I_i, \mathbf{x}_i, y_i, p_i) | i = 1, \dots, N\}$, where I_i is an image, $y_i \in \{1, \dots, C\}$ is the corresponding class label, $\mathbf{x}_i = [\text{lon}_i, \text{lat}_i, \text{time}_i]$ represents the location (longitude and latitude) and time the image was taken, and p_i is the individual, i.e. photographer, who captured the image. Note that the location does not need to be captured alongside the image. \mathcal{D} can be assembled from unrelated image and location datasets as long as both contain the same categories.

At test time, given an image and where and when it was taken we aim to estimate which category it contains, i.e. $P(y|I, \mathbf{x})$. One approach is to model the joint distribution $P(I, \mathbf{x})$ as in [55], but this necessitates that the location information is

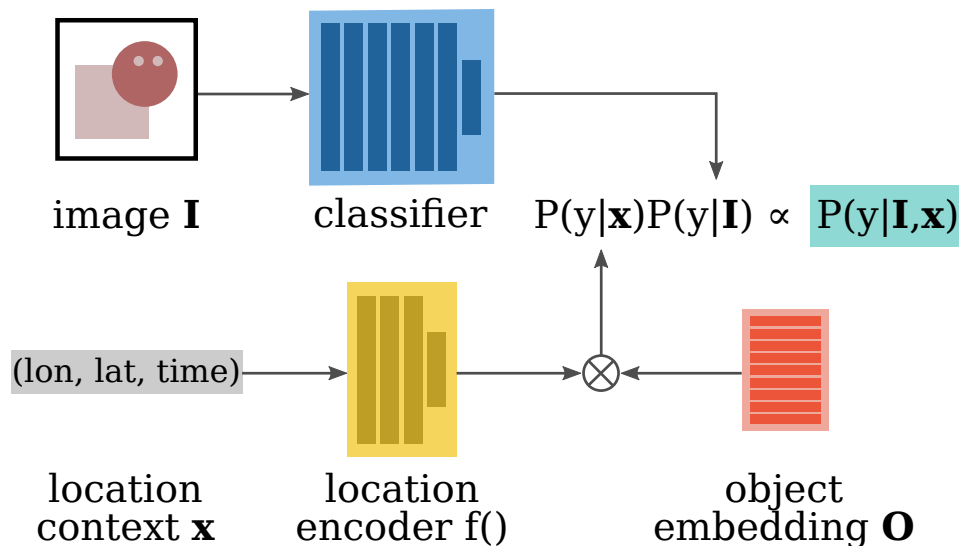


Figure 7.2: **Inference time.** Our goal is to estimate if an object category y is present in an input image I . At test time we make use of additional spatio-temporal information \mathbf{x} in the form of where and when the image was taken.

always available at test time. Instead, inspired by [3], we can incorporate location information as a Bayesian spatio-temporal prior. If we assume that I and \mathbf{x} are conditionally independent given y , then

$$P(y|I, \mathbf{x}) = \frac{P(I, \mathbf{x}|y)P(y)}{P(I, \mathbf{x})} \quad (7.1)$$

$$= \frac{P(I)P(\mathbf{x})}{P(I, \mathbf{x})} \frac{P(y|I)P(y|\mathbf{x})}{P(y)} \quad (7.2)$$

$$\propto P(y|I)P(y|\mathbf{x}), \quad (7.3)$$

where we assume a uniform prior $P(y) = 1/C$ for $y \in \{1 \dots, C\}$. In reality an image may contain location information unrelated to the class label (e.g. the background), but we assume this factorization is valid. By factoring the distribution in this way we can represent the image classifier, $P(y|I)$, and spatio-temporal prior, $P(y|\mathbf{x})$, separately. Note that at test time we do not assume that we have any knowledge of the individual p who captured the image. In this work we focus our attention on representing $P(y|\mathbf{x})$. For $P(y|I)$ we can use any discriminative model that produces a probabilistic output e.g. a convolutional neural network.

Presence-Absence Loss

As we are modeling the spatio-temporal prior independently from the image classifier our training data is now of the form $\mathcal{D} = \{(\mathbf{x}_i, y_i, p_i) | i = 1, \dots, N\}$. In the ideal

case we would have complete information consisting of where and when a given category has both been observed to be present and observed to be *not* present e.g. as in [9, 56]. Then instead of $y_i \in 1, \dots, C$, each spatio-temporal location \mathbf{x}_i would be associated with a binary multi-label vector $\mathbf{y}_i = [y_i^1, \dots, y_i^C]$ where each entry $y_i^c \in \{0, 1\}$ indicates whether or not category c has been observed as being present at \mathbf{x}_i . This formulation results in a standard multi-label learning problem, enabling us to estimate the parameters of the spatio-temporal model by solving

$$\max_{\theta} \sum_{i=1}^N \sum_{c=1}^C y_i^c \log(\hat{y}_i^c) + (1 - y_i^c) \log(1 - \hat{y}_i^c), \quad (7.4)$$

where we define $\hat{y}_i^c = P(y_i^c | \mathbf{x}_i)$ and P is parameterized by θ . However, as discussed previously, presence-absence information is both difficult and time consuming to acquire in real world settings.

Presence-Only Loss

In this work we explore the more challenging presence-only setting where each spatio-temporal location \mathbf{x}_i is associated with a single label $y_i \in \{1, \dots, C\}$ indicating which category was observed. In essence, we have a label vector \mathbf{y}_i where there is only one affirmative entry, i.e. $y_i^c = 1$ for some c , and the remaining entries are unknown. In this setting, Eqn. 7.4 can be written as

$$\max_{\theta} \sum_{i=1}^N \log(\hat{y}_i^{c_i}) + A_i, \quad (7.5)$$

where A_i represents a proxy absence term for the i^{th} training example and c_i is the corresponding observed category. Now the question becomes how to choose A_i .

One common approach for representing A_i is to generate ‘pseudo-negatives’ [1] by randomly sampling absence data from some parametric distribution. For instance, one might set

$$A_i = \log(1 - P(y_i | \mathbf{r}_i)). \quad (7.6)$$

where \mathbf{r}_i is a randomly selected spatio-temporal location with $[\text{lon}(\mathbf{r}_i), \text{lat}(\mathbf{r}_i)] \sim \text{Unif}(\mathbb{S}^2)$ and $\text{time}(\mathbf{r}_i) \sim \text{Unif}([0, 1])$. The implicit assumption is that each category (whether man-made or naturally occurring) occurs in a relatively small subset of $\mathbb{S}^2 \times [0, 1]$, so the probability of a category occurring at a randomly chosen location $\mathbf{r} \in \mathbb{S}^2 \times [0, 1]$ is small as well. To the extent that this assumption holds, these pseudo-negatives are likely to be valid.

An alternative approach is to instead sample absences over locations and times where the presence data for other categories occurs. In this case we would set A_i according to Eqn. 7.6 but sample negative locations from the positive occurrence locations i.e. $\mathbf{r}_i \sim \text{Unif}(\{\mathbf{x}_1, \dots, \mathbf{x}_N\})$. This biases the training towards regions that contain valid data.

Our Approach

In this section we outline how we model and train our spatio-temporal prior $P(y|\mathbf{x})$.

Location and Object Embedding

In many contexts, different objects do not occur independently at a given spatio-temporal location. Knowing that object A is present may provide information regarding the presence or absence of object B at the same place and time. Similarly, different spatio-temporal locations are not independent, and may share commonalities. We exploit this structure to encode low dimensional embeddings of objects and spatio-temporal locations.

Taking inspiration from [9], we model our spatio-temporal prior as $P(y|\mathbf{x}) \propto s(f(\mathbf{x})\mathbf{O})$. Here, $f : \mathcal{R}^3 \rightarrow \mathcal{R}^D$ is a multi-layered fully-connected neural network that maps a spatio-temporal location \mathbf{x} to a D dimensional embedding vector. $\mathbf{O} \in \mathcal{R}^{D \times C}$ represents an object embedding matrix, where each column is a different category. The product $f(\mathbf{x})\mathbf{O}$ results in a C dimensional vector, where each element represents the affinity that a spatio-temporal location \mathbf{x} has for category y . The intuition is that we are representing spatio-temporal locations and object categories in a shared embedding space where the inner product between the embedding of a location \mathbf{x} and an object y is large if y is likely to occur at location \mathbf{x} . Finally, $s()$ is an entry-wise sigmoid operation to ensure that the resulting predictions are in the range $[0, 1]$.

Photographer Embedding

In online image collections we often have access to additional information at training time in the form of the photographer $p \in \mathcal{P}$ who captured the image. To see why this information is valuable, consider the following example. Suppose a photographer p visits location \mathbf{x} and does *not* report object y . If p has never taken an image of an object like y , then this non-report gives us little information. However, if p has a history of reporting categories similar to object y , then this constitutes weak

evidence that y might actually be absent at that location. Thus, we can interpret the same presence-only information in different ways depending on the individual who provides it.

To capture photographer biases, we embed photographers into the same shared embedding space as the objects and locations. This is achieved by learning a photographer embedding matrix $\mathbf{P} \in \mathcal{R}^{D \times |\mathcal{P}|}$ at training time. Like different object categories, photographers may have affinities for particular locations and times, and share similarities in their spatio-temporal patterns with other photographers. This enables us to represent both a photographer's preference for a given location $P(p|\mathbf{x}) \propto s(f(\mathbf{x})\mathbf{P})$, and a photographer's affinity for a given object category $P(y|p) \propto s(\mathbf{O}^T\mathbf{P})$. Once trained, the photographer embeddings \mathbf{P} are not required at test time; see Fig. 8.1.

Joint Embedding Loss

Our goal at training time is to estimate the set of parameters $\theta = [\theta_f, \mathbf{O}, \mathbf{P}]$, where θ_f denotes the weights of the location embedding network $f()$, \mathbf{O} is the category embedding matrix, and \mathbf{P} is the photographer embedding matrix.

We start with the constraint that our model should be conservative i.e. if a category y has been observed at the spatio-temporal location \mathbf{x} in the training set, then $s(f(\mathbf{x})\mathbf{O}_{:,y})$ should be close to 1, otherwise it should be close to 0. Here, $\mathbf{O}_{:,y}$ indicates the y^{th} column of \mathbf{O} . We rely on the location embedding function $f()$ to interpolate between presence locations. This is conservative in the sense that it assumes that an object is absent if it has not been observed. This is a very strong assumption, but it enables the spatio-temporal prior to be aggressive in down-weighting incorrect predictions from the image classifier.

Our first loss encourages the model to predict the presence of objects where they have been observed in the training set and downweight their likelihood where they have not been observed:

$$\begin{aligned} \mathcal{L}_{o_loc}(\mathbf{x}, \mathbf{r}, \mathbf{O}, y) = & \lambda \log(s(f(\mathbf{x})\mathbf{O}_{:,y})) + \\ & \sum_{\substack{i=1 \\ i \neq y}}^C \log(1 - s(f(\mathbf{x})\mathbf{O}_{:,i})) + \\ & \sum_{i=1}^C \log(1 - s(f(\mathbf{r})\mathbf{O}_{:,i})). \end{aligned} \tag{7.7}$$

$P(y \mathbf{x})$ - Prior Type	YFCC	BirdSnap	BirdSnap [†]	NABirds [†]	iNat2017			iNat2018		
	Test	Test	Test	Test	Val	Test Pu	Test Pr	Val	Test Pu	Test Pr
No Prior (i.e. uniform)	50.15	70.07	70.07	76.08	63.27	64.16	63.63	60.20	50.17	50.33
Nearest Neighbor (num)	51.78	70.82	77.76	79.99	65.34	66.04	65.61	68.70	54.54	54.58
Nearest Neighbor (spatial)	51.21	71.57	77.98	80.79	65.85	67.02	66.41	67.55	53.67	53.81
Discretized Grid	51.06	71.09	77.19	79.58	65.49	66.62	66.07	67.27	53.13	53.16
Adaptive Kernel [3]	51.47	71.57	78.65	81.11	64.86	65.83	65.59	65.23	53.17	53.21
Tang et al. [55]	50.43	70.16	72.33	77.34	66.15	67.08	66.53	65.61	54.12	54.25
Ours no date	50.70	71.66	78.65	81.15	69.34	70.62	70.18	72.41	57.68	57.84
Ours full	-	71.84	79.58	81.50	69.60	70.83	70.51	72.68	58.44	58.59

Table 7.1: **Classification accuracy.** Results after combining image classification predictions $P(y|I)$ with different spatio-temporal priors $P(y|\mathbf{x})$. All results are top 1 accuracy with classifier predictions extracted from an InceptionV3 [54] network fine-tuned on each of the respective datasets. [†] indicates that simulated locations, dates, and photographers from the eBird dataset [53] are used. The baseline algorithms do not use date information.

λ is a hyperparameter used to weight the positive observations and \mathbf{r} is a uniformly random spatio-temporal datapoint. Next, we want the affinity between a photographer p and a location \mathbf{x} be high if p was present at \mathbf{x} , and low otherwise:

$$\begin{aligned} \mathcal{L}_{p_loc}(\mathbf{x}, \mathbf{r}, \mathbf{P}, p) = & \log(s(f(\mathbf{x})\mathbf{P}_{:,p})) + \\ & \log(1 - s(f(\mathbf{r})\mathbf{P}_{:,p})). \end{aligned} \quad (7.8)$$

We assume that a photographer has a low affinity for a category unless they have previously observed it:

$$\begin{aligned} \mathcal{L}_{p_o}(\mathbf{O}, \mathbf{P}, y, p) = & \lambda \log(s(\mathbf{O}_{:,y}^T \mathbf{P}_{:,p})) + \\ & \sum_{\substack{i=1 \\ i \neq y}}^C \log(1 - s(\mathbf{O}_{:,i}^T \mathbf{P}_{:,p})). \end{aligned} \quad (7.9)$$

Finally, to estimate the parameters of our prior we maximize

$$\mathcal{L} = \mathcal{L}_{o_loc} + \mathcal{L}_{p_loc} + \mathcal{L}_{p_o}, \quad (7.10)$$

by iterating over each of the datapoints in the training set.

7.5 Experiments

We evaluate the effectiveness of our spatio-temporal prior by performing experiments on several image classification datasets that have location and time information. We choose image classification because for other domains (e.g. species distribution modeling) it is challenging to obtain accurate ground truth information regarding the true spatio-temporal distributions of the categories of interest.

	Top1	Top3	Top5
iNat2017 - InceptionV3 299 × 299			
No Prior (i.e. uniform)	63.27	79.82	84.51
Ours no wrap encode	69.48	84.43	88.15
Ours no photographer	69.39	83.97	87.71
Ours no date	69.34	84.16	87.89
Ours full	69.60	84.41	88.07
iNat2018 - InceptionV3 299 × 299			
No Prior (i.e. uniform)	60.20	77.90	83.29
Ours no wrap encode	72.12	87.00	90.52
Ours no photographer	72.84	87.30	90.75
Ours no date	72.41	87.19	90.60
Ours full	72.68	87.26	90.79
iNat2018 - InceptionV3 520 × 520			
No Prior (i.e. uniform)	66.18	83.32	88.04
Ours no wrap encode	77.09	90.68	93.54
Ours no photographer	77.64	90.82	93.52
Ours no date	77.41	90.80	93.58
Ours full	77.49	90.85	93.57

Table 7.2: **Ablation.** Classification accuracy for different variants of our prior on the iNat2017 and iNat2018 [58] validation sets. In the case of iNat2018, we still observe improvements when combining our prior with a more powerful image classifier - see rows ‘InceptionV3 520 × 520’.

Datasets

While location metadata is readily available for online image collections, many popular image classification datasets do not contain this information e.g. [61, 57, 14, 39]. Some datasets exist with location information, but for only a subset of the images e.g. [24]. However, datasets containing images of different species of plants and animals are available with location, time, and photographer information. To this end, we perform experiments on the iNaturalist 2017 and 2018 (iNat2017 and iNat2018) species classification datasets which contain images collected and annotated by citizen scientists [58]. They have 5,089 and 8,142 categories respectively. While [3] evaluated their location prior on the BirdSnap dataset, the images and location metadata used are not provided by the authors. We recollect the images and location data from the web using the original image URLs. Despite the dataset consisting of images of species commonly found in North America, when we recollected the images and locations we found that the original images are from all over the world and 40% were missing location. Like [3], we also simulate location metadata for BirdSnap [3] and another fine-grained dataset of birds, NABirds [57], by associating each image with a species observation from eBird [53]. Our train locations and photographers are sampled from eBird 2015, and the test set is from 2016. BirdSnap and NABirds contain images from 500 and 555 different species of North America

birds. Finally, we also perform experiments on YFCC100M-GEO100 [55] (YFCC). YFCC contains 100 everyday object categories with associated locations, but no date or photographer information is provided. The train and test split used in [55] is not available and so we created a new one. Unlike the other datasets, many of the object categories in YFCC are not geographically distinct e.g. ‘band’, ‘ford’, or ‘ipod’.

Implementation Details

Our location encoder $f()$ is a fully-connected neural network consisting of an input layer, followed by multiple residual layers [29], and a final output embedding layer. We jointly train the location encoder, along with the photographer and object embeddings using Adam [36] for 30 epochs with a batch size of 1024, using dropout to prevent overfitting. The dimensionality of the shared embedding space is set to $D = 256$. When weighting the positive instances during training we set λ to the number of categories. To counteract the heavily imbalanced nature of many of the datasets, we limit the maximum number of datapoints for each category per epoch. We set the maximum number of datapoints to 100, and for each epoch we randomly select a different subset for each category. The only exception is for YFCC, where capping the data hurt performance. Details of our network architecture are in the supplementary material.

Except where noted, at test time, our model takes three inputs, longitude, latitude, and day of the year, specifying where and when the image of interest was captured. For these three input features \mathbf{x} we explored different methods for ‘wrapping’ the coordinates, i.e. an observation taken on December 31st should result in a similar embedding to one captured on January 1st. Similarly, we want geographical coordinates to wrap around the earth. To achieve this, for each input dimension l of \mathbf{x} we perform the mapping $[\sin(\pi x^l), \cos(\pi x^l)]$, resulting in two numbers for each dimension. Here, we assume that each dimension of the input has been normalized to the range $x^l \in [-1, 1]$.

For the image classifiers $P(y|I)$ we fine-tune a separate InceptionV3 [54] network for each of the datasets beginning with ImageNet initialized weights [14] with an image resolution of 299×299 (unless otherwise noted).

Quantitative Evaluation

In Table 7.1 we evaluate how much our spatio-temporal prior improves image classification performance by comparing it to several baselines. We found that adding a

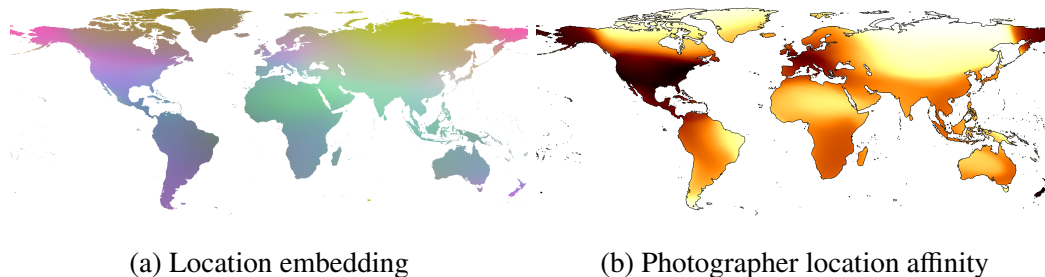


Figure 7.3: **Spatial predictions.** (a) Embeddings for each location on the earth for a model trained on iNat2018 [58]. We observe that the embeddings appears to capture information related to climate zones, despite not being trained on any climate data. (b) Log plot of estimated photographer location preferences. Darker colors indicate that more photographers have captured images in those locations. We can see that there is a large bias towards North America, Europe, and New Zealand.

uniform prior to the outputs of the nearest neighbor based baselines increases their performance. This adds robustness in cases where there are no objects from the training set present near the test locations. The lack of this uniform prior explains the poor results for nearest neighbor based approaches in [55]. For the comparison to Tang et al. [55], we jointly train a linear layer to embed the raw location information along with an output layer to combine the location embedding with the features from the last linear layer of the image classifier. The rest of the weights of the image classifier are not updated. For each of the baseline algorithms we select their hyperparameters (e.g. the number of neighbors) on a held out validation set for each dataset. When location information is not available at test time, we assume a uniform prior over the categories.

Our model performs on par, or better, than the baselines across all datasets. The advantage of our approach is that it is computationally efficient at test time and does not require features from the image classifier during training. Compared to nearest neighbor based methods, it only requires a forward pass through a compact fully-connected neural network. In addition, it also captures structural information such as object and photographer biases. One failure case that is worth noting are the results on YFCC [55]. We observe that all methods perform similar to using no location information (No Prior). This can be explained by the relative lack of spatio-temporal structure in the object categories present in the dataset. Again, this is consistent with the findings in [55], where the authors had to use additional features to increase the performance.

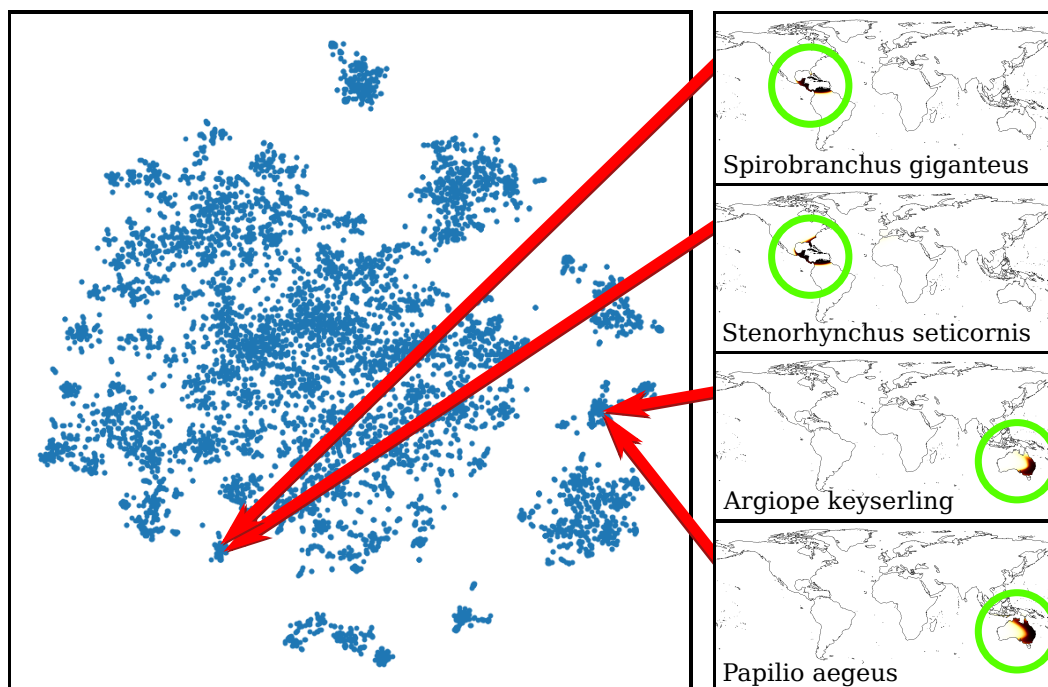


Figure 7.4: **Object embedding.** t-SNE [42] plot of the learned embedding \mathbf{O} for all 8,142 categories from iNat2018 [58]. The location in the object embedding space encodes a category’s preferences for a particular geographical region. We observe that categories that have similar spatio-temporal distributions tend to be close.

Ablation Study

In Table 7.2 we compare the performance of different variants of our model on iNat2017 and iNat2018 [58]. Again, across all metrics there is a large increase in performance compared to the baseline uniform prior. In some cases, we even observe that there is an additional boost in performance when we explicitly model photographer biases.

Training fine-grained image classifiers with larger input images can significantly increase classification performance [13]. We observe that the benefit of our spatio-temporal prior is still apparent even when we use a more powerful classifier that has been training for longer with larger images. This increase in accuracy is also present when we evaluate performance using more lenient evaluation metrics i.e. top 5 vs. top 1 accuracy. This is significant because it highlights that for some datasets the performance boost provided by the spatio-temporal prior is orthogonal to improvements in the underlying image classifier.

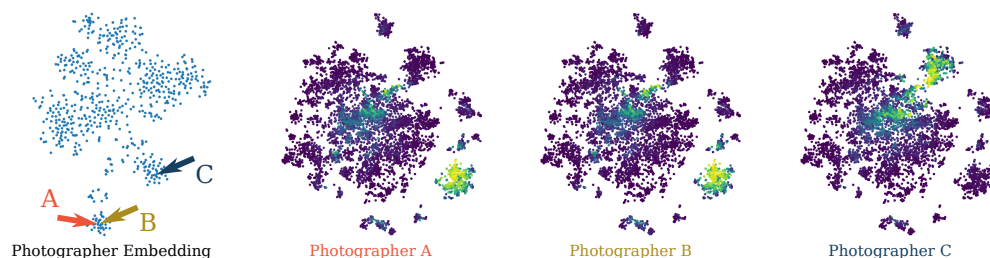


Figure 7.5: **Photographer object affinity.** On the left we see a t-SNE [42] plot of the photographer embeddings \mathbf{P} for iNat2018 [58]. The three plots on the right depict the predicted affinities for three different photographers (A, B, and C) visualized on the category embedding from Fig. 7.4. Brighter colors indicate a higher affinity for a given category. We observe that individuals that are close in the photographer embedding space \mathbf{P} (e.g. A and B) have similar category affinities, compared to those that are far away (e.g. C).

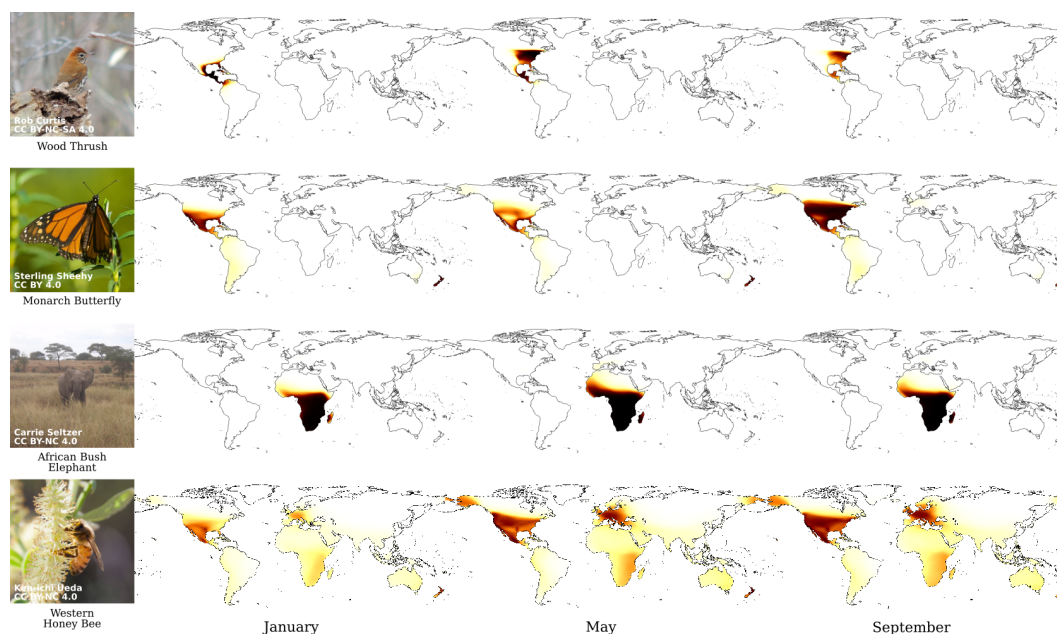


Figure 7.6: **Spatio-temporal predictions.** Predicted distributions for several object categories for three different time points using our full model trained on iNat2018 [58]. Darker colors indicate locations where the categories are predicted to be found. In the first two rows we observe that our model captures seasonal migratory behaviors. On the bottom row, our model correctly predicts that the Western Honey Bee can be found on several different continents. It is worth noting that the results are affected by geographical sampling biases in the iNat2018 dataset.

Qualitative Evaluation

Our model captures the relationship between objects, locations, and photographers. In Fig. 7.3 (a) we can see the resulting embeddings for each input location from our model trained on iNat2018 [58]. By applying the embedding function $f()$ to each location we can generate its D dimensional embedding vector. We then use ICA [32] to project the embedded features to a three dimensional space and mask out the ocean for visualization. Perhaps as expected, there is low frequency structure in the resulting image, i.e. nearby locations tend to support similar objects. One advantage of our approach is that we are not restricted to a fixed discretization. As a result we can generate embeddings for any location and time. In Fig. 7.4 we visualize our learned object embedding \mathbf{O} . Objects that have similar spatio-temporal distributions tend to result in similar embedding vectors.

Distinct from other work, our prior also models the relationship between photographers and locations, and photographers and object categories. In Fig. 7.3 (b) we plot the estimated affinity for each input location across all photographers i.e. $\sum_p s(f(\mathbf{x})\mathbf{P}_{:,p})$. We only show results for photographers who provided at least 100 observations in the iNat2018 [58] training set, resulting in 634 individuals. In Fig. 7.5 we display the estimated affinity for each object category for a set of photographers, i.e. $P(y|p) \propto s(\mathbf{O}^T\mathbf{P})$. We observe that the embedding captures the similarity in object affinity held by different photographers.

Finally, in Fig. 7.6 we use our prior to generate spatio-temporal predictions for several different species from iNat2018 [58]. Each image is generated by querying every location on the surface of the earth, on a specified day of the year, to generate $P(y = y^*|\mathbf{x})$ for the category of interest. In practice, we evaluate 1000×2000 spatial locations for each time point (e.g. first day of the month). This step is very efficient as we can pre-compute $f(\mathbf{x})$ for every location, independent of the category of interest. Again, for visualization we mask out the predictions over the ocean.

Limitations

We are limited by the quality of the provided location data e.g. it can be inaccurate or intentionally obfuscated. We also make strong assumptions about a photographer’s affinity for an individual object category. In reality, these interactions may be complex i.e. once a photographer captures an image of a particular category they may be less likely to take an image of the same object in the near future. There are also known spatial biases in the types of citizen science data we use [2, 8].

However, this may not be a major issue as we can assume that the distribution of test locations and dates is similarly biased. We currently only use location, time, and photographer ID during training. In practice, additional data such as environmental variables may be a valuable signal for specific object categories [5].

7.6 Conclusion

We introduce a spatio-temporal prior to help disambiguate fine-grained categories resulting in improved test time image classification performance. In addition to helping image classification, our model also naturally captures the relationships between locations and objects, objects and objects, photographers and objects, and photographers and locations in an interpretable manner. Importantly, our prior is efficient at test time, both in terms of model size and inference speed, and scales to large numbers of categories.

7.7 Acknowledgements

This work was supported by a Google Focused Research Award and an NSF Graduate Research Fellowship (Grant No. DGE-1745301). We thank Grant Van Horn and Serge Belongie for helpful discussions, along with NVIDIA and AWS for their kind donations.

References

- [1] Morgane Barbet-Massin et al. “Selecting pseudo-absences for species distribution models: how, where and how many?” In: *Methods in Ecology and Evolution* (2012).
- [2] Jan Beck et al. “Spatial bias in the GBIF database and its effect on modeling species’ geographic distributions”. In: *Ecological Informatics* (2014).
- [3] Thomas Berg et al. “Birdsnap: Large-scale fine-grained visual categorization of birds”. In: *CVPR*. 2014.
- [4] Christophe Botella et al. “A Deep Learning Approach to Species Distribution Modelling”. In: *Multimedia Tools and Applications for Environmental & Biodiversity Informatics*. 2018.
- [5] Christophe Botella et al. “Overview of GeoLifeCLEF 2018: location-based species recommendation”. In: (2018).
- [6] Matthew R. Boutell et al. “Learning multi-label scene classification”. In: *Pattern Recognition* (2004).
- [7] Steve Branson et al. “Bird species categorization using pose normalized deep convolutional nets”. In: *BMVC*. 2014.

- [8] Di Chen and Carla P Gomes. “Bias Reduction via End-to-End Shift Learning: Application to Citizen Science”. In: *AAAI*. 2019.
- [9] Di Chen et al. “Deep multi-species embedding”. In: *IJCAI*. 2017.
- [10] Yao-nan Chen and Hsuan-tien Lin. “Feature-aware Label Space Dimension Reduction for Multi-label Classification”. In: *NeurIPS*. 2012.
- [11] Grace Chu et al. “Geo-aware networks for fine-grained recognition”. In: *ICCV Workshops*. 2019.
- [12] Yin Cui et al. “Class-Balanced Loss Based on Effective Number of Samples”. In: *CVPR*. 2019.
- [13] Yin Cui et al. “Large scale fine-grained categorization and domain-specific transfer learning”. In: *CVPR*. 2018.
- [14] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *CVPR*. 2009.
- [15] Abhimanyu Dubey et al. “Pairwise confusion for fine-grained visual classification”. In: *ECCV*. 2018.
- [16] Jane Elith and Catherine H. Graham. “Do they? How do they? WHY do they differ? On finding reasons for differing performances of species distribution models”. In: *Ecography* (2009).
- [17] Robin Engler, Antoine Guisan, and Luca Rechsteiner. “An improved approach for predicting the distribution of rare and endangered species from occurrence and pseudo-absence data”. In: *Journal of Applied Ecology* (2004).
- [18] Daniel Fink et al. “Spatiotemporal exploratory models for broad-scale survey data”. In: *Ecological Applications* (2010).
- [19] William Fithian and Rahul Mazumder. “Flexible Low-Rank Statistical Modeling with Missing Data and Side Information”. In: *Statistical Science* (2018).
- [20] Simone Franceschini et al. “Cascaded neural networks improving fish species prediction accuracy: the role of the biotic information”. In: *Scientific Reports* (2018).
- [21] Yang Gao et al. “Compact bilinear pooling”. In: *CVPR*. 2016.
- [22] Georgia E Garrard et al. “A general model of detectability using species traits”. In: *Methods in Ecology and Evolution* (2013).
- [23] “GBIF - www.gbif.org”. In: (2019).
- [24] Hervé Goëau, Pierre Bonnet, and Alexis Joly. “Plant Identification in an Open-world (LifeCLEF 2016)”. In: *CLEF: Conference and Labs of the Evaluation Forum*. 2016.

- [25] Nick Golding and Bethan V. Purse. “Fast and flexible Bayesian species distribution modelling using Gaussian processes”. In: *Methods in Ecology and Evolution* (2016).
- [26] David J. Harris. “Generating realistic assemblages with a joint species distribution model”. In: *Methods in Ecology and Evolution* (2015).
- [27] Trevor Hastie and Will Fithian. “Inference from presence-only data; the ongoing controversy”. In: *Ecography* (2013).
- [28] James Hays and Alexei A Efros. “IM2GPS: estimating geographic information from a single image”. In: *CVPR*. 2008.
- [29] Kaiming He et al. “Deep residual learning for image recognition”. In: *CVPR*. 2016.
- [30] Troy M. Hegel et al. “Current State of the Art for Statistical Modelling of Species Distributions”. In: *Spatial Complexity, Informatics, and Wildlife Conservation*. 2010.
- [31] Shaoli Huang et al. “Part-stacked CNN for fine-grained visual categorization”. In: *CVPR*. 2016.
- [32] Aapo Hyvarinen and Erkki Oja. “Independent component analysis: algorithms and applications”. In: *Neural networks* (2000).
- [33] *iNaturalist*. www.inaturalist.org, accessed Mar 7 2022.
- [34] Julia Jones, Jeffrey Miller, and Matt White. “Multi-label classification for multi-species distribution modeling”. In: *ICML*. 2011.
- [35] Aditya Khosla et al. “Novel dataset for fine-grained image categorization: Stanford dogs”. In: *CVPR Workshop on Fine-Grained Visual Categorization*. 2011.
- [36] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv* (2014).
- [37] Jonathan Krause et al. “The unreasonable effectiveness of noisy data for fine-grained recognition”. In: *ECCV*. 2016.
- [38] Alex M. Lechner et al. “Investigating species-environment relationships at multiple scales: Differentiating between intrinsic scale and the modifiable areal unit problem”. In: *Ecological Complexity* (2012).
- [39] Tsung-Yi Lin et al. “Microsoft COCO: Common objects in context”. In: *ECCV*. 2014.
- [40] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhansu Maji. “Bilinear CNN models for fine-grained visual recognition”. In: *ICCV*. 2015.
- [41] Jiongxin Liu et al. “Dog breed classification using part localization”. In: *ECCV*. 2012.

- [42] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *JMLR* (2008).
- [43] Darryl I MacKenzie et al. “Estimating site occupancy rates when detection probabilities are less than one”. In: *Ecology* (2002).
- [44] Darryl I. MacKenzie. “What are the Issues with Presence-Absence Data for Wildlife Managers?” In: *The Journal of Wildlife Management* (2005).
- [45] Sylvain Mastrorillo et al. “The use of artificial neural networks to predict the presence of small-bodied fish in a river”. In: *Freshwater Biology* (2003).
- [46] Justin Moat et al. “Refining area of occupancy to address the modifiable areal unit problem in ecology and conservation”. In: *Conservation Biology* (2018).
- [47] Anna Norberg et al. “A comprehensive evaluation of predictive performance of 33 species distribution models at species and community levels”. In: *Ecological Monographs* 89.3 (2019), e01370.
- [48] Stan Openshaw. “The Modifiable Areal Unit Problem”. In: Norwich, England: Geobooks, 1983.
- [49] Stacy L. Özesmi and Uygur Özesmi. “An artificial neural network approach to spatial habitat modelling with interspecific interaction”. In: *Ecological Modelling* (1999).
- [50] Steven J Phillips, Miroslav Dudik, and Robert E Schapire. “A maximum entropy approach to species distribution modeling”. In: *ICML*. 2004.
- [51] Steven J Phillips et al. “Sample selection bias and presence-only distribution models: implications for background and pseudo-absence data”. In: *Ecological Applications* (2009).
- [52] Laura J. Pollock et al. “Understanding co-occurrence by modelling species simultaneously with a Joint Species Distribution Model”. In: *Methods in Ecology and Evolution* (2014).
- [53] Brian L Sullivan et al. “eBird: A citizen-based bird observation network in the biological sciences”. In: *Biological Conservation* (2009).
- [54] Christian Szegedy et al. “Rethinking the inception architecture for computer vision”. In: *CVPR*. 2016.
- [55] Kevin Tang et al. “Improving image classification with location context”. In: *ICCV*. 2015.
- [56] Luming Tang et al. “Multi-Entity Dependence Learning with Rich Context via Conditional Variational Auto-encoder”. In: *AAAI*. 2018.
- [57] Grant Van Horn et al. “Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection”. In: *CVPR*. 2015.

- [58] Grant Van Horn et al. “The iNaturalist species classification and detection dataset”. In: *CVPR*. 2018.
- [59] Kumar Vishal, CV Jawahar, and Visesh Chari. “Accurate localization by fusing images and GPS signals”. In: *CVPR Workshop*. 2015.
- [60] Nam Vo, Nathan Jacobs, and James Hays. “Revisiting IM2GPS in the deep learning era”. In: *ICCV*. 2017.
- [61] Catherine Wah et al. “The caltech-ucsd birds-200-2011 dataset”. In: (2011).
- [62] Fei Wang et al. “Residual attention network for image classification”. In: *CVPR*. 2017.
- [63] Jiang Wang et al. “CNN-RNN: A Unified Framework for Multi-Label Image Classification”. In: *CVPR*. 2016.
- [64] Hans Christian Wittich et al. “Recommending plant taxa for supporting on-site species identification”. In: *BMC Bioinformatics* (2018).
- [65] Tianjun Xiao et al. “The application of two-level attention models in deep convolutional neural network for fine-grained image classification”. In: *CVPR*. 2015.
- [66] Linjie Yang et al. “A large-scale car dataset for fine-grained categorization and verification”. In: *CVPR*. 2015.
- [67] Peggy PW Yen, Falk Huettmann, and Fred Cooke. “A large-scale model for the at-sea distribution and abundance of Marbled Murrelets (*Brachyramphus marmoratus*) during the breeding season in coastal British Columbia, Canada”. In: *Ecological Modelling* (2004).
- [68] Menghua Zhai et al. “Learning Geo-Temporal Image Features”. In: *BMVC*. 2018.
- [69] Min-Ling Zhang and Zhi-Hua Zhou. “Multi-Label Neural Networks with Applications to Functional Genomics and Text Categorization”. In: *IEEE Transactions on Knowledge and Data Engineering* (2006).
- [70] Ning Zhang et al. “Deformable part descriptors for fine-grained recognition and attribute prediction”. In: *ICCV*. 2013.
- [71] Ning Zhang et al. “Part-based R-CNNs for fine-grained category detection”. In: *ECCV*. 2014.
- [72] Bo Zhao et al. “Diversified visual attention networks for fine-grained object classification”. In: *IEEE Transactions on Multimedia* (2017).

SPATIAL IMPLICIT NEURAL REPRESENTATIONS FOR GLOBAL-SCALE SPECIES MAPPING

- [1] Elijah Cole, Grant Van Horn, Christian Lange, Alexander Shepard, Patrick Leary, Scott Loarie, Pietro Perona, and Oisín Mac Aodha. “Spatial Implicit Neural Representations for Global-Scale Species Mapping”. In: *International Conference on Machine Learning*. 2023.

8.1 Abstract

Estimating the geographical range of a species from sparse observations is a challenging and important geospatial prediction problem. Given a set of locations where a species has been observed, the goal is to build a model to predict whether the species is present or absent at any location. This problem has a long history in ecology, but traditional methods struggle to take advantage of emerging large-scale crowdsourced datasets which can include tens of millions of records for hundreds of thousands of species. In this work, we use Spatial Implicit Neural Representations (SINRs) to jointly estimate the geographical range of 47k species simultaneously. We find that our approach scales gracefully, making increasingly better predictions as we increase the number of species and the amount of data per species when training. To make this problem accessible to machine learning researchers, we provide four new benchmarks that measure different aspects of species range estimation and spatial representation learning. Using these benchmarks, we demonstrate that noisy and biased crowdsourced data can be combined with implicit neural representations to approximate expert-developed range maps for many species.

8.2 Introduction

We are currently observing a dramatic decline in global biodiversity, which has severe ramifications for natural resource management, food security, and ecosystem services that are crucial to human health [42, 31]. In order to take effective conservation action we must understand species’ ranges, i.e. where they live. However, we only have estimated ranges for a relatively small number of species in limited areas, many of which are already out of date by the time they are released.

The range of a species is typically estimated through *Species Distribution Modeling*

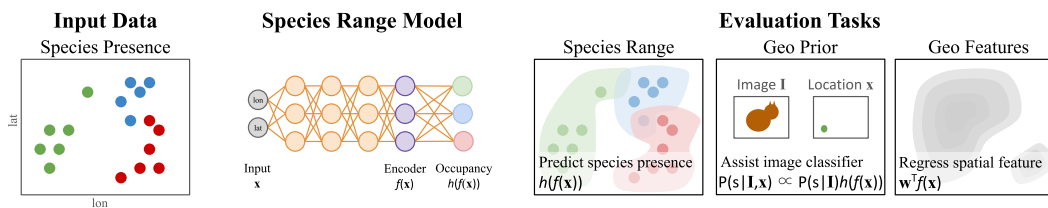


Figure 8.1: We show that sparse species observation data can be used to train Spatial Implicit Neural Representations (SINRs) which are transferable to other geospatial tasks. (*Left*) Here we show sparse, presence-only, spatial observations for three toy species (red, green, and blue). (*Middle*) The species observations are used to train a neural network that consists of a spatial feature encoder and per-species presence predictors. (*Right*) We evaluate on three diverse tasks: (i) estimating species ranges, (ii) assisting image classifiers using geographical range priors, and (iii) regressing geospatial features via our learned SINR.

(SDM) [11], the process of using species observation records to develop a statistical model for predicting whether a species is present or absent at any location. With enough *presence-absence* data (i.e. records of where a species has been confirmed to be present and absent) this problem can be approached using standard statistical learning methods [2].¹ However, presence-absence data is scarce due to the difficulty of verifying that a species is truly absent from an area. *Presence-only* data (i.e. verified observation locations, with no confirmed absences) is much more abundant as it is easier to collect. For instance, the community science platform iNaturalist [17] has collected over 141M presence-only observations to date across 429k species. Though presence-only data is not without drawbacks [16], it is important to develop methods that can take advantage of this vast supply of data.

Deep learning is one of our best tools for making use of large-scale datasets. Deep neural networks also have a key advantage over many existing SDM methods because they can *jointly* learn the distribution of many species in the same model [7, 36, 20]. By learning representations that share information across species, the models can make improved predictions [7]. However, the majority of current deep learning approaches need presence-absence data for training, which prevents them from scaling beyond the small number of species and regions for which sufficient presence-absence data is available.

Our work makes the following contributions:

(i) We show that implicit neural representations trained with noisy crowdsourced

¹The term "presence-absence" should not be taken to convey absolute certainty about whether a species is present or absent. False absences (i.e. non-detections) and, to a lesser extent, false presences are a serious concern in SDM [21].

presence-only data can be used to estimate dense species' ranges. We call these models Spatial Implicit Neural Representations (SINRs).²

(ii) We conduct a detailed investigation of loss functions for learning from presence-only data, their scaling properties, and the resulting geospatial representations.

(iii) We provide a suite of four geospatial benchmark tasks — ranging from species mapping to fine-grained image classification — which will facilitate future research on spatially sparse high-dimensional implicit neural representations, large-scale SDM, and geospatial representation learning.

Training and evaluation code is available at:

<https://github.com/elijahcole/sinr>

8.3 Related Work

Species distribution modeling (SDM) refers to a set of methods that aim to predict where (and sometimes when, and in what quantities) species of interest are likely to be found [11]. The literature on SDM is vast. Readers interested in an overview should consult the review by [11] or the recent review of SDM for computer scientists by [2]. Note that we focus narrowly on the problem of predicting the occurrence of a species at a location, i.e. we do not consider more complex problems like trend or abundance estimation [29].

Traditional approaches to SDM train conventional supervised learning models (e.g. logistic regressors [26], random forests [10], etc.) to learn a mapping between hand-selected sets of environmental features (e.g. altitude, average rainfall, etc.) and species presence or absence [27, 12]. Readers interested in these approaches should consult [25, 39, 40], and the references therein. More recently, deep learning methods have been introduced that instead *jointly* represent multiple different species within the same model [7, 4, 36, 20, 37]. These models are typically trained on crowdsourced data, which can introduce additional challenges and biases that need to be accounted for during training [14, 6, 19, 5]. We build on the work of [20], who proposed a neural network approach that forgoes the need for environmental features (as used by e.g. [4, 36]) by learning to predict species presence from geographical location alone.

The problem of joint SDM with presence-only data can be viewed as an instance of multi-label classification with incomplete supervision. In particular, it is an

²We slightly abuse the terminology by using "SINR" to refer to both the model and the representation it parameterizes.

example of Single Positive Multi-Label (SPML) learning [9, 41, 45]. The goal is to train a model that is capable of making multi-label predictions at test time, despite having only ever observed one positive label per training instance (i.e. no confirmed negative training labels). Our work connects the SPML literature and SDM literature, and sets up large-scale joint species distribution modeling as a challenging real-world SPML task. This setting presents significant new difficulties for SPML, which has largely been limited to artificial label bias patterns [1] and relatively small label spaces (< 100 categories). Some SPML methods such as ROLE [9] are not computationally viable when the label space is large. One of our baselines is based on the SPML method of [45], which is scalable and obtains nearly state-of-the-art performance on the standard SPML benchmarks [9], but it is not a top performer on our new benchmark tasks.

Our work is related to the growing number of papers that use coordinate neural networks for implicitly representing images [34] and 3D scenes [32, 24]. There are many design choices in these methods that are being actively studied, including the impact of the activation functions in the network [32, 30] and the effect of different input encodings [34, 44]. In most research on implicit neural representations, there is an obvious choice of training objective, e.g. mean squared error between the predictions and the data. In the context of presence-only species estimation, this choice is less clear. We systematically investigate this question in our experiments. Our benchmark also facilitates investigations of implicit neural representations with high-dimensional output spaces and sparse supervision.

Quantifying the performance of SDM at scale is notoriously difficult due to the fact that we lack confirmed presence-absence data for most species and locations [2]. One approach is to evaluate performance on a small set of species from limited geographical regions where it is feasible to collect presence-absence data, as done in e.g. [29, 25, 40]. Two of our evaluation tasks are larger-scale versions of this idea, in which we compare the performance of our models against expert range maps. An alternative evaluation approach is to measure the performance on a related "proxy" task. For example, there have been a number of works that use models trained for species range estimation to assist deep image classifiers [3, 35, 20, 8, 22, 38, 33, 43]. By using images from platforms like iNaturalist, we can evaluate different range estimation methods on the task of aiding fine-grained image classification across tens of thousands of species. Finally, we also evaluate the spatial representations learned by our models via transfer learning, using them as inputs for a set of geospatial

regression tasks. These complementary benchmark tasks capture different aspects of performance, and provide a starting point for large-scale SDM evaluation. See Figure 8.1 for an overview of our tasks.

8.4 Methods

Preliminaries

Problem statement. Let $\mathbf{x} = [lon, lat]$ denote a geographical location (i.e. longitude and latitude). Let $\mathbf{y} \in \{0, 1\}^S$ denote the true presence (1) or absence (0) of S different species at location \mathbf{x} . Following [9], we introduce $\mathbf{z} \in \{0, 1, \emptyset\}^S$ to represent our observed data at \mathbf{x} , where $z_j = 1$ if species j is present, $z_j = 0$ if species j is absent, and $z_j = \emptyset$ if we do not know whether species j is present or absent. Our goal is to develop a model that produces an estimate of \mathbf{y} at any location \mathbf{x} over some spatial domain \mathcal{X} , given observed data $\{(\mathbf{x}_i, \mathbf{z}_i)\}_{i=1}^N$. We parameterize this model as $\hat{\mathbf{y}} = h_\phi(f_\theta(\mathbf{x}))$, where $f_\theta : \mathcal{X} \rightarrow \mathbb{R}^k$ is a location encoder with parameters θ and $h_\phi : \mathbb{R}^k \rightarrow [0, 1]^S$ is a multi-label classifier with parameters ϕ . The prediction $\hat{\mathbf{y}} \in [0, 1]^S$ is our estimate of how likely each species is to be present at \mathbf{x} .

Intuitively, the location encoder f_θ provides a representation of geographical space that is used by the multi-label classifier h_ϕ to predict species presence at each location. If θ is fixed or if f is a differentiable function of θ , then we can use standard methods like stochastic gradient descent to approximately solve

$$\theta^*, \phi^* = \operatorname{argmin}_{\theta, \phi} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(\hat{\mathbf{y}}_i, \mathbf{z}_i) \quad (8.1)$$

where $\hat{\mathbf{y}}_i = h_\phi(f_\theta(\mathbf{x}_i))$ and \mathcal{L} is a suitably chosen loss function. Once trained, we say that $h_\phi \circ f_\theta$ has learned a Spatial Implicit Neural Representation (SINR) for the distribution of each species in the training set. Along the way we can learn f_θ , which produces a representation for any location on earth. See Figure 8.3 for visualizations of some of these geospatial representations.

Input encoding. Each species observation is associated with spatial coordinates $\mathbf{x} = [lon, lat]$. In practice, we rescale these values so that $lon, lat \in [-1, 1]$ and, following [20], we guard against boundary effects using a sinusoidal encoding. The results is an input vector

$$\mathbf{x} = [\sin(\pi lon), \cos(\pi lon), \sin(\pi lat), \cos(\pi lat)]. \quad (8.2)$$

Alternative input encodings for related coordinate networks have been explored in the existing literature [22, 34, 23, 44]. This choice is orthogonal to the losses we explore, so we leave the evaluation of input encodings to future work.

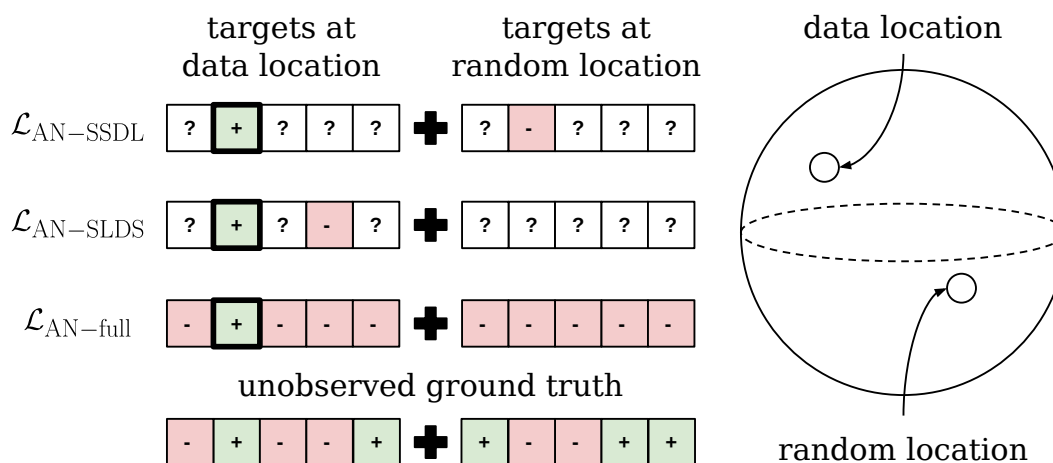


Figure 8.2: Illustration of the data used by three loss functions from Section 8.4. For each loss, we visualize the targets that the network is trained to predict. Each loss can be broken into two parts: one part that updates the network’s predictions at the location of a training example (*data location*) and one part that updates the network’s predictions at another location chosen randomly (*random location*). Each loss has access to one confirmed positive label (bold boxes). The rest of the labels are unobserved (non-bold boxes), and the losses make different, imperfect, assumptions about those unobserved labels.

Implicit neural representations. Traditionally, representation learning aims to transform complex objects (e.g. images, text) into simpler objects (e.g. low-dimensional vectors) that facilitate downstream tasks like classification or regression [15]. Implicit neural representations offer a different perspective, in which a signal is represented by a neural network that maps the signal domain (e.g. \mathbb{R} for audio, \mathbb{R}^2 for images) to the signal values [32, 34]. In this work we learn implicit neural representations from a large collection of crowdsourced data containing observations of many species. This yields an implicit neural representation for the geospatial distribution of each species, as well as a representation for any location on earth.

Presence-absence vs. presence-only data. Species observation datasets come in two varieties: (i) *Presence-absence* data consists of locations where a species has been observed to be present and locations where it has been confirmed to be absent. That is, we say we have presence-absence data for species j if $|\{\mathbf{z}_i : z_{ij} = 0\}| > 0$ and $|\{\mathbf{z}_i : z_{ij} = 1\}| > 0$. Unfortunately, presence-absence data is costly to obtain at scale because confirming absence requires skilled observers to exhaustively search an area. (ii) *Presence-only* data is easier to acquire and thus more abundant because absences are not collected, i.e. $z_{ij} \in \{1, \emptyset\}$, for $i \in [N]$ and $j \in [S]$.

Learning from Large-Scale Presence-Only Data

In the context of training SPML *image* classifiers, a simple but effective approach is to assume that unobserved labels are negative [9]. This approach is based on a probabilistic argument: since natural images tend to contain a small number of categories compared to the size of the label set, the vast majority of the labels will be negative. This is also true for species distribution modeling. Given an arbitrary location and a large set of candidate species, nearly all of them will be absent. In this section we describe several simple and scalable loss functions based on this idea. We illustrate three of our losses in Figure 8.2.

"Assume negative" loss (same species, different location). As confirmed absences are not available in the presence-only setting, a common approach is to use randomly generated "pseudo-negatives" [28]. This first loss pairs each observation of a species with a pseudo-negative for that species at another location chosen uniformly at random:

$$\mathcal{L}_{\text{AN-SSDL}}(\hat{\mathbf{y}}, \mathbf{z}) = -\frac{1}{n_{\text{pos}}} \sum_{j=1}^S \mathbb{1}_{[z_j=1]} [\log(\hat{y}_j) + \log(1 - \hat{y}'_j)] \quad (8.3)$$

where $\hat{\mathbf{y}}' = h_\phi(f_\theta(\mathbf{r}))$ with $\mathbf{r} \sim \text{Uniform}(\mathcal{X})$ and $n_{\text{pos}} = \sum_{j=1}^S \mathbb{1}_{[z_j=1]}$. This approach generates pseudo-negatives (i.e. random absences) across the globe, but many of them are likely to be "easy" because they are far from the true species range.

"Assume negative" loss (same location, different species). This loss pairs each observation of a species with a pseudo-negative at the same location for a different species:

$$\mathcal{L}_{\text{AN-SLDS}}(\hat{\mathbf{y}}, \mathbf{z}) = -\frac{1}{n_{\text{pos}}} \sum_{j=1}^S \mathbb{1}_{[z_j=1]} [\log(\hat{y}_j) + \log(1 - \hat{y}'_{j'})] \quad (8.4)$$

where $j' \sim \text{Uniform}(\{j : z_j \neq 1\})$. Intuitively, this approach generates pseudo-negatives that are aligned with the spatial distribution of the observed data.

Full "assume negative" loss. The previous two losses are inefficient in the sense that they do not use all of the entries in $\hat{\mathbf{y}}$. We can combine the pseudo-negative sampling strategies of $\mathcal{L}_{\text{AN-SSDL}}$ and $\mathcal{L}_{\text{AN-SLDS}}$ and use all available predictions as follows:

$$\begin{aligned} \mathcal{L}_{\text{AN-full}}(\hat{\mathbf{y}}, \mathbf{z}) = & -\frac{1}{S} \sum_{j=1}^S [\mathbb{1}_{[z_j=1]} \lambda \log(\hat{y}_j) \\ & + \mathbb{1}_{[z_j \neq 1]} \log(1 - \hat{y}_j) + \log(1 - \hat{y}'_j)] \end{aligned} \quad (8.5)$$

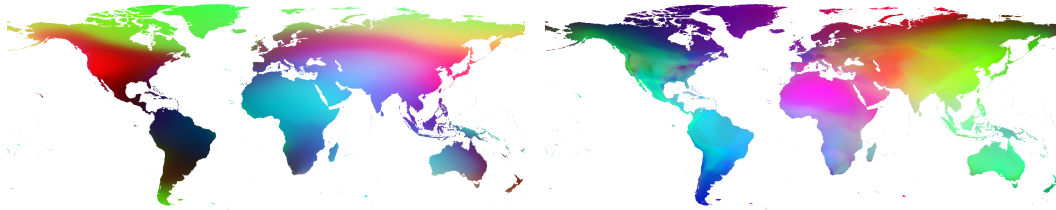


Figure 8.3: Visualization of the 256-dimensional features from learned location encoders f_θ projected to three dimensions using Independent Component Analysis (ICA). All models use the $\mathcal{L}_{\text{AN-full}}$ loss and take coordinates as input. (*Left*) This corresponds to a SINR model trained with a maximum of 10 examples per class. The features are smooth and do not appear to encode much high frequency spatial information. (*Right*) In contrast, the SINR model trained with a maximum of 1000 examples per class contains more high frequency information. The increase in training data appears to enable this model to better encode spatially varying environmental properties. Note, ICA is performed independently per-model, so similar colors do not indicate correspondence between the two images.

where $\hat{\mathbf{y}}' = h_\phi(f_\theta(\mathbf{r}))$ with $\mathbf{r} \sim \text{Unif}(\mathcal{X})$. The hyperparameter $\lambda > 0$ can be used to prevent the negative labels from dominating the loss. This is equivalent to the loss from [20], but without their user modeling terms. Their version (including user modeling terms) is \mathcal{L}_{GP} in Table 8.1 ("GP" = "Geo Prior").

Maximum entropy loss. [45] recently proposed a simple but effective and scalable technique for SPML image classification. Their approach encourages predictions for unobserved labels to maximize entropy instead of forcing them to zero like the "assume negative" approaches we have been discussing. We can apply this idea to $\mathcal{L}_{\text{AN-SSDL}}$, $\mathcal{L}_{\text{AN-SLDS}}$, and $\mathcal{L}_{\text{AN-full}}$ by replacing all terms of the form " $-\log(1-p)$ " with terms of the form " $H(p)$ ", where $H(p) = -(p \log(p) + (1-p) \log(1-p))$ is the Bernoulli entropy. We write these "maximum entropy" (ME) variants as $\mathcal{L}_{\text{ME-SSDL}}$, $\mathcal{L}_{\text{ME-SLDS}}$, and $\mathcal{L}_{\text{ME-full}}$. ([45] also includes a pseudo-labeling component, but we omit this because [45] shows that it provides only a small improvement.)

8.5 Experiments

In this section we investigate the performance of SINR models on four species and environmental prediction tasks.

Models

As described in Section 8.4, our SINR models consist of a location encoder f_θ and a multi-label classifier h_ϕ which produce a vector of predictions $\hat{\mathbf{y}} = h_\phi(f_\theta(\mathbf{x}))$ for a location \mathbf{x} . The location encoder f_θ is implemented as the fully connected

neural network shown in the supplementary material. We implement the multi-label classifier h_ϕ as a single fully connected layer with sigmoid activations. For fair comparisons, we follow a similar architecture to [20]. Full implementation details can be found in the supplementary material.

Besides SINR, we study two other model types. The first is logistic regression [26], in which the location encoder f_θ is replaced with the identity function and h_ϕ is unchanged. Logistic regression is commonly used for SDM in the ecology literature. It also has the virtue of being highly scalable since it can be trained using GPU-accelerated batch-based optimization. The second type of non-SINR model is the discretized grid model. These models do not use a location encoder at all, but instead make predictions based on binning the training data [3]. Full details for these models can be found in the supplementary material. These baselines allow us to quantify the importance of the deep location encoder in our SINR models.

Training Data

We train our models on presence-only species observation data obtained from the community science platform iNaturalist [17]. The training set consists of 35.5 million observations covering 47,375 species observed prior to 2022. Each species observation includes the geographical coordinate where the species was observed. We only included species in the training set if they had at least 50 observations. Some species are far more common than others, and thus the dataset is heavily imbalanced (see the supplementary material). Later we use this data in its entirety during training ("All"), with different maximum observations per class ("X / Class"), or with different subsets of classes. See the supplementary material for more details on the training dataset.

Evaluation Tasks and Metrics

We propose four tasks for evaluating large-scale species range estimation models. We give brief descriptions here, and provide further details in the supplementary material.

S&T: eBird Status and Trends. This task quantifies the agreement between our presence-only predictions and expert-derived range maps from the *eBird Status & Trends* dataset [13], covering 535 bird species with a focus on North America. The spatial extent of this task is visualized in the supplementary material. Performance is measured using mean average precision (MAP), i.e. computing the per-species average precision (AP) and averaging across species.

Table 8.1: Results for four geospatial tasks: **S&T** (eBird Status & Trends species mapping), **IUCN** (IUCN species mapping), **Geo Prior** (fine-grained image classification with a geographical prior), and **Geo Feature** (geographical feature regression). Tasks and metrics are defined in Section 8.5. We assess performance as a function of the loss function and the amount of training data (" $\# / \text{Class}$ "). Model inputs may be coordinates (" Coords. "), environmental features (" Env. ") or both (" $\text{Env.} + \text{Coords.}$ "). The logistic regression (" LR ") and "Best Discretized Grid" baselines do not have an entry for the **Geo Feature** task as they do not learn a location encoder. We also do not evaluate models tagged with " Env. " on the **Geo Feature** task because they are trained on closely related environmental features. Higher values are better for all tasks.

Loss	Model Type	$\# / \text{Class}$	S&T (MAP)	IUCN (MAP)	Geo Prior ($\Delta \text{Top-1}$)	Geo Feature (Mean R^2)
<i>Baselines:</i>						
N/A	Best Discretized Grid [3]	All	61.56	37.13	+4.1	-
$\mathcal{L}_{\text{AN-full}}$	LR [26] - Coords.	1000	26.41	0.93	-0.6	-
$\mathcal{L}_{\text{AN-full}}$	LR [26] - Env.	1000	32.91	1.23	-5.6	-
$\mathcal{L}_{\text{AN-full}}$	LR [26] - Env. + Coords.	1000	35.42	1.11	-3.9	-
$\mathcal{L}_{\text{ME-SSDL}}$ [45]	SINR - Coords.	1000	62.74	42.55	+1.6	0.726
$\mathcal{L}_{\text{ME-SLDS}}$ [45]	SINR - Coords.	1000	74.37	32.22	+2.1	0.734
$\mathcal{L}_{\text{ME-full}}$ [45]	SINR - Coords.	1000	73.61	58.60	+1.5	0.749
\mathcal{L}_{GP} [20]	SINR - Coords.	1000	73.14	59.51	+5.2	0.724
$\mathcal{L}_{\text{AN-SSDL}}$	SINR - Coords.	10	51.12	27.63	+3.4	0.631
$\mathcal{L}_{\text{AN-SSDL}}$	SINR - Coords.	100	63.98	47.42	+4.7	0.721
$\mathcal{L}_{\text{AN-SSDL}}$	SINR - Coords.	1000	66.99	53.47	+4.9	0.744
$\mathcal{L}_{\text{AN-SSDL}}$	SINR - Coords.	All	68.36	55.75	+4.8	0.739
$\mathcal{L}_{\text{AN-SLDS}}$	SINR - Coords.	10	63.73	27.14	+4.6	0.693
$\mathcal{L}_{\text{AN-SLDS}}$	SINR - Coords.	100	72.18	38.40	+6.1	0.731
$\mathcal{L}_{\text{AN-SLDS}}$	SINR - Coords.	1000	76.19	42.26	+6.2	0.739
$\mathcal{L}_{\text{AN-SLDS}}$	SINR - Coords.	All	75.78	41.11	+6.1	0.748
$\mathcal{L}_{\text{AN-full}}$	SINR - Coords.	10	65.36	49.02	+4.3	0.712
$\mathcal{L}_{\text{AN-full}}$	SINR - Coords.	100	72.82	62.00	+6.6	0.736
$\mathcal{L}_{\text{AN-full}}$	SINR - Coords.	1000	77.15	65.84	+6.1	0.755
$\mathcal{L}_{\text{AN-full}}$	SINR - Coords.	All	77.94	65.59	+5.0	0.759
$\mathcal{L}_{\text{AN-full}}$	SINR - Env.	10	60.10	41.68	+3.8	-
$\mathcal{L}_{\text{AN-full}}$	SINR - Env.	100	74.54	66.64	+6.7	-
$\mathcal{L}_{\text{AN-full}}$	SINR - Env.	1000	79.65	70.54	+6.4	-
$\mathcal{L}_{\text{AN-full}}$	SINR - Env.	All	80.54	69.25	+5.3	-
$\mathcal{L}_{\text{AN-full}}$	SINR - Env. + Coords.	10	67.12	62.99	+4.7	-
$\mathcal{L}_{\text{AN-full}}$	SINR - Env. + Coords.	100	76.88	74.49	+6.8	-
$\mathcal{L}_{\text{AN-full}}$	SINR - Env. + Coords.	1000	80.48	76.07	+6.5	-
$\mathcal{L}_{\text{AN-full}}$	SINR - Env. + Coords.	All	81.39	74.67	+5.5	-

IUCN: Expert Range Maps. This task compares our predictions against expert range maps from the International Union for Conservation of Nature (IUCN) Red List [18]. Unlike the bird-centric *S&T*, this task covers 2,418 species from different taxonomic groups, including birds, from all over the world. The spatial extent of this task is visualized in the supplementary material. Performance is measured using

MAP.

Geo Prior: Geographical Priors for Image Classification. This task measures the utility of our range maps as priors for fine-grained image classification [3, 20]. As illustrated in Figure 8.1, we combine the output of an image classifier with a range estimation model and measure the improvement in classification accuracy. The intuition is that an accurate range model can downweight the probability of a species if it is not typically found at the location where the image was taken. For this task we collect 282,974 images from iNaturalist, covering 39,444 species from our training set. Each image is accompanied by the latitude and longitude at which the image was taken. The performance metric for this task (" Δ Top-1") is the change in image classifier top-1 accuracy when using our range predictions as a geographical prior. Note that the geographical prior is applied to the classifier at test time — the image classifier is not trained with any geographical information. A positive value indicates that the prior improves classifier performance. Unlike *S&T* and *IUCN*, this is an *indirect* evaluation of range map quality since we assess how useful the range predictions are for a downstream task.

Geo Feature: Environmental Representation Learning. Instead of evaluating the species predictions, this transfer learning task evaluates the quality of the underlying geospatial representation learned by a SINR. The task is to predict nine different geospatial characteristics of the environment, e.g. above-ground carbon, elevation, etc. First, we use the location encoder f_θ to extract features for a grid of evenly spaced locations across the contiguous United States. After splitting the locations into train and test data, we use ridge regression to predict the geospatial characteristics from the extracted features. Performance is evaluated using the coefficient of determination R^2 on the test set, averaged across the nine geospatial characteristics.

Results

Which loss is best? No loss is best in every setting we consider. However, some losses do tend to perform better than others. In Table 8.1 we observe that, when we control for input type and the amount of training data, $\mathcal{L}_{\text{AN-full}}$ outperforms $\mathcal{L}_{\text{AN-SSDL}}$ and $\mathcal{L}_{\text{AN-SLDS}}$ most of the time. $\mathcal{L}_{\text{AN-full}}$ has a decisive advantage on the *S&T* and *IUCN* tasks and a consistent but small advantage on the *Geo Feature* task. Both $\mathcal{L}_{\text{AN-full}}$ and $\mathcal{L}_{\text{AN-SLDS}}$ perform well on the *Geo Prior* task, significantly outperforming $\mathcal{L}_{\text{AN-SSDL}}$. We note that $\mathcal{L}_{\text{AN-full}}$ is a simplified version of \mathcal{L}_{GP} from [20], but $\mathcal{L}_{\text{AN-full}}$ outperforms \mathcal{L}_{GP} on every task.

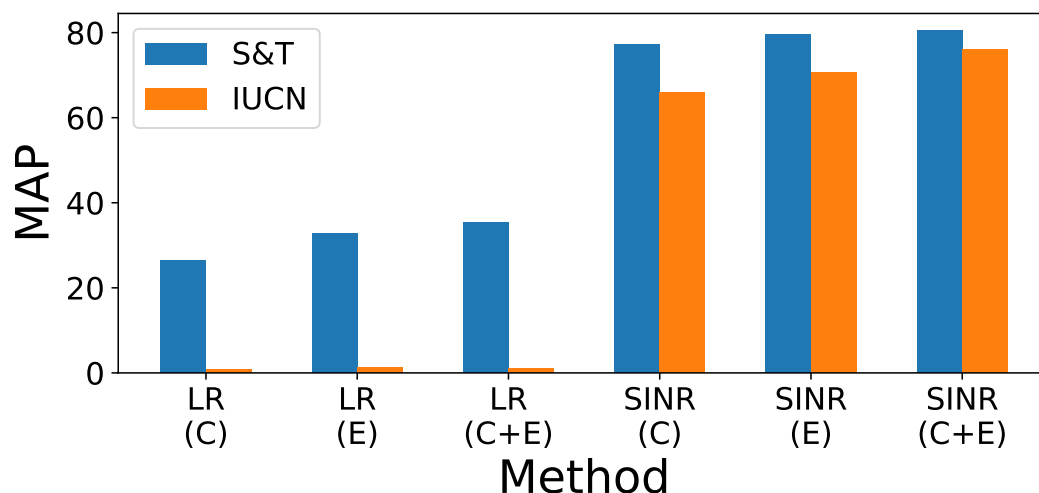


Figure 8.4: Results for the *S&T* and *IUCN* tasks. All models are trained with 1000 examples per class using the $\mathcal{L}_{\text{AN-full}}$ loss. We compare logistic regression ("LR") models against SINR models, using either coordinates (C), environmental covariates (E), or both (C+E) as inputs. These values can also be found in Table 8.1.

Pseudo-negatives that follow the data distribution are usually better. $\mathcal{L}_{\text{AN-SSDL}}$ and $\mathcal{L}_{\text{AN-SLDS}}$ differ only in the fact that $\mathcal{L}_{\text{AN-SSDL}}$ samples pseudo-negatives from random locations while $\mathcal{L}_{\text{AN-SLDS}}$ samples pseudo-negatives from data locations (see Figure 8.2). In Table 8.1 we see that $\mathcal{L}_{\text{AN-SLDS}}$ outperforms $\mathcal{L}_{\text{AN-SSDL}}$ for all tasks except *IUCN*. This could be due to the fact that some *IUCN* species have ranges far from areas that are well-sampled by iNaturalist. In the Black Oystercatcher range shown in the supplementary material we see that $\mathcal{L}_{\text{AN-SSDL}}$ can behave poorly in areas with little training data. This highlights the importance of using diverse tasks to study range estimation methods.

Implicit neural representations significantly improve performance. We can assess the impact of the deep location encoder by comparing SINR and LR in models Table 8.1. For instance, if we use the $\mathcal{L}_{\text{AN-full}}$ loss with 1000 examples per class and coordinates as input, SINR outperforms LR by over 50 MAP on the *S&T* task. Both methods use the same inputs and training loss — the only difference is that SINR uses a deep location encoder while LR does not. Figure 8.4 shows that same pattern holds whether we use coordinates, environmental features, or both as inputs. For each input type, a deep location encoder provides significant benefits.

Environmental features are not necessary for good performance. In Figure 8.4 we show the *S&T* and *IUCN* performance of different models trained with coordinates only, environmental features only, or both. We see that SINR models trained

with coordinates perform nearly as well as SINR models trained with environmental features. For the SINR models in Figure 8.4, coordinates are 97% as good as environmental features for the *S&T* task, 93% as good for the *IUCN* task, and 95% as good for the *Geo Prior* task. This suggests that SINRs can successfully use sparse presence-only data to learn about the environment, so that using environmental features as input provides only a marginal benefit.

Coordinates and environmental features are complementary. Figure 8.4 shows that it is better to use the concatenation of coordinates and environmental features than it is to use either coordinates or environmental features alone. This is true for LR and SINR. This indicates that the coordinates and environmental features are carrying some complementary information. However, as we discuss in the supplementary material, environmental features introduce an additional layer of complexity compared to models that use only coordinates.

Joint learning across categories is beneficial, but more data is better. In Figure 8.5 we study the effect of the amount of training data on performance for the *S&T* task. We first note that, unsurprisingly, increasing the number of training examples per species reliably and significantly improves performance. One possible mechanism for this is suggested by Figure 8.3, which shows a more spatially detailed representation emerging with more training data. More interestingly, Figure 8.5 also shows that adding training data for additional species (which are not evaluated at test time) improves performance as well. That is, the model can better predict the distributions of the *S&T* birds by also learning the distributions of other birds, plants, insects, etc. Intuitively, it seems reasonable that training on more species could lead to a richer and more useful geospatial representation. However, the direct benefit of additional training data for the species of interest is far larger. If we were given a fixed budget of training examples to allocate among species as we wished, we should prefer to have a larger number of training examples per species (instead of fewer training examples per species, but spread across a greater number of species).

Low-shot performance is surprisingly good. In Table 8.1 we see that a SINR trained with $\mathcal{L}_{AN-full}$ and only 10 examples per category (i.e. $\sim 1\%$ of the training data) beats the "Best Discretized Grid" baseline (which uses all of the training data) on every task. SINRs seem to be capable of capturing general spatial patterns using relatively little data. While this is encouraging, we expect that more data is necessary to capture fine detail as suggested by Figure 8.3 and Figure 8.7.

How are our tasks related? In this work we study four spatial prediction tasks. This

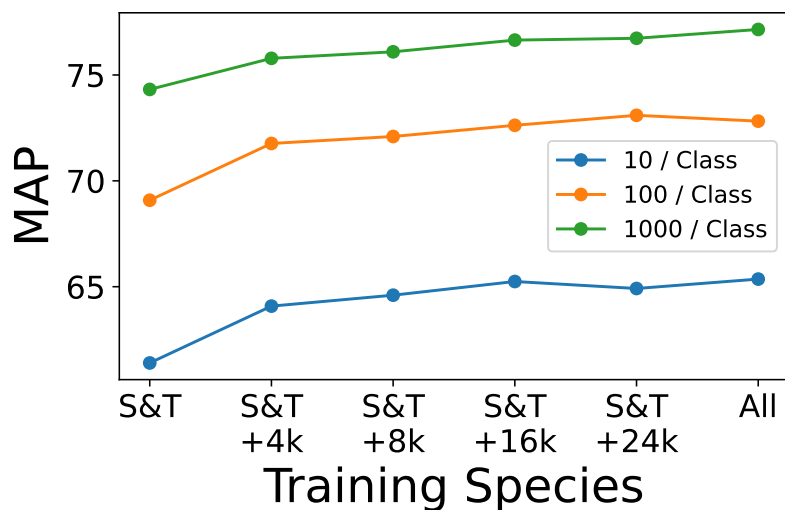


Figure 8.5: *S&T* task performance with $\mathcal{L}_{\text{AN-full}}$ as a function of the number of training examples per class (i.e. species) and number of classes. The horizontal axis gives the set of species used for training. "S&T" indicates that we only train on the 535 species in the S&T task. For "S&T + X" we add in X species chosen uniformly at random. For "All" we train on all 47k species. Note that the "10 / Class" point for "S&T" is trained with a higher learning rate than usual ($5e-3$ instead of $5e-4$) due to the small number of training examples per epoch. The values for "All" are also present in Table 8.1. All models use coordinates as input.

tasks differ in their spatial domains, evaluation metrics, and categories of interest, but it is reasonable to wonder to what extent they may be related. In Figure 8.6 we show the pairwise correlations between scores on our tasks. Some tasks are highly correlated (e.g. *S&T* and *Geo Features*, 0.92) while others are not (e.g. *IUCN* and *Geo Prior*, 0.39).

Imbalance hurts performance, but not too much. In Table 8.1 we notice that a SINR trained with all of the training data often performs worse than a SINR trained on up to 1000 examples per class. This pattern is clearest for the *IUCN* and *Geo Prior* tasks. Capping the number of training examples per class reduces the amount of training data, but it also reduces class imbalance in the training set (some categories have as many as $\sim 10^5$ training examples). It seems that the benefit of reducing class imbalance outweighs the benefit of additional training data in these cases. However, it is important to keep in mind that the performance drops we are discussing are small. For instance, for a SINR trained with $\mathcal{L}_{\text{AN-full}}$ and coordinates as input, switching from 1000 training examples to all of the training data changes performance by -0.79 MAP for the *S&T* task, -0.25 MAP for the *IUCN* task, -1.1

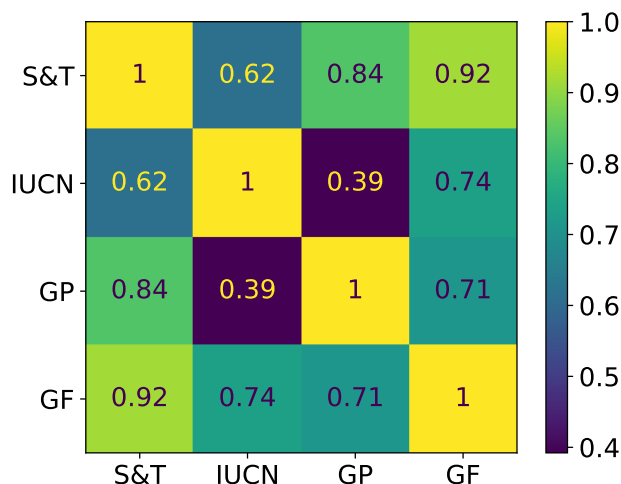


Figure 8.6: Performance correlations across our four tasks: *S&T*, *IUCN*, *Geo Prior* (GP), and *Geo Feature* (GF). Values are Pearson product-moment correlation coefficients. The correlations are computed across 12 SINR models: $\mathcal{L}_{AN-SSDL}$, $\mathcal{L}_{AN-SLDS}$, and $\mathcal{L}_{AN-full}$ for 10, 100, 1000, and All training examples per class. All models use coordinates as input.

Δ Top-1 for the *Geo Prior* task, and +0.004 for the *Geo Feature* task. Given the extreme imbalance in the training set and the fact that we do not explicitly handle class imbalance during training, it may be surprising that the performance drops are not larger.

Loss function rankings may not generalize across domains. The presence-only SDM problem in this work and the single positive image classification problem in [9] are both SPML problems. Despite this formal equivalence, it does not seem that the best methods for SPML image classification are also the best methods for presence-only SDM. [45] show that their "maximum entropy" loss performs much better than the "assume negative" loss across a number of image classification datasets. However, all of the "maximum entropy" losses in Table 8.1 ($\mathcal{L}_{ME-SSDL}$, $\mathcal{L}_{ME-SLDS}$, $\mathcal{L}_{ME-full}$) underperform their "assume negative" counterparts ($\mathcal{L}_{AN-SSDL}$, $\mathcal{L}_{AN-SLDS}$, $\mathcal{L}_{AN-full}$). Thus, the benchmarks in this paper are complementary to those in [9] and may be useful in developing a more holistic understanding of SPML learning.

Limitations

It is important to be aware of the limitations associated with our analysis. As noted, the training set is heavily imbalanced, both in terms of the species themselves and where the data was collected. In practice, some of the most biodiverse regions are

underrepresented. This is partially because some species are more common and thus more likely to be observed than others by iNaturalist users. We do not explicitly deal with species imbalance in the training data, other than by showing that the ranking of methods does not significantly vary even when the training data for each species is capped to the same upper limit (see Table 8.1).

Reliably evaluating the performance of SDMs for many species and locations is a long standing challenge. To address this issue, we present a suite of complementary benchmarks that attempt to evaluate different facets of this spatial prediction problem. However, obtaining ground truth range data for thousands of species remains very difficult. While we believe our benchmarks to be a significant step forward, they are likely to have blind spots, e.g. they are limited to well-described species and can contain inaccuracies.

Finally, care should be taken before making conservation decisions based on the outputs of models such as the ones presented here. Our goal in this work is to demonstrate the promise of large-scale representation learning for species distribution modeling. Our models have not been calibrated or validated beyond the experiments illustrated above.

8.6 Conclusion

We explored the problem of species range mapping through the lens of learning spatial implicit neural representations (SINRs). In doing so, we connected recent work on implicit coordinate networks and learning multi-label classifiers from limited supervision. We hope our contributions encourage more machine learning researchers to work on this important problem. While the initial results are encouraging, there are many avenues for future work. For example, our models make no use of time [20], do not account for spatial bias [6], and have no inductive biases for encoding spatially varying signals [30].

8.7 Acknowledgements

We thank the iNaturalist and eBird communities for their data collection efforts, as well as Matt Stimas-Mackey and Sam Heinrich for help with data curation. This project was funded by the Climate Change AI Innovation Grants program, hosted by Climate Change AI with the support of the Quadrature Climate Foundation, Schmidt Futures, and the Canada Hub of Future Earth. This work was also supported by the Caltech Resnick Sustainability Institute and an NSF Graduate Research Fellowship (grant number DGE1745301).

References

- [1] Julio Arroyo, Pietro Perona, and Elijah Cole. “Understanding Label Bias in Single Positive Multi-Label Learning”. In: 2023.
- [2] Sara Beery*, Elijah Cole*, Joseph Parker, Pietro Perona, and Kevin Win-
ner. “Species distribution modeling for machine learning practitioners: A
review”. In: *ACM SIGCAS conference on computing and sustainable soci-
eties*. 2021, pp. 329–348. DOI: 10.48550/arXiv.2107.10400.
- [3] Thomas Berg et al. “Birdsnap: Large-scale fine-grained visual categorization
of birds”. In: *CVPR*. 2014.
- [4] Christophe Botella et al. “A Deep Learning Approach to Species Distribu-
tion Modelling”. In: *Multimedia Tools and Applications for Environmental
& Biodiversity Informatics*. 2018.
- [5] Christophe Botella et al. “Jointly estimating spatial sampling effort and
habitat suitability for multiple species from opportunistic presence-only
data”. In: *Methods in Ecology and Evolution* (2021).
- [6] Di Chen and Carla P Gomes. “Bias reduction via end-to-end shift learning:
Application to citizen science”. In: *AAAI*. 2019.
- [7] Di Chen et al. “Deep multi-species embedding”. In: *IJCAI*. 2017.
- [8] Grace Chu et al. “Geo-aware networks for fine-grained recognition”. In:
ICCV Workshops. 2019.
- [9] Elijah Cole, Oisín Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris,
and Nebojsa Jojic. “Multi-label learning from single positive labels”. In:
*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
Recognition*. 2021, pp. 933–942. DOI: 10.48550/arXiv.2106.09708.
- [10] D Richard Cutler et al. “Random forests for classification in ecology”. In:
Ecology 88.11 (2007), pp. 2783–2792.
- [11] Jane Elith and John R Leathwick. “Species distribution models: ecological
explanation and prediction across space and time”. In: *Annual review of
ecology, evolution, and systematics* 40 (2009), pp. 677–697.
- [12] Jane Elith et al. “Novel methods improve prediction of species’ distribu-
tions from occurrence data”. In: *Ecography* (2006).
- [13] Daniel Fink et al. “eBird Status and Trends, Data Version: 2020; Released:
2021”. In: *Cornell Lab of Ornithology, Ithaca, New York* 10 (2020).
- [14] Daniel Fink et al. “Spatiotemporal exploratory models for broad-scale survey
data”. In: *Ecological Applications* (2010).
- [15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT
Press, 2016.

- [16] Trevor Hastie and Will Fithian. “Inference from presence-only data; the ongoing controversy”. In: *Ecography* 36.8 (2013), pp. 864–867.
- [17] *iNaturalist*. www.inaturalist.org, accessed Mar 7 2022.
- [18] IUCN 2022. *The IUCN Red List of Threatened Species*. 2022-2. <https://www.iucnredlist.org>, accessed 9 May 2023.
- [19] Alison Johnston et al. “Estimating species distributions from spatially biased citizen science data”. In: *Ecological Modelling* (2020).
- [20] Oisín Mac Aodha, Elijah Cole, and Pietro Perona. “Presence-only geographical priors for fine-grained image classification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9596–9606. DOI: 10.48550/arXiv.1906.05272.
- [21] Darryl I MacKenzie et al. “Estimating site occupancy rates when detection probabilities are less than one”. In: *Ecology* (2002).
- [22] Gengchen Mai et al. “Multi-Scale Representation Learning for Spatial Feature Distributions using Grid Cells”. In: *ICLR*. 2020.
- [23] Gengchen Mai et al. “Sphere2Vec: Multi-Scale Representation Learning over a Spherical Surface for Geospatial Predictions”. In: *arXiv:2201.10489* (2022).
- [24] Ben Mildenhall et al. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *ECCV*. 2020.
- [25] Anna Norberg et al. “A comprehensive evaluation of predictive performance of 33 species distribution models at species and community levels”. In: *Ecological Monographs* 89.3 (2019), e01370.
- [26] Jennie Pearce and Simon Ferrier. “An evaluation of alternative algorithms for fitting species distribution models using logistic regression”. In: *Ecological modelling* 128.2-3 (2000), pp. 127–147.
- [27] Steven J Phillips, Miroslav Dudík, and Robert E Schapire. “A maximum entropy approach to species distribution modeling”. In: *ICML*. 2004.
- [28] Steven J Phillips et al. “Sample selection bias and presence-only distribution models: implications for background and pseudo-absence data”. In: *Ecological Applications* (2009).
- [29] Joanne M Potts and Jane Elith. “Comparing species abundance models”. In: *Ecological modelling* (2006).
- [30] Sameera Ramasinghe and Simon Lucey. “Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps”. In: *ECCV*. 2022.
- [31] Kenneth V Rosenberg et al. “Decline of the North American avifauna”. In: *Science* (2019).

- [32] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. “Scene representation networks: Continuous 3d-structure-aware neural scene representations”. In: *NeurIPS* (2019).
- [33] Marta Skreta, Alexandra Luccioni, and David Rolnick. “Spatiotemporal Features Improve Fine-Grained Butterfly Image Classification”. In: *Tackling Climate Change with Machine Learning Workshop at NeurIPS*. 2020.
- [34] Matthew Tancik et al. “Fourier features let networks learn high frequency functions in low dimensional domains”. In: *NeurIPS*. 2020.
- [35] Kevin Tang et al. “Improving image classification with location context”. In: *ICCV*. 2015.
- [36] Luming Tang et al. “Multi-entity dependence learning with rich context via conditional variational auto-encoder”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.
- [37] Mélisande Teng et al. “Bird Distribution Modelling using Remote Sensing and Citizen Science data”. In: *Tackling Climate Change with Machine Learning Workshop, ICLR* (2023).
- [38] J Christopher D Terry, Helen E Roy, and Tom A August. “Thinking like a naturalist: Enhancing computer vision of citizen science images by harnessing contextual data”. In: *Methods in Ecology and Evolution* (2020).
- [39] Roozbeh Valavi et al. “Modelling species presence-only data with random forests”. In: *Ecography* (2021).
- [40] Roozbeh Valavi et al. “Predictive performance of presence-only species distribution models: a benchmark study with reproducible code”. In: *Ecological Monographs* (2022).
- [41] Thomas Verelst et al. “Spatial Consistency Loss for Training Multi-Label Classifiers from Single-Label Annotations”. In: *WACV*. 2023.
- [42] Robert Watson et al. *Summary for policymakers of the global assessment report on biodiversity and ecosystem services of the Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services*. IPBES Secretariat, 2019.
- [43] Lingfeng Yang et al. “Dynamic MLP for Fine-Grained Image Classification by Leveraging Geographical and Temporal Information”. In: *CVPR*. 2022.
- [44] Jianqiao Zheng et al. “Trading Positional Complexity vs. Deepness in Coordinate Networks”. In: *ECCV*. 2022.
- [45] Donghao Zhou et al. “Acknowledging the Unknown for Multi-label Learning with Single Positive Labels”. In: *ECCV*. 2022.

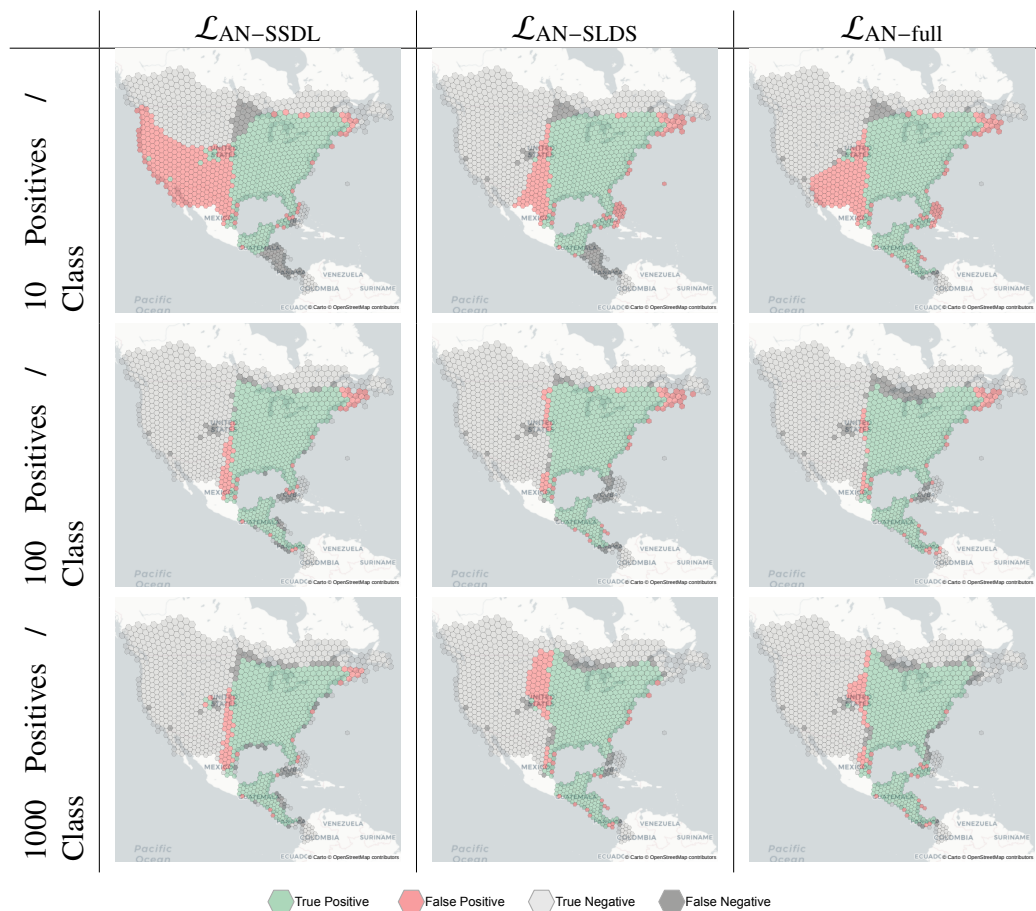


Figure 8.7: Visualization of SINR predictions for Wood Thrush (<https://ebird.org/species/woothr>) when varying the amount of training data (rows) for different loss functions (columns). Model predictions are generated at the centroid of the rendered hexagons for a coarse H3 grid (resolution three), signifying locations where we can evaluate the model outputs for the *S&T* task. We convert the predictions to binary values using the threshold that maximizes the F1 score on the *S&T* data. This is done for each configuration independently. In practice this threshold would be chosen by a practitioner to meet particular project requirements. A model that matches the *S&T* task exactly would show only green and light grey hexagons. All models improve their range maps when given access to more data, as expected. $\mathcal{L}_{AN-SSDL}$ overestimates the western range extent and misses the southern extent with few examples, but refines these extents with additional data. $\mathcal{L}_{AN-full}$ starts off with most of the range covered (few "False Negative" hexagons) and proceeds to tighten the boundaries with more data. The range predicted by $\mathcal{L}_{AN-SLDS}$ is somewhere in between. All models use coordinates as input.

Part IV

Conclusion

Chapter 9

CONCLUSION

This chapter summarizes some of the lessons of this thesis and discusses opportunities for future work.

9.1 Lessons Learned

Build Benchmarks

Benchmarks (i.e. the datasets and metrics we use to measure progress in machine learning) are often considered to be secondary contributions, but ultimately they steer the trajectory of the machine learning research community. A dataset is like a telescope — you can use it to see new things, but what you can see depends on how it was built. Different datasets reveal different behaviors and properties of our algorithms. For instance, the rapid advances in self-supervised learning over the last few years were based mostly on ImageNet. Chapters 2 and 3 used new benchmarks to show that these algorithms have a major limitation: they only learn coarse visual similarity. This observation reveals fine-grained self-supervised as a new frontier for fundamental research. Similar examples can be found in Chapter 4, 7, and 8. Well-designed benchmarks do not just provide another generic problem to solve, they stimulate the development of new ideas and algorithms in machine learning.

Attack Difficult Applied Problems

The most reliable way to discover opportunities for machine learning research is to try to apply existing techniques to an important real-world problem. Unless you are very lucky, you will find that even state-of-the-art techniques do not work very well. While machine learning has made impressive progress over the last decade, this progress is largely limited to the *traditional ML paradigm*, where there is abundant, accurately labeled training data which is approximately independent and identically distributed. These ideal conditions are not often met in practice, especially for scientific problems where the high cost of expert time means that accurate labels are scarce. Noisy labels and contextual data may be more abundant, but they are seldom compatible with traditional ML algorithms. Far from being a distraction from "pure" machine learning research, applications drive progress and discovery in machine learning. Every chapter in this thesis relies on this fact.

Use Domain Knowledge Creatively

Many important application domains are equipped with rich contextual data. However, this data is often ignored because it does not fit neatly into traditional machine learning approaches. However, it is possible for standard machine learning algorithms to take advantage of contextual data (i.e. metadata or prior knowledge) without the need to re-design foundational algorithm components (e.g. losses, model architectures) for each domain. For instance, Chapter 7 introduced geographical priors that are compatible with any image classification algorithm. Similarly, Chapter 4 showed that one can use label hierarchy information to improve many different weakly supervised object localization algorithms. It is worthwhile to look for opportunities to leverage domain knowledge without throwing out the useful building blocks the machine learning community has carefully refined.

9.2 Future Work

Beyond Static Benchmarks

The machine learning community works almost exclusively with *static* benchmark datasets. This thesis is no exception. However, static curated datasets are merely snapshots of the richness and variety of the real world. We need to develop techniques for learning from *datastreams*, i.e. essentially unlimited pipelines of raw data accompanied by heterogeneous metadata. Datastreams have already emerged in the context of earth observation, VR/AR platforms, high-fidelity simulation environments, and community science platforms like iNaturalist, all of which generate enormous amounts of data that varies widely in quality, format, and metadata completeness, with extremely limited access to expert human supervision. It may seem that the right solution is simply to scale up our models to match the quantity of data, but large models may be too slow to process all of the incoming data. This would make it impossible to solve problems like identifying important or unusual data in the datastream. Over the next decade, *datastream learning* will further challenge our traditional ML paradigms and require us to rethink how we organize the learning process at scale.

Spatial Learning

The projects in Part III of this thesis begin to expand the Visipedia project to include spatial data. Even in the context of spatial learning for species distribution modeling, there is a considerable amount of work still to do. One exciting direction would be to use the models developed in Chapter 8 as the basis for active learning systems

that ask community scientists to investigate specific locations. But what kinds of architectures and training procedures should we use to learn the most ecologically useful geospatial representations? And how should we design the interaction between the users and the algorithm to account for factors like engagement, safety, and scientific utility? This is an emerging area, and there is a lot of basic machine learning research to be done to understand how the elements of Visipedia — data, users, and algorithms — should be implemented for spatial data.

Next Steps for Visipedia

Since its inception, the Visipedia project has largely focused on problems in ecology. These ecology problems have provided the context in which Visipedia researchers have refined crowdsourcing, human-in-the-loop learning, representation learning, and other key technologies. I believe the time is right for Visipedia to expand to other domains. I am particularly excited about opportunities in other areas of biology e.g. cell and molecular biology, medicine, neuroscience, etc. Deep learning has made inroads into all of these areas, but — as we have observed in the context of ecology — I expect that the most impactful work will emerge not from pure automation, but from a Visipedia-style engagement between domain experts and machine learning researchers.