

# UC San Diego

## UC San Diego Electronic Theses and Dissertations

### Title

Scaling up Recognition in Expert Domains with Crowd-source Annotations

### Permalink

<https://escholarship.org/uc/item/5xr686tw>

### Author

Wang, Pei

### Publication Date

2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

**Scaling up Recognition in Expert Domains with Crowd-source Annotations**

A dissertation submitted in partial satisfaction of the  
requirements for the degree  
Doctor of Philosophy

in

Electrical Engineering (Machine Learning and Data Science)

by

Pei Wang

Committee in charge:

Professor Nuno Vasconcelos, Chair  
Professor David Kriegman  
Professor Truong Nguyen  
Professor Bhaskar D. Rao  
Professor Xiaolong Wang

2022

Copyright  
Pei Wang, 2022  
All rights reserved.

The dissertation of Pei Wang is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

To my family.

EPIGRAPH

*Nothing is impossible  
to a willing heart.*  
—John Heywood

## TABLE OF CONTENTS

Dissertation Approval Page . . . . .	iii
Dedication . . . . .	iv
Epigraph . . . . .	v
Table of Contents . . . . .	vi
List of Figures . . . . .	ix
List of Tables . . . . .	xi
Acknowledgements . . . . .	xii
Vita . . . . .	xv
Abstract of the Dissertation . . . . .	xvii
Chapter 1	
Introduction . . . . .	1
1.1 Learning in Expert Domains . . . . .	2
1.2 Existing Methods . . . . .	3
1.3 Contributions of the Thesis . . . . .	6
1.3.1 Gradient-based Algorithms for Machine Teaching . . . . .	8
1.3.2 A Generalized Explanation Framework for Visualization of Deep Learning Model Predictions . . . . .	8
1.3.3 A Machine Teaching Framework for Scalable Recognition	9
1.3.4 Towards Professional Level Crowd Annotation of Expert Domain Data . . . . .	10
1.4 Organization of the Thesis . . . . .	10
Chapter 2	
Gradient-based Algorithms for Machine Teaching . . . . .	12
2.1 Introduction . . . . .	13
2.2 Related Work . . . . .	16
2.3 Gradient-Based Machine Teaching . . . . .	18
2.3.1 Machine Teaching . . . . .	18
2.3.2 The Optimal Student Assumption . . . . .	19
2.3.3 Functional Optimization . . . . .	22
2.3.4 The Optimal Teacher . . . . .	23
2.3.5 Multi-class Extension . . . . .	30
2.3.6 Connections to Boosting . . . . .	32
2.4 Experiments . . . . .	32
2.4.1 Evaluation with Simulated Learners . . . . .	33

	2.4.2	Evaluation with Real Learners . . . . .	34
	2.5	Conclusion . . . . .	36
Chapter 3		A Generalized Explanation Framework for Visualization of Deep Learning Model Predictions . . . . .	37
	3.1	Introduction . . . . .	38
	3.2	Related Work . . . . .	41
	3.3	A Unified View of Explainable AI . . . . .	45
		3.3.1 Attributive Explanations . . . . .	46
		3.3.2 Deliberative Explanations . . . . .	46
		3.3.3 Counterfactual Explanations . . . . .	48
	3.4	Implementation of GALORE . . . . .	49
		3.4.1 Explanation Framework . . . . .	49
		3.4.2 Attributive Explanations . . . . .	50
		3.4.3 Self-aware Attributive Explanations . . . . .	51
		3.4.4 Deliberative Explanations . . . . .	51
		3.4.5 Counterfactual Explanations . . . . .	52
	3.5	Implementation . . . . .	54
		3.5.1 Attribution Maps . . . . .	56
		3.5.2 Confidence Scores . . . . .	57
		3.5.3 Network Implementation . . . . .	59
	3.6	Evaluation . . . . .	59
		3.6.1 User Experiments . . . . .	59
		3.6.2 Proxy Tasks . . . . .	60
	3.7	Experiments . . . . .	63
		3.7.1 Experimental Setup . . . . .	63
		3.7.2 Ablation Study . . . . .	64
		3.7.3 Sanity Checks . . . . .	71
		3.7.4 Visualizations . . . . .	71
		3.7.5 Comparison to State of the Art . . . . .	73
	3.8	Human Studies . . . . .	76
		3.8.1 Insecurity Evaluation . . . . .	76
		3.8.2 Application to Machine Teaching . . . . .	77
	3.9	Conclusion . . . . .	79
Chapter 4		A Machine Teaching Framework for Scalable Recognition . . . . .	81
	4.1	Introduction . . . . .	82
	4.2	Related Work . . . . .	85
	4.3	The MEMORABLE Framework . . . . .	88
		4.3.1 Machine Teaching . . . . .	88
		4.3.2 How Important is Annotator Accuracy? . . . . .	90
		4.3.3 The Role of Explanations . . . . .	92
	4.4	Counterfactual MaxGrad (CMaxGrad) . . . . .	93



4.5	Evaluation of Student Teaching . . . . .	97
4.5.1	On the Simulated Learners . . . . .	99
4.5.2	On the Real Learners . . . . .	100
4.6	Evaluation of Scalable Recognition . . . . .	100
4.6.1	Comparison with the State of the Art . . . . .	102
4.6.2	Enhancements . . . . .	103
4.7	Conclusion . . . . .	104
Chapter 5	Towards Professional Level Crowd Annotation of Expert Domain Data	106
5.1	Introduction . . . . .	107
5.2	Related Work . . . . .	110
5.3	Challenges of POSER annotation . . . . .	113
5.4	SSL with Human Filtering . . . . .	114
5.4.1	Motivation . . . . .	114
5.4.2	Support Set Generation . . . . .	116
5.4.3	Explanation Generation . . . . .	117
5.4.4	Implementation . . . . .	118
5.4.5	Comparison to Other Methods . . . . .	119
5.5	Experiment . . . . .	121
5.5.1	Annotation Performance . . . . .	122
5.5.2	Ablation Study . . . . .	124
5.5.3	Comparisons on Crowd-source Platforms . . . . .	128
5.5.4	Comparisons by Human Simulation . . . . .	131
5.6	Conclusion . . . . .	133
Chapter 6	Discussion and Conclusion . . . . .	134
Bibliography	. . . . .	138

## LIST OF FIGURES

Figure 1.1: Annotations with different formats. . . . .	3
Figure 1.2: Examples of five gull species . . . . .	4
Figure 1.3: The comparison among different methods for annotation limited problem given a small labeled set and a large unlabeled set. . . . .	5
Figure 1.4: Classification accuracy comparison of different methods on expert domain data. ‘Baseline’: the supervised training baseline only on the labeled set. ‘SOTA’: the state of the art method. . . . .	6
Figure 2.1: Iterative machine teaching process. . . . .	14
Figure 2.2: Novel set selection by MaxGrad. . . . .	25
Figure 2.3: Example images from our two datasets . . . . .	33
Figure 2.4: Test set accuracy of simulated students as a function of teaching iterations (teaching example number). . . . .	34
Figure 3.1: An ideal explainable deep learning system. . . . .	39
Figure 3.2: Left: Illustration of the deliberations made by a human to categorize an ambiguous image. Insecurities are ambiguous regions. Right: Deliberative explanations expose this deliberative process. . . . .	42
Figure 3.3: The derivation of a counterfactual explanation. . . . .	43
Figure 3.4: GALORE explanation architecture. . . . .	58
Figure 3.5: Effect of confidence scores on precision-recall curves and IoU of different GALORE explanations. . . . .	65
Figure 3.6: Impact of attribution function on GALORE explanation performance. Top: precision-recall on CUB200. Bottom: IoU on ADE20K. . . . .	66
Figure 3.7: Impact of network architecture on GALORE explanation performance. Top: precision-recall on CUB200. Bottom: IoU on ADE20K. . . . .	67
Figure 3.8: Robustness of GALORE to image shifts on CUB200. . . . .	68
Figure 3.9: Precision-recall of GALORE explanations obtained with pre-trained and random weights on CUB200. . . . .	68
Figure 3.10: Deliberative explanations produced by GALORE for two images from CUB. . . . .	69
Figure 3.11: Deliberative explanations produced by GALORE for four images from ADE20K. . . . .	70
Figure 3.12: Counterfactual explanations (true and counter classes shown below each example, ground truth class-specific part attributes in parenthesis). . .	73
Figure 3.13: Counterfactual explanations by GALORE on ADE20K. . . . .	75
Figure 3.14: PIoU of proposed counterfactual explanations as a function of the segmentation threshold on CUB200. Left: VGG16, right: ResNet-50. .	76
Figure 3.15: MTurk interface for human evaluation of deliberative explanations. . .	77
Figure 3.16: Visualization of machine teaching experiment. . . . .	78

Figure 4.1:	The proposed MEMORABLE framework for large-scale recognition in fine-grained domains. . . . .	85
Figure 4.2:	Confusion matrices for human annotators trained by different machine teaching algorithms on Butterflies dataset. . . . .	89
Figure 4.3:	Labeling and classification accuracies of simulated turkers. . . . .	90
Figure 4.4:	Interface. When the teaching image is “Viceroy” but the worker selected “Monarch”, the shown feedback will be given. . . . .	98
Figure 4.5:	Test set accuracy of simulated students as a function of teaching iterations (teaching example number). . . . .	99
Figure 4.6:	Sample images of Gull dataset. . . . .	101
Figure 4.7:	Comparison of counterfactual explanations generated by different models. Two examples are shown. Top: true class is “Viceroy” and counter class “Monarch”; bottom: true class is “Queen” and counter class “Red Admiral”. . . . .	103
Figure 5.1:	Different approaches to the labeling of a query image. . . . .	108
Figure 5.2:	Interface (right black box) used in SSL-HF . . . . .	112
Figure 5.3:	Deliberative explanation for a query image of a ‘Mangrove Cuckoo’ and simplified explanation used in SSL-HF (green box). Examples from the ambiguous classes are shown on the bottom for illustration only. . . . .	117
Figure 5.4:	Confusion matrix for human filter results. . . . .	122
Figure 5.5:	Human annotation accuracy vs. pseudo label accuracy. . . . .	123
Figure 5.6:	Results of each iteration under different metrics. . . . .	124
Figure 5.7:	Ablation study for thresholds. . . . .	125
Figure 5.8:	Result comparison of different support set sample choices. . . . .	127
Figure 5.9:	Result comparison of different support set sizes. . . . .	128
Figure 5.10:	Accuracy comparison under different thresholding strategies . . . . .	130
Figure 5.11:	The trade-off comparison of supervised/SSL/SSL-HF. . . . .	131

## LIST OF TABLES

Table 2.1:	Test set accuracies for MTurk learners. Methods with superscript “*” represent our implementations. Values are presented by mean(std). . . .	35
Table 3.1:	Implementation of different explanation strategies under the GALORE framework. . . . .	55
Table 3.2:	Comparison to the state of the art in counterfactual explanations. . . .	74
Table 4.1:	Test set labeling accuracy, mean (std), of MTurkers. Methods with superscript “*” represent our implementations. Values are presented by mean(std). . . . .	100
Table 4.2:	Test accuracy comparison with mean (std). The lower group shows our results whereas the upper other literature. . . . .	102
Table 5.1:	Ablation study for support sets. . . . .	124
Table 5.2:	Ablation study for support sets. . . . .	128
Table 5.3:	Classification accuracy (mean(std)) comparison with the state of the art SSL. * denotes the results are generated by human simulation. Missing std because of no std reported in the original literature. . . . .	129
Table 5.4:	Labeling and classification accuracy (mean(std)) comparison. . . . .	130
Table 5.5:	Classification accuracy of simulated experiments with different $R$ , and Two-sample two-tailed T test. Mean(std)/t-score, using p-value of 0.05.	131

## ACKNOWLEDGEMENTS

I can hardly believe that the long Ph.D. journey is approaching to its end. The five-year experience of studying at UCSD has left an indelible impression on my heart. I have nothing but deep gratitude to all those who have made my life at UCSD special.

First and foremost, I would like to express my deepest appreciation to my Ph.D. advisor Professor Nuno Vasconcelos. It is my great honor to be supervised by him. He is an exceptional advisor with thorough knowledge of the subject, always enthusiastic and excited about the area. His professional instructions on research philosophy, technical writing, presentation, and argument as well as critical thinking helped me grow professionally and reshaped my research mindset. One important thing I want to mention is that he is a very humorous person. His language art usually makes the research discussion relaxed and easy. I am also extremely grateful to Professors Bhaskar D. Rao, David Kriegman, Truong Nguyen and Xiaolong Wang for being my committee members. Without their suggestions and supports, finishing the Ph.D. degree would be hard for me.

I am also thankful to all my mentors and collaborators of my two internship experiences, Yijun Li, Jingwan (Cynthia) Lu, Krishna Kumar Singh, Federico Perazzi at Adobe, and Zhaowei Cai, Hao Yang, Gurumurthy Swaminathan, Bernt Schiele, Stefano Soatto, Pietro Perona, Avinash Ravichandran, Marzia Polito at Amazon. Thanks to Yijun and Cynthia for giving me my first internship opportunity when I was still a junior Ph.D. student. The internship experiences offered me a unique opportunity to collaborate with some of the world's leading thinkers in artificial intelligence, working on the projects that push the frontiers of AI and science. The expertise and professionalism these researchers showed were instrumental to my Ph.D. study.

I am also grateful to my lab mates of SVCL, Zhaowei Cai, Bo Liu, Pedro Morgado, Yingwei Li, Yusheng Li, Yi Li, Tz-Ying Wu, Chih-Hui Ho, Jiacheng Cheng, Jiteng Mu, Zhiyuan Hu, Xudong Wang, Zhihang Ren, Brandon Leung, Xin Dong, Amir Persekian,

Gautam Nain, and other friends I met in UCSD, Kabir Nagrecha, Ruoyu Zhao, Bingyu Shen, Jose Joy, Mathew Sam, Yucheng Huang, Ravi Teja Konduru, Yongchuan Huang, Joseph Walker, etc. It has been a pleasure and an incredible learning experience to work with them. Their warm help in work and life made the challenging Ph.D. study easier and more cheerful. I would also like to extend my sincere thanks to all friends outside UCSD, Dashan Gao, Patrick Langechuan Liu, Subarna Tripathi, Geng Ji, Baoxiang Pan, Zhe Wang, Zhengdong Wang, Lu He, etc.

I would like to acknowledge all staff of the ECE department, International Students and Programs Office, Graduate Division, etc and my primary physician, other doctors and nurses of Student Health Center who helped me. Without their help, it is also impossible to complete my Ph.D. study. I gratefully acknowledge the support given by the National Science Foundation (NSF) for my advisor, which partially funded my work, and the NVIDIA GPU donations as well as the Nautilus platform.

Lastly and most importantly, I am deeply indebted to my family. The unconditional support from my parents has kept me relieved when I encountered difficulties. Their belief in me has kept my spirits and motivation high during this process. The most special thankfulness is given to my wife. Words cannot express my gratitude and appreciation to her twelve-year companionship starting from high school. She is the closest person to me in U.S. Together we have overcome many unimaginable challenges and crises in the past years. I am quite hopeful about our future lives and love. You are all that I love. I promise to stick by you through the rest of my life. Thanks should also go to my sister Fei Ren and her family. It is their caregiving that relieved my insecurity when I first stepped into North America.

Chapter 2 is, in full, based on the material as they appear in the publications of “Gradient-Based Algorithms for Machine Teaching”, Pei Wang, Kabir Nagrecha, Nuno Vasconcelos, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*

(CVPR), 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 3 is, in full, based on the materials as they appear in the publication of “Deliberative Explanations: visualizing network insecurities”, Pei Wang, Nuno Vasconcelos, In *Advances of Neural Information Processing Systems* (NeurIPS), 2019, and “SCOUT: Self-aware Discriminant Counterfactual Explanations”, Pei Wang, Nuno Vasconcelos, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2020, as well as the material as it appears in the submission of “A Generalized Explanation Framework for Visualization of Deep Learning Model Predictions”, Pei Wang, Nuno Vasconcelos, In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (TPAMI). The dissertation author was the primary investigator and author of these papers.

Chapter 4 is, in full, based on the material as they appear in the publications of “A Machine Teaching Framework for Scalable Recognition”, Pei Wang, Nuno Vasconcelos, In *Proceedings of IEEE International Conference on Computer Vision* (ICCV), 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 5 is, in full, based on the material as it appears in the submission of “Towards Crowd-Source Annotation of Expert Domain Data”, Pei Wang, Nuno Vasconcelos, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2023. The dissertation author was the primary investigator and author of this paper.

## VITA

- 2014 B.S. in Measurement Control Technology and Instrument, University of Electronic Science and Technology of China, China
- 2017 M.S. in Pattern Recognition and Intelligent System, Chinese Academy of Sciences, China
- 2022 Ph.D. in Electrical Engineering (Machine Learning and Data Science), University of California San Diego

## PUBLICATIONS

**Pei Wang**, Zhaowei Cai, Hao Yang, Gurumurthy Swaminathan, Nuno Vasconcelos, Bernt Schiele, and Stefano Soatto, “Omni-DETR: Omni-Supervised Object Detection with Transformers”, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022

**Pei Wang**, and Nuno Vasconcelos, “A Machine Teaching Framework for Scalable Recognition”, In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2021

**Pei Wang**, Kabir Nagrecha, Nuno Vasconcelos, “Gradient-based Algorithms for Machine Teaching”, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021

**Pei Wang**, Yijun Li, Nuno Vasconcelos, “Rethinking and Improving the Robustness of Image Style Transfer”, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021

**Pei Wang**, Yijun Li, Krishna Kumar Singh, Cynthia Lu, Nuno Vasconcelos, “IMAGINE: Image Synthesis by Image-Guided Model Inversion”, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021

Yunsheng Li, Lu Yuan, Yinpeng Chen, **Pei Wang**, Nuno Vasconcelos, “Dynamic Transfer for Multi-Source Domain Adaptation”, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021

Tz-Ying Wu, Pedro Morgado, **Pei Wang**, Chih-Hui Ho, Nuno Vasconcelos, “Solving Long-tailed Recognition with Deep Realistic Taxonomic Classifier”, In *Proceedings European Conference on Computer Vision (ECCV)*, 2020

**Pei Wang**, Nuno Vasconcelos, “SCOUT: Self-aware Discriminant Counterfactual Explanations”, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020



**Pei Wang**, Nuno Vasconcelos, “Deliberative Explanations: visualizing network insecurities”, In *Advances of Neural Information Processing Systems (NeurIPS)*, 2019

**Pei Wang**, Nuno Vasconcelos, “Towards Realistic Predictors”, In *Proceedings European Conference on Computer Vision (ECCV)*, 2018

## ABSTRACT OF THE DISSERTATION

### **Scaling up Recognition in Expert Domains with Crowd-source Annotations**

by

Pei Wang

Doctor of Philosophy in Electrical Engineering (Machine Learning and Data Science)

University of California San Diego, 2022

Professor Nuno Vasconcelos, Chair

The success of deep learning in image recognition is substantially driven by large-scale, well-curated data. On visual recognition of common objects, the data can be scalably annotated on online crowd-sourcing platforms because the labeling does not need any prior knowledge. However, the case is not true for images of expertise like biological or medical imaging in which labeling them needs background knowledge. Although data collection is still usually easy, the annotation is difficult. Existing self-supervised or semi-supervised solutions train a model that tries to learn from a small amount of labeled data and a large amount of unlabeled data. These solutions show good performances on common object recognition but have been found not to work effectively on fine-grained expert domains.

In this thesis, we propose a new solution with crowd source annotations to address the problem. Inspired by the fact that supervised learning on as much as data can always perform better, our method tries to scale up the annotation. This is implemented by two different approaches, machine teaching and human filtering. Machine teaching first teaches humans with a short carefully designed course to learn the expertise knowledge so that they can label the data later. Human filtering simplifies the process to a binary selection procedure without preceding training. Beyond these two approaches, a unified explanation framework is developed to generate visualizations that are merged into two approaches, enabling easier and more accurate annotation results. Experiments show that both methods significantly outperform various alternative approaches in several benchmarks. They have also been found to be versatile and can benefit from more advanced machine learning techniques in the future. Overall, we believe that this thesis opens up a new direction to think about the expert domain classification problem, in general.

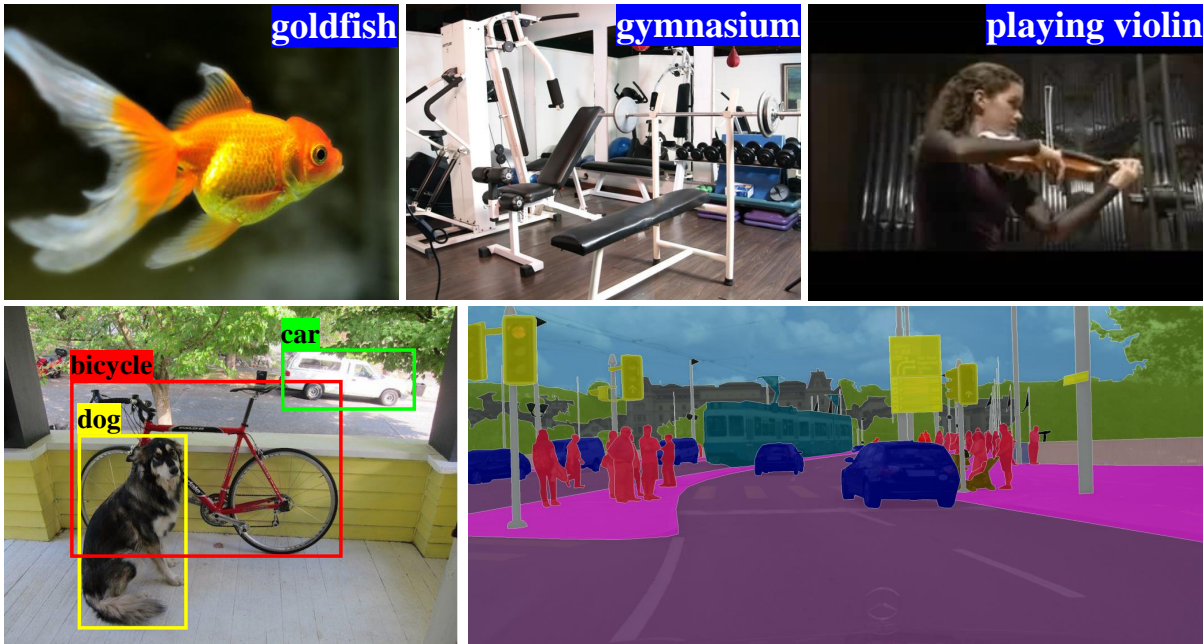
# Chapter 1

## Introduction

## 1.1 Learning in Expert Domains

Deep learning has been successfully applied to almost all computer vision fields, e.g., image recognition [1, 2, 3], object detection [4, 5, 6], image segmentation [7, 8, 9], etc. Its performance has surpassed or is comparable to humans on many tasks [10, 11, 12, 13]. Such success is largely driven by the easier access of well-curated large scale data, for example, ImageNet [14] for image recognition, Kinetics [15] for action recognition, Objects365 [16] for object detection, and comprehensive datasets like Open Images [17] or COCO [18]. Figure 1.1 presents some annotation formats. The annotation could be to identify the category of an object image (e.g. ‘goldfish’ or not), scene (e.g. ‘gymnasium’ or not) or video (e.g. ‘playing violin’ or not), to label the location of each object by a bounding box and its associated category, or to segment object instances given classes. Since these images usually contain common objects, humans can label them with their prior knowledge or just complete a simply preceding training, e.g. learning how to draw a decent bounding box for an object. On crowd-sourcing platforms like Amazon Mechanical Turk (MTurk) [19], the data can be, thus, *scalably* annotated. These datasets are able to be easily built and their sizes are also potentially enlargeable. This facility enables *scalable* recognition by supervised learning with full annotations on the whole accessible set, which is the most explored method in machine learning and maintains the lead performance on all vision tasks. When this is possible, we say that learning is *scalable* and the recognition can be scaled up.

However, the case is not true for expert domains, such as biological or medical imaging. These problems involve fine-grained classes. The class differences are very subtle and hard to distinguish for laypeople, for instance, to recognize the bird species of Figure 1.2. While data collection is still easy in these domains, annotation requires trained specialists and is too expensive if not plainly infeasible at scale. For example, while it is easy to crawl the web or deploy cameras in the wild to collect a large number of animal images, it is

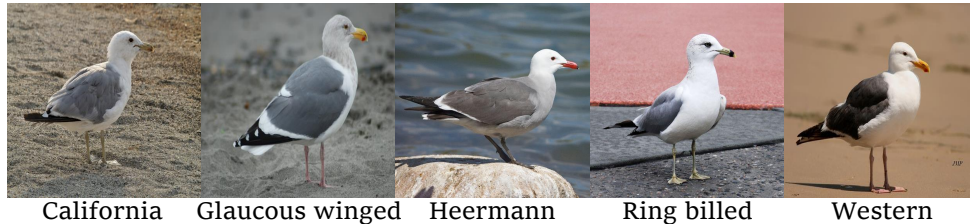


**Figure 1.1:** Annotations with different formats.

usually expensive to recruit the biologists or taxonomists needed to label them. Unlike recognition on coarse-grained domains, such difficulty makes scalable annotations very hard so as to scale up recognition in expert domains. This has become a main obstacle to deploy the deep learning in domains with expertise.

## 1.2 Existing Methods

Due to the difficulty of labeling, in expert domains, while it is typically not difficult to collect a large dataset, usually only a small part of examples can be realistically labeled by experts. This results in a small amount of expert-labeled data and a large amount of unlabeled data. Since most of data are without labels, rather than supervised learning, various other methods have arisen to tackle this label hungry problem. These include few-shot learning (FSL) that tries to learn just from the small labeled data [20, 21, 22], self-supervised learning (SeSL) [23, 24, 25] where a feature extractor is first learned both

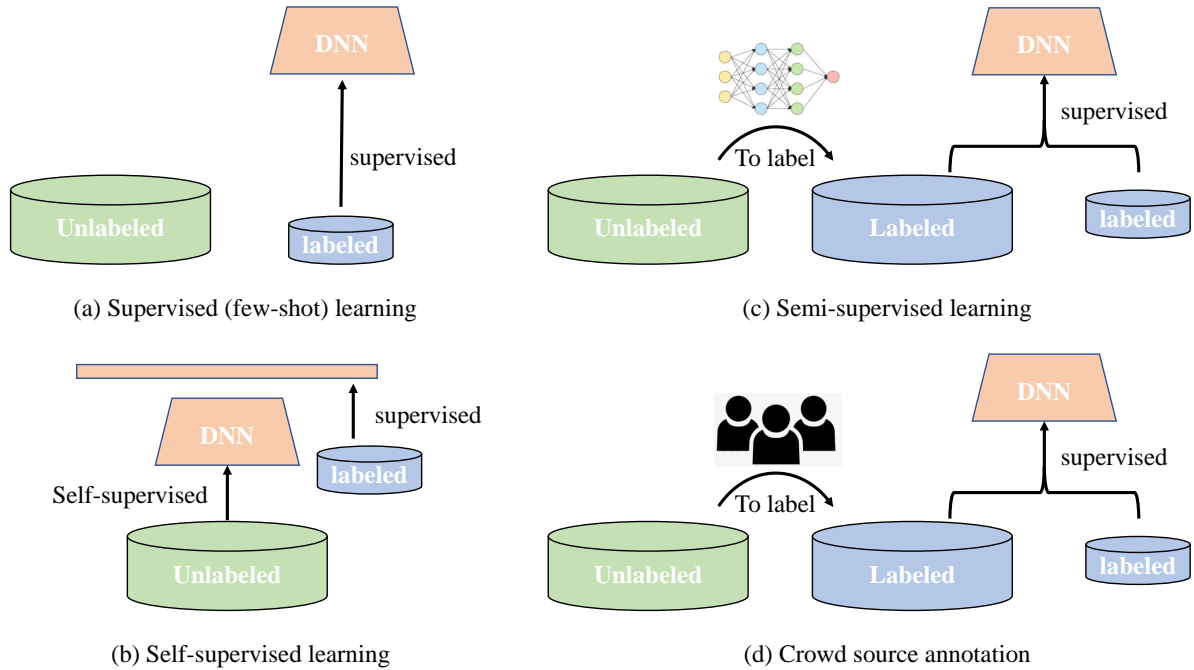


**Figure 1.2:** Examples of five gull species

on the large unlabeled data and small labeled data via some pretext tasks and then a top classifier on the labeled data with supervised learning, and semi-supervised learning (SSL) [26, 27, 28] in which the pseudo label produced by a network is used to supervise the training on the unlabeled data, etc. Figure 1.3 compares them. Despite having shown great success in coarse-grained recognition, these methods have been found not to work well on the label limited fine-grained domains [29, 30, 31].

This is partially because of many potential reasons. On few-shot learning, existing methods are evaluated without considering domain shift. The novel class with limited training examples and base classes with numerous examples are sampled from the same set. When the set is the expert domain, the evaluation setting becomes impractical because a large labeled base set is not available [31]. For SSL, the most popular approach is to use a classifier trained on the labeled data to produce pseudo-labels for unlabeled examples, based on their estimated confidence. Since the size of the labeled set is originally small, the classifier is not accurate. In addition to this problem, deep networks produce poorly calibrated confidence estimation [32, 33]. The two effects compound and pseudo-labels are not trustworthy. The key of SeSL is to learn the feature representation. However, on fine-grained recognition, many classes differ by subtle details, e.g., color or texture patterns of a body part, which could potentially mess up some critical components of SeSL like data augmentation [30].

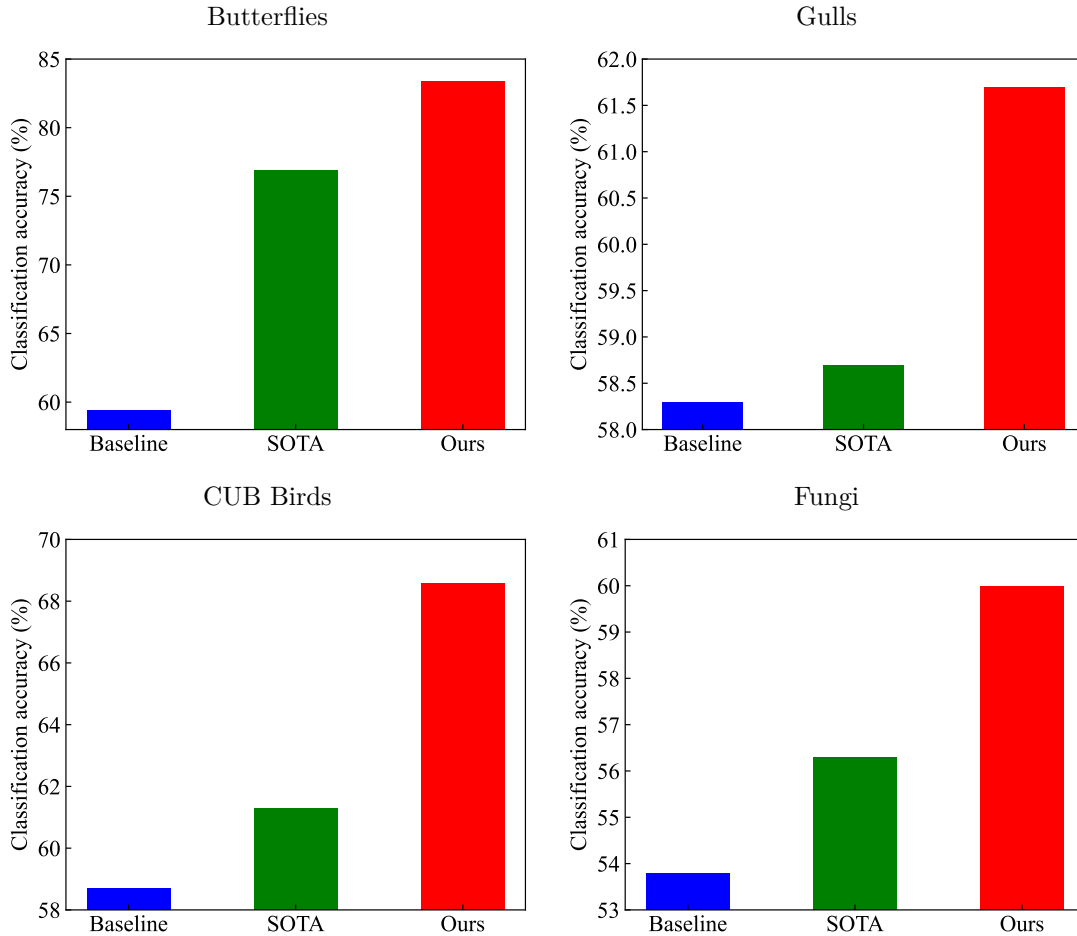
For SSL and SeSL, most methods are evaluated on common object recognition like CIFAR [34] and ImageNet [14]. While bypassing human annotation is always desirable to



**Figure 1.3:** The comparison among different methods for annotation limited problem given a small labeled set and a large unlabeled set.

lower the cost, the scalability of the annotation process is not a major concern for common object recognition. The image is relatively easy and cheap to annotate by crowd-sourcing, and for which scalable annotation is already possible on Mturk and similar platforms. In this case, we can use supervised learning which is unbeatable by far in practice and it makes little practical sense to try SSL-type of approaches. This is not the case for problems where annotation requires domain experts, and the costs of scaling it to an ImageNet-style dataset size are unbearable. Figure 1.4 summarizes and compares performances of different methods on several representative benchmarks [29, 35]. It can be seen that these methods, in general, are just barely better than the baseline of standard supervised training on the small labeled set. This suggests that the current solutions are not on the right track.





**Figure 1.4:** Classification accuracy comparison of different methods on expert domain data. ‘Baseline’: the supervised training baseline only on the labeled set. ‘SOTA’: the state of the art method.

### 1.3 Contributions of the Thesis

All of the above methods are trying to learn from the small labeled data. However, they are still upper bounded by the supervised learning on both the labeled and unlabeled sets if we can obtain its annotation. There is a huge gap between such supervised learning upper bound and existing methods [29]. Motivated by this observation, because we have known that learning from large labeled data works the best, in this thesis, we pursue an alternative solution. Instead of trying to learn from the small labeled data, we try

to make the small labeled data bigger. We come up with an idea to tackle the problem by scaling up the annotation on expert domains with crowd-sourcing. The main idea behind it is to teach or guide laypeople to annotate expert-domain unlabeled data so that they can provide professional annotations even if they do not have any prior knowledge. With such achievement, although the scalable annotating is infeasible by experts, it can be achieved by common humans, since several crowdsourcing platforms have appeared in recent years [19, 36, 37], which makes it easier to recruit large numbers of image annotators online. Once the label is provided for the unlabeled data, supervised learning methods can be directly applied, which is the most well-explored machine learning method and still the upper bound. The idea is illustrated in Figure 1.3 (d). Different from SSL, unlabeled examples are labeled by trained humans instead of a machine, thus leveraging their strong learning ability for low shot [38, 39, 40] and confidence calibration [41]. A significant improvement has been achieved (shown with a red bar in Figure 1.4).

Our solution with crowd source annotations has two implementations by machine teaching and human filtering. They both rely on crowd source human workers to label the unlabeled data. Machine teaching first carefully designs a short course in order to teach a few fine-grained species to humans so that they can label new data later. Human filtering simplifies the teaching and labeling process to a binary similarity comparison task. To pursue a more effective and accurate labeling, visualization based explanations are adopted to guide the labeling procedure. Beyond existing attribution based explanations, two new families of explanations are proposed and unified into a single framework. These two explanations are combined with machine teaching and human filtering, and found to be helpful to the labeling process.

### 1.3.1 Gradient-based Algorithms for Machine Teaching

The problem of machine teaching is considered. A new formulation is proposed under the assumption of an optimal student, where optimality is defined in the usual machine learning sense of empirical risk minimization. This is a sensible assumption for machine learning students and for human students in crowdsourcing platforms, who tend to perform at least as well as machine learning systems. It is shown that, if allowed unbounded effort, the optimal student always learns the optimal predictor for a classification task. Hence, the role of the optimal teacher is to select the teaching set that minimizes student effort. This is formulated as a problem of functional optimization where, at each teaching iteration, the teacher seeks to align the steepest descent directions of the risk of (1) the teaching set and (2) entire example population. The optimal teacher, denoted MaxGrad, is then shown to maximize the gradient of the risk on the set of new examples selected per iteration. MaxGrad teaching algorithms are finally provided for both binary and multiclass tasks, and shown to have some similarities with boosting algorithms. Experimental evaluations demonstrate the effectiveness of MaxGrad, which outperforms previous algorithms on the classification task, for both machine learning and human students from MTurk, by a substantial margin.

### 1.3.2 A Generalized Explanation Framework for Visualization of Deep Learning Model Predictions

Attribution-based explanations are popular in computer vision but of limited use for fine-grained classification problems typical of expert domains, where classes differ by subtle details. In these domains, users also seek understanding of “why” a class was chosen and “why not” an alternative class. A new *Generalized explanation Framework* (GALORE) is proposed to satisfy all these requirements, by unifying attributive explanations

with explanations of two other types. The first is a new class of explanations, denoted *deliberative*, proposed to address the “why” question, by exposing the network insecurities about a prediction. The second is the class of counterfactual explanations, which have been shown to address the “why not” question but are now more efficiently computed. GALORE unifies these explanations by defining them as combinations of attribution maps with respect to various classifier predictions and a confidence score. An evaluation protocol that leverages object recognition (CUB200) and scene classification (ADE20K) datasets combining part and attribute annotations is also proposed. Experiments show that confidence scores can improve explanation accuracy, deliberative explanations provide insight into the network deliberation process, the latter correlates with that performed by humans, and counterfactual explanations enhance the performance of human students in machine teaching experiments.

### 1.3.3 A Machine Teaching Framework for Scalable Recognition

We consider the scalable recognition problem in the fine-grained expert domain where large-scale data collection is easy whereas annotation is difficult. Existing solutions are typically based on semi-supervised or self-supervised learning. We propose an alternative new framework, MEMORABLE, based on machine teaching and online crowdsourcing platforms. A small amount of data is first labeled by experts and then used to teach online annotators for the classes of interest, who finally label the entire dataset. Preliminary studies show that the accuracy of classifiers trained on the final dataset is a function of the accuracy of the student annotators. A new machine teaching algorithm, CMaxGrad, is then proposed to enhance this accuracy by introducing explanations in a state-of-the-art machine teaching algorithm. For this, CMaxGrad leverages counterfactual explanations, which take into account student predictions, thereby providing feedback that is student-specific, explicitly addresses the causes of student confusion, and adapts to the level of competence

of the student. Experiments show that both MEMORABLE and CMaxGrad outperform existing solutions to their respective problems.

### 1.3.4 Towards Professional Level Crowd Annotation of Expert Domain Data

Image recognition on expert domains is usually fine-grained and requires expert labeling, which is costly. This limits dataset sizes and the accuracy of learning systems. To address this challenge, we consider annotating expert data with crowdsourcing. This is denoted as PrOfeSsional lEvel cRowd (POSER) annotation. A new approach, based on semi-supervised learning (SSL) and denoted as SSL with human filtering (SSL-HF) is proposed. It is a human-in-the-loop SSL method, where crowd-source workers act as filters of pseudo-labels, replacing the unreliable confidence thresholding used by state-of-the-art SSL methods. To enable annotation by non-experts, classes are specified implicitly, via positive and negative sets of examples and augmented with deliberative explanations, which highlight regions of class ambiguity. In this way, SSL-HF leverages the strong low-shot learning and confidence estimation ability of humans to create an intuitive but effective labeling experience. Experiments show that SSL-HF significantly outperforms various alternative approaches in several benchmarks.

## 1.4 Organization of the Thesis

The rest of the thesis is structured as follows. Chapter 2 introduces the MaxGrad machine teaching algorithm. It is the foundation that we use to design our first crowd source annotation method, MEMORABLE. In Chapter 3, in order to improve the human annotation performance, we introduce a unified explanation framework, GALORE, which can generate two new families of explanations, deliberative explanations and discriminant

counterfactual explanations. These two explanations will be shown to help the machine teaching methods and human filtering in the following chapters. In Chapter 4, we combine MaxGrad with counterfactual explanation and present the MEMORABLE method. Chapter 5 introduces another crowd source annotation method SSL-HF, by human filtering, which can be enhanced by deliberative explanations. Chapter 6, finally, does a comprehensive comparison and discussion on the two methods and two explanations. We summarize and conclude the thesis.

## Chapter 2

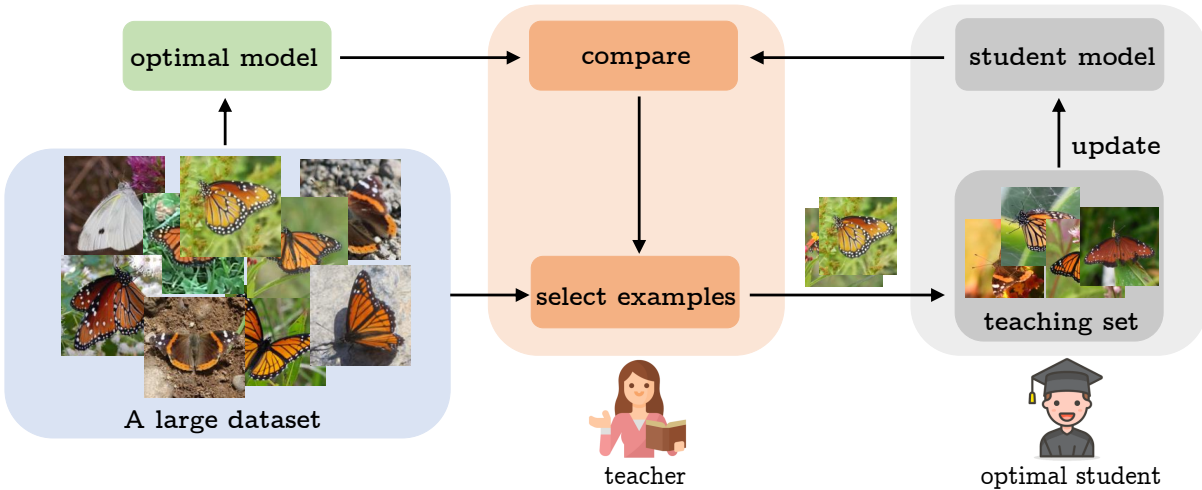
# Gradient-based Algorithms for Machine Teaching

## 2.1 Introduction

The success of deep learning has been driven, in large part, by the availability of large and carefully curated datasets for tasks such as image recognition [14, 42], action recognition [15, 43], object detection [18], etc. These datasets usually contain everyday objects, actions, or scenes and can be scalably annotated on crowdsourcing platforms such as Amazon Mechanical Turk (MTurk). This is, however, usually not true for expert domains, such as biology or medical imaging. While data collection can still be easy in these domains, annotations require highly specialized and domain specific knowledge. This is beyond the reach of crowdsourcing annotators. On the other hand, annotation by specialists is usually too expensive and rarely feasible at a large scale. This has motivated extensive research in alternative and less label-intensive forms of learning, including few-shot learning [20, 44], transfer learning [45, 46], semi-supervised learning [47, 48], and self-supervised learning [49, 50]. However, these approaches usually underperform supervised learning from large and fully labeled datasets. In result, there has recently been interest in machine teaching algorithms capable of training crowdsource annotators to label data from specialized domains.

The goal of machine teaching is to design systems that can teach students efficiently and automatically. Machine teaching is a broad research problem [51], where humans can utilize domain knowledge to teach machines or vice-versa. In this work, we restrict the discussion to the narrow task of image classification, where a machine teaches human learners to discriminate between different image classes. Although the proposed ideas are general, we target the application of teaching image annotators in crowd-sourcing platforms. This exploits the fact that a relatively small annotated dataset can be leveraged to train crowd workers, which can then annotate large numbers of images, enabling scalable supervised learning of image classifiers. While classification has been the task of choice for much machine teaching work, it should be noted that several other tasks and applications





**Figure 2.1:** Iterative machine teaching process.

have also been investigated [52, 53, 54].

The machine teaching set-up considered in this work is the iterative interaction set-up of Figure 2.1. At each iteration, the teacher selects new examples from a large dataset, to complement a small set of examples, known as the *teaching set*, which is used by the student to learn the target task. By comparing the current student model and the optimal model for the large dataset, the teacher seeks to select the examples that most help the student learn. The central question in this set-up is how to select the teaching set. Ideally, this set should pack as much information for class discrimination as possible into the smallest number of examples.

In the literature, there have been many attempts to design optimal teaching algorithms [55, 56, 57, 58]. This usually requires the assumption of certain student properties. Although past works have proposed different student models, these frequently rely on assumptions that are questionable for the crowdsourcing context. For example, a popular assumption [55, 59, 60] is that the student only has access to a countable set of hyperplane hypotheses. While justified by the fact that human students have limited ability and memory, this assumption overly underestimates their learning ability. In fact,

several machine teaching works explicitly assume that students have limited capacity or are otherwise sub-optimal learners [58, 61, 55, 62]. This is not supported by studies with real students, which found that humans have strong learning ability [63, 64, 65, 66].

In this work, we assume that the student is an optimal learner. Optimality is defined in the standard machine learning sense, i.e. that the student learns a predictor of minimum empirical risk in the teaching set. This always holds for machine learning students, which are defined in this way, and is sensible for human students, who usually do not underperform machine learning students, especially on few-shot learning scenery in practice. It does assume that students are engaged in the learning task, i.e. giving their best effort. This is sensible in the crowdsourcing scenario, where students are free-willing participants rated by their task performance. We show that, if allowed unbounded effort, the optimal student will always learn the optimal predictor for the task. This implies that the only role of the teacher is to optimize learning speed, i.e. select the teaching examples that enable the student to learn with least effort.

We then formulate the search for the optimal teacher as a problem of functional optimization where, at each teaching iteration, the teacher aims to align the steepest descent direction of the teaching set risk with that of the empirical risk over the entire example population. This is shown to have as optimal solution the *MaxGrad* teacher, which maximizes the gradient of the risk on the set of new examples selected per iteration. MaxGrad teaching algorithms are finally provided for both binary and multiclass tasks, and shown to have some similarities with boosting algorithms [67, 68, 69]. Experimental evaluations demonstrate the effectiveness of MaxGrad, which outperforms previous algorithms on the classification task, for both machine learning and human students from MTurk.

## 2.2 Related Work

**Simulated studies:** In the past two decades, a variety of algorithms have been proposed to model the teacher-student interaction and seek the optimal teaching sequence. [70] explored several heuristics for the selection of the teaching set, based on insights derived from active learning, including a preference for points closest to the boundary, a handcrafted indicator of classification difficulty, curriculum learning, and a coverage model. [71] explored the use of recurrent neural networks as models of student learning. [72] modeled student learning as a Bayesian update process. [73, 74] used reinforcement learning based models to develop teaching policies for computer-based tutoring systems. All these methods have been developed and evaluated with synthetic data or handcrafted features, and did not explore the teaching of human learners with natural images. Note that there are some related algorithm families to machine teaching, including active learning [75, 76], few-shot learning [20, 21], curriculum learning [77, 78] and knowledge distillation [79]. For example, the main difference from active learning is that in the latter the learner selects examples without knowing the ground truth. In machine teaching, examples are selected by the teacher, who knows all labels. We recommend [56, 51] for extensive comparisons.

**Human studies:** Most of existing literature on human evaluations only work on simple binary classification problem [62, 55, 80]. A representative is STRICT [55]. It simulates the student as a hyperplane in a finite hypothesis space. The learning process is modeled as a Markov chain, assuming that learners perform a random walk in hypothesis space, according to the teacher’s feedback. Expected error rate is the criterion for teaching set selection. Since its minimization is NP-hard, a surrogate objective is optimized in a greedy manner. Following STRICT, many extensions or generalizations have been proposed [80, 59, 60]. For example, beyond pure label feedback, methods have been proposed to account for feature-based feedback, both for synthetic data [80] and real images [59], using an attribution map [81]. [59] also extended STRICT to multiclass

problems.

Alternatively to STRICT, [56, 57] modeled teaching as an iterative process and the learner as a linear classifier, which is updated at each iteration based uniquely on the example seen at that iteration. Beyond [56], [57] treats the student network as a black-box, which more closely resembles real student learning. [61] approximates the student’s class conditional distribution given the teaching set with a Gaussian random field but it is designed for online learning, a different setting from that studied in this work. All these methods assume that the learner is sub-optimal or has limited capacity. However, there is little evidence to support this. On the contrary, many studies have found that humans have strong learning ability [63, 64, 65, 66], which is also intuitive. We argue that assuming an optimal learner is more sensible in very specialised domains, at least for image classification in the crowdsourcing context.

**Feature space:** The practical implementation of machine teaching requires a feature extractor to implement the simulated student. Since several prior works were introduced before the popularization of deep learning, they rely on handcrafted features [55, 60]. These are unlikely to be close to human perception and tend to produce low-accuracy classifiers. More recently, it has become standard practice to use features extracted by a deep convolutional network, which is a better model of human perception [82, 83, 84] and produces better classifiers. This is a practice that we also adopt. However, previous works have used networks fine-tuned on a dataset from the target domain [59]. This vastly simplifies the teaching problem, as it is equivalent to assuming that the student already is an expert in the target domain before the teaching starts. We instead rely on a model pretrained on ImageNet. This reflects the assumption that the student is competent in generic image classification tasks, but has no experience in the target domain. This assumption usually holds for the crowdsource setting, whenever the target domain requires specific expertise.

**Other approaches:** Recently, several works have investigated the use of explanations during the teaching phase, to improve teaching performance. The results are so far inconclusive, as these works show limited improvements [59, 60, 80], particularly in light of the noise inherent to human evaluations, or even a negative impact [59, 60]. While MaxGrad could in principle be combined with visual explanations, we leave this for future work. There have also been proposals for interactive online machine teaching [61], where the selection of teaching examples is not based on a simulated student, but derived from the responses of human users in real-time. However, online updates are costly and difficult to scale to large numbers of simultaneous users. The extension of the ideas used to derive MaxGrad to this setting is a topic that we intend to investigate in the future.

## 2.3 Gradient-Based Machine Teaching

In this section, we introduce the MaxGrad algorithm.

### 2.3.1 Machine Teaching

In machine teaching for classification, the goal of the teacher is to assemble a *teaching set*  $\mathcal{L} = \{(x_i^l, y_i^l)\}_{i=1}^K$  of examples  $x_i^l$  and class labels  $y_i^l$ , which a student uses to learn a classifier. In this work, we adopt the pool-based teaching setting [51]. This assumes that the teacher has access to a much larger example dataset  $\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$  from which it selects a subset to assemble  $\mathcal{L}$ . This is different from synthesis teaching where the teaching examples are synthetically generated. Pool-based teaching is more realistic for image labeling applications, because artificial images may appear nonsensical to a (human) student. The goal of machine teaching is to enable the student to learn the optimal predictor  $f^*(x)$  for the entire example population  $\mathcal{D}$ , from the smallest teaching set  $\mathcal{L}$ , i.e. the smallest possible number of teaching examples  $K$ .

As usual in machine learning, the optimal predictor  $f^*$  is defined as the predictor that minimizes the risk  $\mathcal{R}_{\mathcal{D}}[f]$  associated with a loss function on  $\mathcal{D}$ . The details of the loss function depend on the task. For simplicity, we discuss binary classification firstly and extend all ideas to the multi-class setting in section 2.3.5. For binary classification,  $y \in \mathcal{Y} = \{-1, +1\}$ ,  $f(x)$  maps  $x \in \mathcal{X}$  to  $\mathbb{R}$  and the optimal predictor is

$$f^* = \arg \min_f \mathcal{R}_{\mathcal{D}}[f] = \arg \min_f \sum_{(x_i, y_i) \in \mathcal{D}} \phi(y_i f(x_i)), \quad (2.1)$$

where  $\phi(\cdot)$  is a margin loss function. This predictor is assumed known to the teacher.

The end-goal of the teacher is to assemble the teaching set  $\mathcal{L} \subset \mathcal{D}$  that achieves the best trade-off between two conflicting requirements: the student learns the optimal predictor  $f^*$  while spending the least effort. This reflects the fact that longer teaching sequences lead to better student performance, but the student has a limited set of learning resources, e.g. a limited attention span. For example, image annotators on crowd-sourcing platforms are well known to drop tasks that are too tedious to master. In this work, we assume that student effort is proportional to the cardinality of the teaching set  $|\mathcal{L}|$ . This leads to the formulation of the optimal teacher as the one which minimizes some distance  $d(f^*, f^s)$  between the predictor  $f^s$  learned by the student from  $\mathcal{L}$  and the optimal predictor  $f^*$ , under a constraint on student effort  $|\mathcal{L}| \leq \zeta$ .

### 2.3.2 The Optimal Student Assumption

In this work, we rely on the assumption that the student is an optimal learner.

**Definition 1** *The student is an optimal learner with respect to loss  $\phi$  if and only if, given*

a teaching set  $\mathcal{L}$ , it learns the predictor that minimizes the risk defined by  $\phi$  and  $\mathcal{L}$ ,

$$\mathcal{R}_{\mathcal{L}}(f) = \sum_{(x_i, y_i) \in \mathcal{L}} \phi(y_i f(x_i)). \quad (2.2)$$

Note that the risk of (2.2) is defined over  $\mathcal{L}$ , the teaching set that the student has access to, not the entire population  $\mathcal{D}$ . The optimal student assumption holds trivially when the student is a machine learning algorithm, because learning algorithms are designed to minimize (2.2). Since human learners tend to perform at least as well as machine learning algorithms for most tasks, especially learning on few shot examples, it is a sensible assumption for human students as well. Under this definition of student, the machine teaching problem can then be formalized as a bilevel optimization problem.

**Definition 2** *Under the assumption of an optimal learner with respect to loss  $\phi$ , a teacher is optimal if and only if it produces the teaching set*

$$\mathcal{L}^* = \operatorname{argmin}_{\mathcal{L}} d(f^*, f^s(\mathcal{L})) \quad (2.3)$$

$$f^s(\mathcal{L}) = \operatorname{argmin}_f \sum_{(x_i, y_i) \in \mathcal{L}} \phi(y_i f(x_i)). \quad (2.4)$$

$$|\mathcal{L}| \leq \zeta \quad (2.5)$$

where  $f^*$  is given by (2.1),  $d(.,.)$  is a distance function, and  $\zeta$  a bound on student effort to process the examples in  $\mathcal{L}$ .

In what follows, the teaching process is assumed to be iterative.

**Definition 3** *An iterative machine teaching procedure iterates between a step of example selection, by the teacher, and a learning step by the student. At iteration  $t$ , the teacher produces a teaching set  $\mathcal{L}^t$ , which the student uses to learn a predictor  $f^t(x)$ . The teacher*

then selects from  $\mathcal{D}^t = \mathcal{D} - \mathcal{L}^t$  the examples to add to  $\mathcal{L}^t$  in order to produce  $\mathcal{L}^{t+1}$ . The student starts the process with an initial predictor  $f^0(x)$ . This can be derived from prior experience or  $f^0(x) = 0$ .

The following result is an immediate consequence of these definitions.

**Corollary 1** *Consider the iterative machine teaching procedure of Definition 3 and assume that the teacher selects at least one new example per learning iteration. If  $\zeta$  is large enough, the optimal student of Definition 1 is guaranteed to learn the optimal predictor  $f^*$  of (2.1) after a finite number of iterations.*

**Proof** Under the optimal student assumption, the predictor learned by the student at iteration  $t$  is

$$f^t = \arg \min_f \mathcal{R}_{\mathcal{L}^t}[f] = \arg \min_f \sum_{(x_i, y_i) \in \mathcal{L}^t} \phi(y_i f(x_i)). \quad (2.6)$$

If the teacher selects at least one new example per iteration,  $\mathcal{L}^t$  increases with  $t$ , i.e.  $\mathcal{L}^{t-1} \subset \mathcal{L}^t$ . Since  $\mathcal{D}$  has finite size  $n$ ,  $\exists k \leq n$  s.t.  $\mathcal{L}^k = \mathcal{D}$ . It follows that, if  $\zeta \geq |\mathcal{D}|$ , the student will eventually learn from  $\mathcal{L}^k$ . From (2.6) and (2.1) it follows that  $f^k = f^*$ .

In summary, for an optimal student and a sufficient level of effort, the distance  $d(f^*, f^s)$  of (2.3) always converges to zero. It follows that the only role of the teacher is to optimize learning speed, i.e. select the set of examples that enable the student to learn with the least effort. We next define an optimal teacher from this point of view. This, however, requires a brief review of basic concepts in functional optimization.



### 2.3.3 Functional Optimization

Given two vector spaces  $\mathcal{X}$ ,  $\mathcal{Y}$  and a differentiable function  $R: \mathcal{X} \rightarrow \mathcal{Y}$ , the differential  $dR(u, \psi)$  of  $R$  at  $u \in \mathcal{X}$  in the direction  $\psi \in \mathcal{X}$  is given by

$$dR(u, \psi) = \left. \frac{d}{d\tau} R(u + \tau\psi) \right|_{\tau=0}. \quad (2.7)$$

For example, the margin loss function  $\mathcal{M}(f) = \phi(y(x)f(x))$  has differential  $d\mathcal{M}(f, \psi) = y\phi'(yf)\psi$ . Given a set of directions  $\Psi = \{\psi_1, \dots, \psi_n\}$  such that  $\psi_i \in \mathcal{X}, \forall i$ , the gradient of  $R$  with respect to  $\Psi$  at  $u$  is the vector

$$\nabla_{\Psi} R(u) = (\langle dR(u, \psi_1), \psi_1 \rangle, \dots, \langle dR(u, \psi_n), \psi_n \rangle)^T. \quad (2.8)$$

Let  $Sp(\Psi)$  be the span of  $\Psi$  and  $\gamma$  a direction in  $Sp(\Psi)$ , i.e.  $\gamma = \sum_i \alpha_i \psi_i$  for some vector  $\alpha$ . The derivative of  $R$  at  $u$  along direction  $\gamma \in Sp(\Psi)$  is

$$\partial_{\gamma} R(u) = \langle \nabla_{\Psi} R, \alpha \rangle, \quad (2.9)$$

where  $\langle \alpha, \beta \rangle = \int \alpha(x)\beta(x)dx$  when  $\alpha$  and  $\beta$  are functions and  $\langle \alpha, \beta \rangle = \sum_i \alpha_i \beta_i$  when they are finite dimensional vectors.

A dataset  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ , defines a set of canonical directions  $\Psi(\mathcal{D}) = \{\delta(x - x_i)\}_{i=1}^n$ , where  $\delta(x)$  is the Dirac delta function. The differentials of the margin loss along these directions are  $d\mathcal{M}(f, \psi_k) = y\phi'(yf)\delta(x - x_k)$  and the empirical risk

$$R_{\mathcal{D}}(f) = \sum_{(x_i, y_i) \in \mathcal{D}} \phi(y_i f(x_i)) = \sum_{(x_i, y_i) \in \mathcal{D}} \mathcal{M}(f(x_i)) \quad (2.10)$$

has gradient

$$\nabla_{\Psi(\mathcal{D})}R_{\mathcal{D}}(f) = (w_1, \dots, w_n)^T, \quad w_i = y_i\phi'(y_i f(x_i)) \quad (2.11)$$

where  $\phi'$  is the derivative of  $\phi$ . For any function  $g$  in the span of  $\Psi(\mathcal{D})$ , i.e.

$$g(x) = \sum_i g(x_i)\delta(x - x_i), \quad (2.12)$$

the derivative of the risk at  $f$  along the direction of  $g$  is

$$\partial_g R_{\mathcal{D}}(f) = \sum_{(x_i, y_i) \in \mathcal{D}} w_i g(x_i). \quad (2.13)$$

The risk  $R_{\mathcal{D}}(f)$  is minimized at  $f^*$  if  $\partial_g R_{\mathcal{D}}(f^*) = 0, \forall g \in Sp(\Psi(\mathcal{D}))$ , which holds if

$$\nabla_{\Psi(\mathcal{D})}R_{\mathcal{D}}(f^*) = 0. \quad (2.14)$$

### 2.3.4 The Optimal Teacher

With these results we are ready to introduce a criterion for teacher optimality, under the iterative teaching procedure of Definition 3. We start by introducing the set of permissible choices for the teaching set, i.e the set of teaching sets that the teacher is allowed to choose from at iteration  $t$ . Under the iterative teaching procedure,  $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \mathcal{N}^t$ , i.e. the teacher augments  $\mathcal{L}^{t-1}$  with a set of examples  $\mathcal{N}^t$  not contained in it, which we denote as the *novel examples* of iteration  $t$ . The set of permissible choices includes all such novel sets

$$\mathcal{P}^t(\tau) = \{\mathcal{N} \subset \mathcal{D}^{t-1} \mid |\mathcal{N}| \leq \tau\} \quad (2.15)$$

The parameter  $\tau$  upper-bounds the student effort per teaching iteration, enabling

the teacher to control the trade-off between number of teaching iterations and student effort. Since, the total effort spent up to iteration  $t$  is upper-bounded by  $t\tau$ , it follows from (2.5) that the student can learn for up to  $T = \zeta/\tau$  iterations. In the iterative setting, it is easier to control the level of effort per iteration than the overall level of effort  $\zeta$ . In fact, the standard practice in the literature [55, 59, 56] is to allow a single novel example per iteration, i.e. set  $\tau = 1$ , and then limit the number of iterations  $T$ . The definition of set of permissible choices above loosens this constraint.

The question for the teacher is how to select the set of novel examples  $\mathcal{N}^t$  in some optimal way. We next introduce the definition of optimality used in this work.

**Definition 4** *Consider the iterative machine teaching procedure of Definition 3, with optimal student of Definition 1. Let  $g^*$  be the direction of steepest descent of the population risk*

$$g^* = \arg \min_{g \in Sp(\mathcal{D}), \|g\|=1} \partial_g R_{\mathcal{D}}(f^t) \quad (2.16)$$

and  $\mathcal{P}^t$  be the set of permissible choices for iteration  $t$ . The optimal teacher selects the set of novel examples

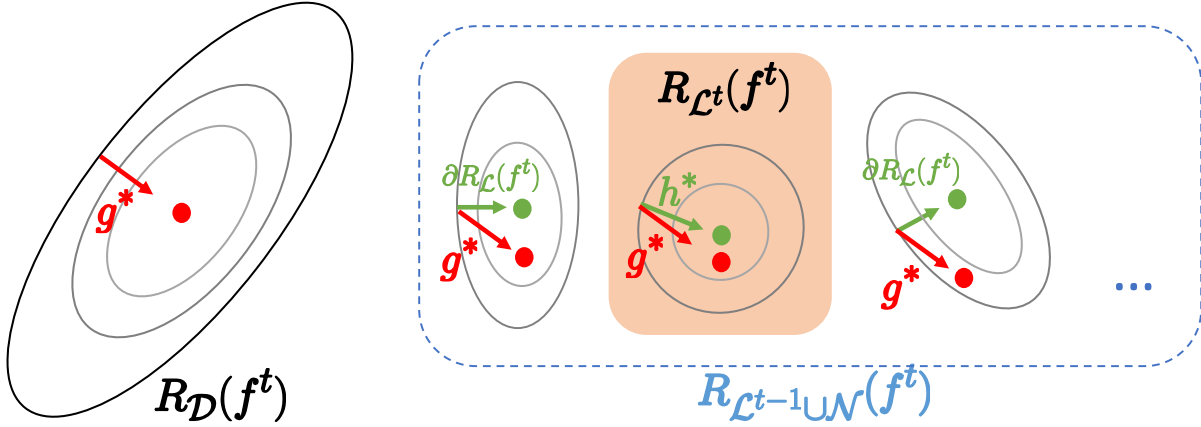
$$\mathcal{N}^t = \arg \max_{\mathcal{N} \in \mathcal{P}^t} \langle g^*, h^*(\mathcal{L}^{t-1} \cup \mathcal{N}) \rangle \quad (2.17)$$

where

$$h^*(\mathcal{L}) = \arg \min_{h \in Sp(\mathcal{L}), \|h\|=1} \partial_h R_{\mathcal{L}}(f^t) \quad (2.18)$$

is the direction of steepest descent on the teaching set risk. The teaching set is then updated into  $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \mathcal{N}^t$ .

This definition encodes the fact that the ideal teaching set  $\mathcal{L}^t$  would allow the student to give steepest descent steps on the population risk  $R_{\mathcal{D}}$ , to enable the fastest progress towards  $f^*$ . However, the student does not have access to  $\mathcal{D}$ , only to  $\mathcal{L}^{t-1}$  and a set of novel examples from  $\mathcal{P}^t$ . The optimal teacher of (2.17) selects the novel set  $\mathcal{N}^t \in \mathcal{P}^t$



**Figure 2.2:** Novel set selection by MaxGrad.

that leads to the teaching set  $\mathcal{L}^t$  whose steepest descent direction  $h^*(\mathcal{L}^t)$  is *closest* to the steepest descent direction  $g^*$  of  $\mathcal{D}$ . This is illustrated in Figure 2.2.

To derive the solution of (2.17), we leverage the following property of functional derivatives.

**Lemma 1** *For any decomposition of  $\mathcal{D} = \mathcal{A} \cup \mathcal{B}$  into two disjoint subsets  $\mathcal{A}$  and  $\mathcal{B}$  (such that  $\mathcal{A} \cap \mathcal{B} = \emptyset$ ) and any direction  $g \in Sp(\Psi(\mathcal{D}))$*

$$\partial_g R_{\mathcal{D}}(f) = \partial_g R_{\mathcal{A}}(f) + \partial_g R_{\mathcal{B}}(f). \quad (2.19)$$

**Proof** Assume without loss of generality that  $\mathcal{A} = \{(x_1, y_1), \dots, (x_m, y_m)\}$  and  $\mathcal{B} = \{(x_{m+1}, y_{m+1}), \dots, (x_n, y_n)\}$  for any  $1 < m < n$ . Then, it follows from (2.11) that

$$\nabla_{\Psi(\mathcal{D})}^T R_{\mathcal{D}}(f) = (w_1, \dots, w_m, w_{m+1}, \dots, w_n)^T \quad (2.20)$$

$$= \left( \nabla_{\Psi(\mathcal{A})}^T R_{\mathcal{A}}(f), \nabla_{\Psi(\mathcal{B})}^T R_{\mathcal{B}}(f) \right) \quad (2.21)$$

$$= \left( \nabla_{\Psi(\mathcal{A})}^T R_{\mathcal{A}}(f), 0 \right) + \left( 0, \nabla_{\Psi(\mathcal{B})}^T R_{\mathcal{B}}(f) \right) \quad (2.22)$$

$$= \nabla_{\Psi(\mathcal{D})}^T R_{\mathcal{A}}(f) + \nabla_{\Psi(\mathcal{D})}^T R_{\mathcal{B}}(f) \quad (2.23)$$

---

**Algorithm 1 MaxGrad**


---

**Input** Data  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ , codewords  $\mathcal{Y}$ , max iter.  $T$ , effort  $\tau$ .

- 1: **Initialization:**  $\mathcal{L}^0 \leftarrow \emptyset$ ,  $f^1$ ,  $\mathcal{D}^0 \leftarrow \mathcal{D}$ .
- 2: **for**  $t = \{1, \dots, T\}$  **do**
- 3:   compute  $\xi(x_i)$  for all examples in  $\mathcal{D}^{t-1}$ .
- 4:   order examples by decreasing  $\xi(x_i)$  and select top  $\tau$  to create  $\mathcal{N}^t$ .
- 5:   teaching set update:  $\mathcal{L}^t \leftarrow \mathcal{L}^{t-1} \cup \mathcal{N}^t$
- 6:   student update:  $f^{t+1} = f^*(\mathcal{L}^t)$ .
- 7:    $\mathcal{D}^t \leftarrow \mathcal{D}^{t-1} \setminus \mathcal{N}^t$
- 8: **end for**

**Output**  $\mathcal{L}^t$

---

	binary	multi-class
$\mathcal{Y}$	$\{-1, +1\}$	$\{y^1, \dots, y^C\}, y^i \in \mathcal{R}^d$
$\xi(x_i)$	$(\phi'(y_i f^t(x_i)))^2$	$w_i^2 \left\  y^{c_i} - \sum_{k \neq c_i} y^k \epsilon_k(x_i, c_i) \right\ ^2$
$w_i$	N/A	$\sum_{k \neq c_i} \phi' \left[ \frac{1}{2} \langle f^t(x_i), y^{c_i} - y^k \rangle \right]$
$\epsilon_k(x, c)$	N/A	$\frac{\phi' \left[ \frac{1}{2} \langle f^t(x), y^c - y^k \rangle \right]}{\sum_{k \neq c} \phi' \left[ \frac{1}{2} \langle f^t(x), y^c - y^k \rangle \right]}$
$\phi(v)$	N/A	$e^{-v}$
$f^*(\mathcal{L}^t)$	$\arg \min_f \sum_{(x_i, y_i) \in \mathcal{L}^t} \phi(y_i f(x_i))$	$\arg \min_f \sum_{(x_i, y_i) \in \mathcal{L}^t} \sum_{l=1, l \neq y_i}^C \phi \left( \frac{1}{2} \langle y^{y_i} - y^l, f(x_i) \rangle \right)$

and (2.19) follows from (2.9).

The following result uses this property to show that, given what the optimal student has learned until iteration  $t$ , the derivative of the population risk is independent of the teaching set  $\mathcal{L}^{t-1}$  already studied.

**Lemma 2** *Consider the iterative machine teaching procedure of Definition 3. Then, the predictor  $f^t$  learned by the optimal student of Definition 1 at iteration  $t$  is such that, for any direction  $g$  in  $Sp(\Psi(\mathcal{D}))$*

$$\partial_g R_{\mathcal{D}}(f^t) = \partial_g R_{\mathcal{D}^{t-1}}(f^t). \quad (2.24)$$

**Proof** Assume, without loss of generality, that  $\mathcal{L}^{t-1}$  contains examples  $\{x_i\}_{i=1}^k$  and  $\mathcal{D}^{t-1}$  examples  $\{x_i\}_{i=k+1}^n$ , for some  $1 < k < n$ . Then

$$\nabla_{\Psi(\mathcal{D})}^T R_{\mathcal{L}^{t-1}}(f^t) = \left( \nabla_{\Psi(\mathcal{L}^{t-1})}^T R_{\mathcal{L}^{t-1}}(f^t), \nabla_{\Psi(\mathcal{D}^{t-1})}^T R_{\mathcal{L}^{t-1}}(f^t) \right) \quad (2.25)$$

$$= \left( \nabla_{\Psi(\mathcal{L}^{t-1})}^T R_{\mathcal{L}^{t-1}}(f^t), 0 \right). \quad (2.26)$$

Since the student is optimal, (2.6) holds and, using (2.13),  $\nabla_{\Psi(\mathcal{L}^{t-1})} R_{\mathcal{L}^{t-1}}(f^t) = 0$ . Hence,  $\nabla_{\Psi(\mathcal{D})} R_{\mathcal{L}^{t-1}}(f^t) = 0$  and, from (2.9),  $\partial_g R_{\mathcal{L}^{t-1}}(f^t) = 0$ . Since, from Lemma 1,

$$\partial_g R_{\mathcal{D}}(f^t) = \partial_g R_{\mathcal{L}^{t-1}}(f^t) + \partial_g R_{\mathcal{D}^{t-1}}(f^t), \quad (2.27)$$

(2.24) follows.

The following theorem uses these results to derive the example selection strategy of the optimal teacher.

**Theorem 1** *Consider the iterative machine teaching procedure of Definition 3, with optimal student as in Definition 1, and set of permissible choices of (2.15). The optimal teacher of Definition 4 selects the teaching set  $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \mathcal{N}^t$  with novel examples*

$$\mathcal{N}^t = \arg \max_{\mathcal{N} \in \mathcal{P}^t} \|\nabla_{\Psi(\mathcal{N})}^T R_{\mathcal{N}}(f^t)\|^2 \quad (2.28)$$

$$= \arg \max_{\mathcal{N} \in \mathcal{P}^t} \sum_{(x_i, y_i) \in \mathcal{N}} w_i^2 \quad (2.29)$$

where  $w_i = \phi'(y_i f^t(x_i))$ .

**Proof** For any  $g = \sum_{x_i \in \mathcal{D}} \alpha_i \delta(x - x_i)$ ,  $\|g\| = 1$  if and only if  $\|\alpha\| = 1$  and, from (2.9),

$$\partial_g R_{\mathcal{D}}(f^t) = \left\langle \nabla_{\Psi(\mathcal{D})} R_{\mathcal{D}}(f^t), \alpha \right\rangle \geq -\|\nabla_{\Psi(\mathcal{D})} R_{\mathcal{D}}(f^t)\| \|\alpha\| = -\|\nabla_{\Psi(\mathcal{D})} R_{\mathcal{D}}(f^t)\|. \quad (2.30)$$

Since equality is achieved when  $\alpha$  is the direction

$$\alpha^* = -\frac{1}{\|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{D}}(f^t)\|}\nabla_{\Psi(\mathcal{D})}R_{\mathcal{D}}(f^t), \quad (2.31)$$

the steepest descent solution of (2.16) is

$$g^* = \sum_{x_i \in \mathcal{D}} \alpha_i^* \delta(x - x_i) \quad (2.32)$$

Similarly, the steepest descent direction of (2.18) is

$$h^*(\mathcal{L}) = \sum_{x_i \in \mathcal{L}} \nu_i^* \delta(x - x_i) \quad (2.33)$$

with

$$\nu^* = -\frac{1}{\|\nabla_{\Psi(\mathcal{L})}R_{\mathcal{L}}(f^t)\|}\nabla_{\Psi(\mathcal{L})}R_{\mathcal{L}}(f^t), \quad (2.34)$$

Assuming, without loss of generality, that  $\exists k$  such that  $x_i \in \mathcal{L}$  for  $i < k$ , then

$$h^*(\mathcal{L}) = \sum_{x_i \in \mathcal{D}} \beta_i^* \delta(x - x_i) \quad (2.35)$$

where

$$(\beta^*)^T = (\nu^T, 0) = -\frac{1}{\|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{L}}(f^t)\|}\nabla_{\Psi(\mathcal{D})}^T R_{\mathcal{L}}(f^t), \quad (2.36)$$

and

$$\langle g^*, h^*(\mathcal{L}) \rangle = \langle \alpha^*, \beta^* \rangle \quad (2.37)$$

$$= \left\langle -\frac{1}{\|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{D}}(f^t)\|} \nabla_{\Psi(\mathcal{D})}R_{\mathcal{D}}(f^t), -\frac{1}{\|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{L}}(f^t)\|} \nabla_{\Psi(\mathcal{D})}R_{\mathcal{L}}(f^t) \right\rangle \quad (2.38)$$

$$= \frac{\|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{L}}(f^t)\|^2}{\|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{D}}(f^t)\| \|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{L}}(f^t)\|} \quad (2.39)$$

$$= \frac{\|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{L}}(f^t)\|}{\|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{D}}(f^t)\|}, \quad (2.40)$$

where we have used the fact that

$$\nabla_{\Psi(\mathcal{D})}^T R_{\mathcal{D}}(f^t) = \left( \nabla_{\Psi(\mathcal{D})}^T R_{\mathcal{L}}(f^t), \nabla_{\Psi(\mathcal{D})}^T R_{\mathcal{D}-\mathcal{L}}(f^t) \right). \quad (2.41)$$

It follows that the solution of (2.17) is

$$\mathcal{N}^t = \arg \max_{\mathcal{N} \in \mathcal{P}^t} \|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{L}^{t-1} \cup \mathcal{N}}(f^t)\|^2. \quad (2.42)$$

$$= \arg \max_{\mathcal{N} \in \mathcal{P}^t} \left\{ \|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{L}^{t-1}}(f^t)\|^2 + \|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{N}}(f^t)\|^2 \right\} \quad (2.43)$$

$$= \arg \max_{\mathcal{N} \in \mathcal{P}^t} \|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{N}}(f^t)\|^2 \quad (2.44)$$

$$= \arg \max_{\mathcal{N} \in \mathcal{P}^t} \|\nabla_{\Psi(\mathcal{N})}R_{\mathcal{N}}(f^t)\|^2 \quad (2.45)$$

where we have used the fact that, from Lemma 2,  $\|\nabla_{\Psi(\mathcal{D})}R_{\mathcal{L}^{t-1}}(f^t)\|^2 = 0$ .

The theorem shows that the optimal teacher strategy is to select the set of novel examples  $\mathcal{N}$  available in  $\mathcal{P}^t$  of *largest* risk gradient. For this reason, we denote the teacher as the *MaxGrad* teacher. Since, for margin losses,  $\phi'$  has largest magnitude for negative arguments,  $w_i$  is largest for examples of *negative* margin, i.e. which are incorrectly classified by the current student predictor  $f^t$ . Hence,  $w_i$  is a measure of how difficult each example



is, under the current state of student knowledge. Similarly measures the difficulty, for the student, of the novel examples in  $\mathcal{N}$ . It follows from (2.29) that the MaxGrad teacher always selects the *hardest set of novel examples* in  $\mathcal{P}^t$ . Furthermore, since  $\mathcal{H}(\mathcal{N})$  is a sum of non-negative terms, it is an increasing function of  $|\mathcal{N}|$ . This implies that the teacher has a preference for larger sets of novel examples. As long as there are examples that the student has not mastered ( $w_i > 0$ ), it will choose a set of  $\tau$  examples per iteration. Hence,  $|\mathcal{N}^t| = \tau$  for all  $t < T$  and the overall learning complexity is  $T\tau$ . This implies that the number of iterations is upper bounded by  $\zeta/\tau$ , which makes it equivalent to specifying a maximum level of effort  $\zeta$  or a maximum number of iterations  $T$  for the teaching process. Finally, because the set of permissible choices includes all novel sets of cardinality  $\tau$ , the solution of (2.29) is trivial: it suffices to compute  $w_i$  for all examples in  $\mathcal{D}^{t-1}$  and select the  $\tau$  examples of largest  $w_i^2$ . The resulting machine teaching procedure is summarized by Algorithm 1.

### 2.3.5 Multi-class Extension

We have discussed binary classification tasks, where  $f(x) \in \mathbb{R}$ , class labels  $y \in \{-1, 1\}$ , the margin of example  $(x, y)$  is defined as  $yf(x)$  and a margin loss is a function  $\phi(yf(x))$  for some decreasing  $\phi \in \mathbb{R}^+$ . All ideas can be generalized for the  $C$ -class case, by extending these definitions. A common generalization is to use a  $d$ -dimensional predictor,  $f(x) \in \mathbb{R}^d$ , a set of  $C$  class label codewords  $y^c \in \mathcal{Y} = \{y^1, \dots, y^C\}$ , where  $y^c \in \mathbb{R}^d$ , and define the margin of example  $x$  with respect to class  $y^k$  as

$$\mathcal{M}(y^k, f(x)) = \min_{l \neq k} \frac{1}{2} \langle y^k - y^l, f(x) \rangle. \quad (2.46)$$

A family of margin losses is then defined as [67]

$$L[y^k, f(x)] = \sum_{l=1, l \neq k}^C \phi\left(\frac{1}{2} \langle y^k - y^l, f(x) \rangle\right), \quad (2.47)$$

where  $\phi: \mathbb{R} \rightarrow \mathcal{R}^+$  are strictly positive. A theoretical discussion of the properties of these losses can be found in [67]. The empirical risk then becomes

$$R_{\mathcal{D}}(f) = \sum_{(x_i, y_i) \in \mathcal{D}} L[y^{y_i}, f(x_i)]. \quad (2.48)$$

and, given a dataset  $\mathcal{D} = \{(x_i, c_i)\}$  and a corresponding set of directions  $\Psi(\mathcal{D}) = \{\psi_1, \dots, \psi_n\}$  such that  $\psi_i = \delta(x - x_i)$  the gradient of  $R_{\mathcal{D}}(f)$  evaluated at  $f^t$  with respect to  $\Psi(\mathcal{D})$  has entries

$$[\nabla_{\Psi(\mathcal{D})} R_{\mathcal{D}}(f^t)]_i = w_i \left( y^{c_i} - \sum_{k \neq c_i} y^k \epsilon_k(x_i, c_i) \right), \quad (2.49)$$

with

$$w_i = \sum_{k \neq c_i} \phi' \left[ \frac{1}{2} \langle f^t(x_i), y^{c_i} - y^k \rangle \right] \quad (2.50)$$

$$\epsilon_k(x, c) = \frac{\phi' \left[ \frac{1}{2} \langle f^t(x), y^c - y^k \rangle \right]}{\sum_{k \neq c} \phi' \left[ \frac{1}{2} \langle f^t(x), y^c - y^k \rangle \right]}. \quad (2.51)$$

Note that  $[\nabla_{\Psi(\mathcal{D})} R_{\mathcal{D}}(f^t)]_i$  is a d-dimensional vector. The gradient norm of (2.29) is then

$$\|\nabla_{\Psi(\mathcal{N})}^T R_{\mathcal{N}}(f^t)\|^2 = \sum_{(x_i, c_i) \in \mathcal{N}} \left\| [\nabla_{\Psi(\mathcal{D})} R_{\mathcal{D}}(f^t)]_i \right\|^2 \quad (2.52)$$

$$= \sum_{(x_i, c_i) \in \mathcal{N}} \xi(x_i) \quad (2.53)$$

where  $\xi(x_i) = w_i^2 \|y^{c_i} - \sum_{k \neq c_i} y^k \epsilon_k(x_i, c_i)\|^2$ . In this work, we adopt the exponential loss by setting  $\phi(v) = e^{-v}$ , leading to the multi-class version of Algorithm 1 for the implementation

of the optimal multi-class teacher.

### 2.3.6 Connections to Boosting

The algorithm above has certain similarities with boosting. Note that the weights of (2.11) are the weights of boosting at the end of the iteration that produces  $f$  as strong classifier. Boosting then selects the weak learner  $g^*$  that maximizes (2.13), adds this to  $f$  to produce the new strong classifier and iterates. Since examples of large weight are those worse classified by  $f$ , the algorithm focuses on the hardest examples (for the currently learned classifier) to pick the next weak learner. The MaxGrad teacher does essentially the same. In this case,  $f^t$  is the predictor currently learned by the optimal student, and the teacher selects the hardest examples for the student. However, in boosting, this is used to perform one learning iteration and select one weak learner. In machine teaching, the student is assumed to be able to fully learn  $\mathcal{L}^t$ , i.e. does not simply perform a gradient iteration on  $R_{\mathcal{D}}(f^t)$  but actually solves (2.4). If, for example, the student is a machine learning algorithm, this can be done by implementing the complete boosting algorithm on  $R_{\mathcal{L}^t}(f)$ . While boosting uses the entire example population  $\mathcal{D}$  to perform a boosting iteration and select a single weak learner, the machine teaching algorithm selects the best set  $\mathcal{N}^t$  of  $\tau$  novel examples to add to  $\mathcal{L}^t$  and performs any number of boosting iterations needed to solve the new teaching set  $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \mathcal{N}^t$ . In summary, while boosting assumes a weak learner with access to the entire dataset  $\mathcal{D}$ , the machine teaching algorithm assumes a strong learner with access to the limited information available in  $\mathcal{L}^t$ .

## 2.4 Experiments

**Dataset:** MaxGrad was evaluated on two datasets, Butterflies and Chinese Characters illustrated in Figure 2.3. Butterflies [59] is a fine-grained multi-class dataset of



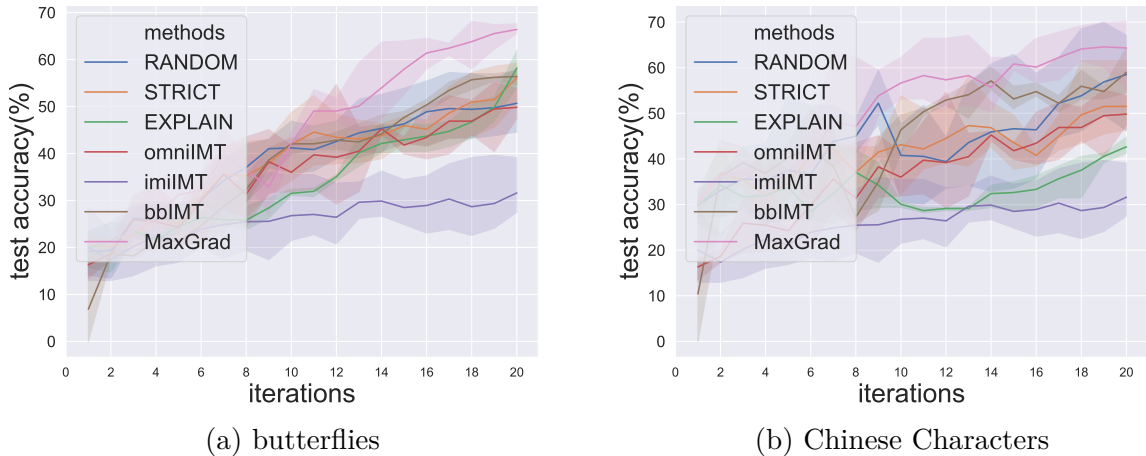
**Figure 2.3:** Example images from our two datasets

images of five butterfly species, captured in a large variety of settings, from the iNaturalist dataset [85]. It is a challenging dataset due to the large intra-class image diversity, low image resolution, and high similarity of some of the species. Chinese Characters [86, 61] consists of three similar Chinese characters: Grass, Mound, and Stem. The images vary in difficulty, due to a large variety of handwriting styles and image qualities. We use the training-testing split of [59] on both cases. The data is accessible in [87]. The teaching set is selected from the training set.

**Implementation details:** Both datasets were subject to standard normalizations. The pre-trained ResNet-18 [1] on ImageNet is used to simulate the student. This is equivalent to assuming a student that starts from a good generic understanding of image classification. The student learners are trained 10 epochs by gradient descent with batch size equal to  $|\mathcal{L}^t|$  and weight decay of  $1e-4$ . The learning rate is set to  $1e-4$  with 0.9 momentum. For fair comparison with other methods [56, 55, 59, 57], novel sets of size  $\tau = 1$  were used in all experiments, i.e. a single example is selected per iteration.

### 2.4.1 Evaluation with Simulated Learners

We start with evaluations on simulated learners, i.e. a classifier. This enables a simple evaluation setting and fully reproducible experiments. Figure 2.4 shows the accuracies of student networks taught with examples selected randomly (RANDOM), by STRICT [55], EXPLAIN [59], omniscient teacher (omniIMT) [56], imitation teacher



**Figure 2.4:** Test set accuracy of simulated students as a function of teaching iterations (teaching example number).

(imiIMT) [56], black-box IMT (bbIMT) [57], and MaxGrad. While student performance improves with teaching set size for all methods, MaxGrad has the fastest growth and the best performance for all iterations. The gains are significant: in butterflies and characters it achieves an accuracy at 15 iterations that others do not reach before 20 iterations. Of all algorithms, it is also the only to stably outperform RANDOM.

## 2.4.2 Evaluation with Real Learners

We next tested the algorithm on MTurk users. Note that a student network was still used to assemble the teaching set, which was then used to train MTurkers. In this case, the student network was trained without any stochasticity. We used gradient descent and gave up data augmentation techniques (e.g. random crops or flips) that are not accessible to the human students. The codewords  $y^c$  of Algorithm 1 were initialized with the canonical basis and refined during the student optimization.

The MTurk experiments followed the setting of [59, 55], using 40 workers per dataset. The teaching process consists of two phases, teaching and testing. Before teaching, workers

**Table 2.1:** Test set accuracies for MTurk learners. Methods with superscript “\*” represent our implementations. Values are presented by mean(std).

	Butterflies	Chinese Char.
RANDOM [59]	65.20	47.05
STRICT [55]	65.00	51.51
EXPLAIN [59]	68.33	65.44
omniIMT* [56]	70.07(18.30)	64.36(19.58)
imiIMT* [56]	72.70(17.63)	64.46(23.72)
bbIMT* [57]	76.09(18.05)	64.37(19.57)
RANDOM*	63.15(18.17)	51.53(24.47)
MaxGrad	<b>80.33</b> (19.76)	<b>81.89</b> (12.93)

were shown a brief introduction to the teaching task. In the teaching stage, they were shown a sequence of 20 images. At each iteration, they were asked to select a category from a list of candidate options, and received feedback declaring their choice ‘Correct’ or ‘Incorrect,’ as well as the true class. Upon this, learners had to wait for a minimum of 2 seconds before proceeding to the next iteration. After teaching, 20 randomly selected test images were assigned to each learner, who was asked to classify them. These random images were different per learner and no feedback was provided as they were classified.

Table 2.1 reports the accuracy of image classification by the students on the test set. The results shown in the top third of the table (RANDOM, STRICT and EXPLAIN) are taken from [59]. For completeness, we repeated the experiments with random image selection, which produced similar results, as shown in the bottom third. The remaining three results of previous methods (center third of the table) are obtained with our own implementation. MaxGrad significantly outperforms the previous approaches, achieving gains of almost 5 (17) points on Butterflies (Chinese characters). Finally, we observe that human test accuracy is higher than that of the simulated student used to collect the training set, shown in Figure 2.4. This confirms that the optimal student assumption is realistic for human learners.

## 2.5 Conclusion

In this work, we have proposed MaxGrad, a new gradient-based machine teaching algorithm derived from the optimal student assumption. We have demonstrated its effectiveness on both synthetic and human student teaching experiments. While we have not considered the integration of teaching and explanations yet, MaxGrad can be generalized to accommodate the latter. For example, explanations can be merged with classifier training using attention mechanisms. This will be discussed in Chapter 4.

Chapter 2 is, in full, based on the material as they appear in the publication of “Gradient-Based Algorithms for Machine Teaching”, Pei Wang, Kabir Nagrecha, Nuno Vasconcelos, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2021. The dissertation author was the primary investigator and author of this paper.

# Chapter 3

## A Generalized Explanation

## Framework for Visualization of Deep Learning Model Predictions



## 3.1 Introduction

While deep learning systems enabled significant advances in computer vision, their black-box nature creates difficulties for many applications. In general, it is difficult to *trust* a system that cannot justify its decisions. This motivated a large literature on explainable AI (XAI) methods, which complement network predictions with human-understandable explanations [88, 89, 90, 91, 92, 93, 94, 95, 96, 97]. In computer vision, the dominant XAI paradigm is that of visual explanations computed by *attribution* functions, which generate heatmaps localizing the image pixels [98, 99, 95, 100] or regions [101, 102, 103, 104] responsible for network predictions. Figure 3.1 (center) shows the heatmap produced for a bird image by a deep learning system that predicts the label ‘Cardinal’ with confidence value 0.76.

While attributive explanations provide a *coarse* justification for the predictions, e.g. localizing the object within a larger background or highlighting one among distinct objects in the field of view, they are not sufficient for applications that require fine-grained classification. This can be seen in Figure 3.1, where it is clear that the highlighted pixels belong to the bird but unclear which regions of the bird are responsible for the ‘Cardinal’ prediction. While the explanation would be satisfactory for a classification problem opposing ‘Birds’ to ‘Dogs’, it is not helpful for one opposing ‘Cardinals’ to ‘Summer Tanagers’ or other bird species. In this case, the attributive explanation selects the entire bird and it is hard to know what differentiates one class from the other.

Fine-grained classification problems are prevalent in expert domains, such as medical imaging or biology, where there is a need to distinguish objects that differ in subtle details, and even for everyday applications that involve a large number of classes. For such problems, users are likely to demand more from the explanation system. As Figure 3.1 illustrates, given the relatively low confidence value of 0.76, a user may want to know exactly why the system chose the ‘Cardinal’ label. Beyond the post-hoc analysis of classification results,

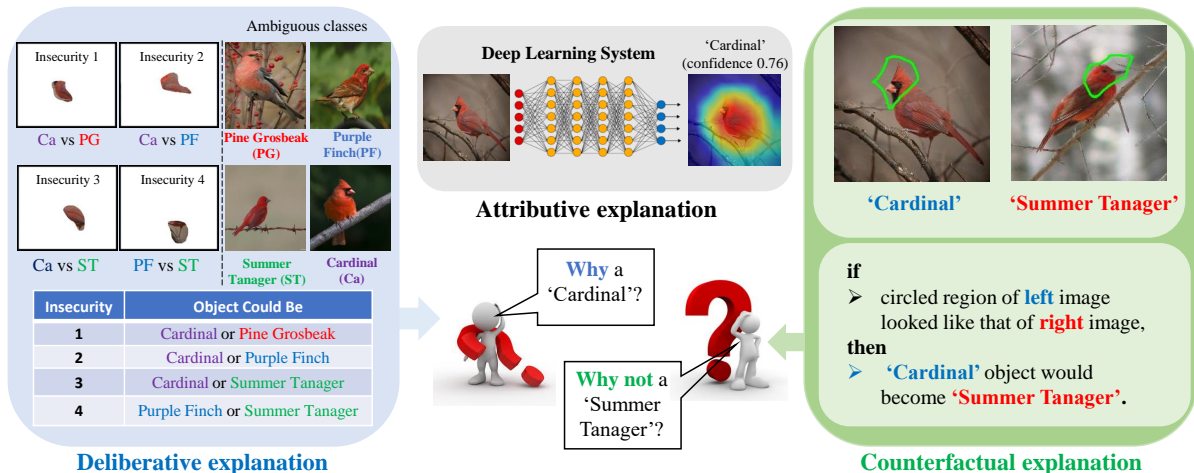


Figure 3.1: An ideal explainable deep learning system.

where the user is passive, explanations also play a critical role in interactive applications, such as machine teaching systems where users are taught to annotate images [105, 51, 59]. In this case, users naturally ask counterfactual questions, such as “why is this a Cardinal and not a Summer Tanager?” where an alternative or counter-class (‘Summer Tanager’) is provided. None of these questions can be satisfied by existing attribution-based visual explanations.

In this work, we propose a *Generalized explanation framework* (GALORE) for the solution of all these problems. The proposed framework includes a new class of explanations, denoted as *deliberative*, which address the “why?” question of the left of Figure 3.1, and unifies them with the attributive explanations at the top center of the figure, and *counterfactual explanations*<sup>1</sup> that address the “why not?” question on the right side of the Figure. The unification is based on the definition of all explanations as combinations of multiple attribution maps, which vary according to the explanation type. Since attributions are very efficient to compute, the proposed framework establishes a

<sup>1</sup>As discussed in [106] and defined in [107, 108], counterfactual explanations are similar to contrastive explanations. Both aim to answer questions “Why P and not Q”, although some literature emphasizes that counterfactual explanations should generate alternative examples, illustrating how objects change for the alternative decision [109, 110]. We make no distinction and use the two terms interchangeably.

family of low-complexity explanations that can be used in various applications, ranging from naive to expert domains, and supporting both passive post-hoc analysis of predictions or interactive applications such as machine teaching.

A core requirement of deliberative and counterfactual explanations is the ability to reason in terms of the *difficulty* posed to the classification by different image regions. Understanding why the classifier chose a class requires knowing what other classes could have been plausibly selected, and what image regions made those alternatives plausible, i.e. what image regions the classifier found ambiguous for the decision. This is the essence of deliberative explanations, which produce a list of such regions, denoted as *insecurities*, as illustrated in the left of Figure 3.1. On the other hand, counterfactual explanations require the identification of regions that discriminate the predicted from the counterfactual class, i.e. which have high probability under the predicted class and low probability under the counterfactual. These regions can then be shown to the user, as illustrated in right of Figure 3.1, to identify corresponding parts in objects from predicted and counter class.

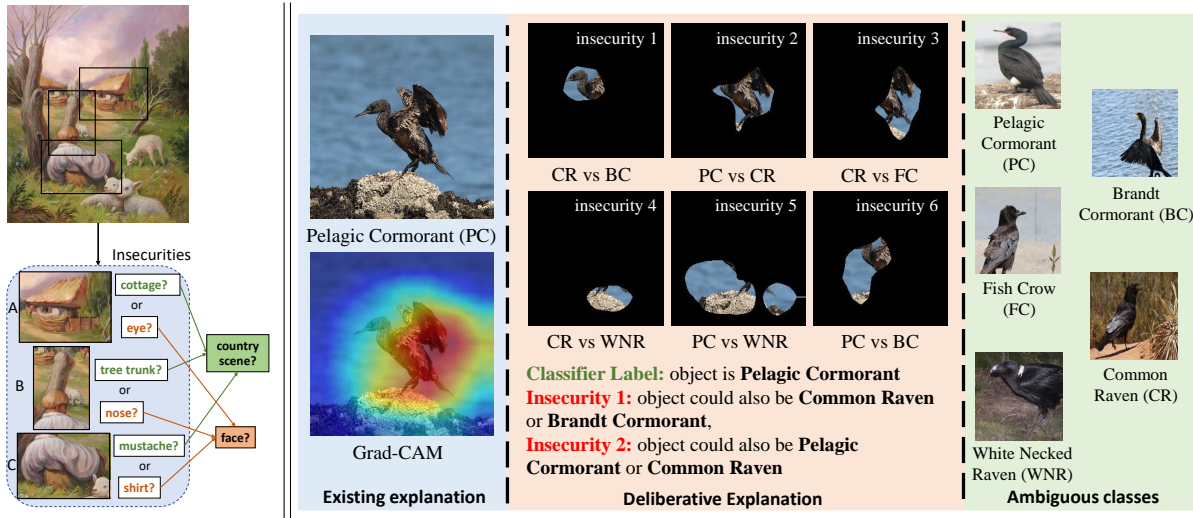
Reasoning about ambiguities or class probabilities requires the classifier to produce confidence scores [111, 112, 113, 114], i.e. measure the confidence with which the image belongs to each of the possible classes. From these scores, it is possible to derive how difficult the classification is (the probability of the ground-truth class), how ambiguous it is (similarity between the probabilities of the top classes), or how much the image discriminates between two classes (large probability for one and small for another). We refer to the ability to measure these quantities as *self-awareness*, since it allows a classifier to quantify the confidence in its decisions. One of the insights of this work is that attributions of confidence scores allow the extension of these measures to image regions, so as to identify which regions are ambiguous, discriminant, or difficult to classify. This is naturally integrated in the GALORE framework, by simply combining the attribution maps for self-awareness with the attributions for class predictions required to compute the different explanations.

Beyond explanations, a significant challenge to XAI is the lack of explanation ground truth for performance evaluation. Besides user-based evaluations [115], whose results are difficult to replicate, we propose a quantitative metric based on a proxy localization task. This relies on standard metrics from the object detection literature and attribute annotations for different object parts or scene components. We show that these metrics can be adapted to the evaluation of the different types of explanations proposed with minor specializations. Compared to human experiments, the proposed proxy evaluation has the advantages of being substantially easier to perform and fully replicable.

Overall, this work makes six contributions. First, it introduces a new family of deliberative explanations, which visualize the deliberations made by the network to reach its predictions. Second, it introduces a new definition of counterfactual explanations as combinations of attributive explanations, making them more efficient to compute. Third, it proposes the unified GALORE framework to generate attributive, deliberative, and counterfactual explanations. Fourth, it shows how to leverage self-awareness to improve explanation accuracy, for different types of explanations. Fifth, it proposes a new experimental protocol for quantitative evaluation of deliberative and counterfactual explanations. Sixth, experimental results, using both this protocol and human experiments, show that the proposed deliberative explanations are intuitive, suggesting that the deliberative process of modern networks correlates with human reasoning, and that counterfactual explanations can substantially benefit applications like machine teaching.

## 3.2 Related Work

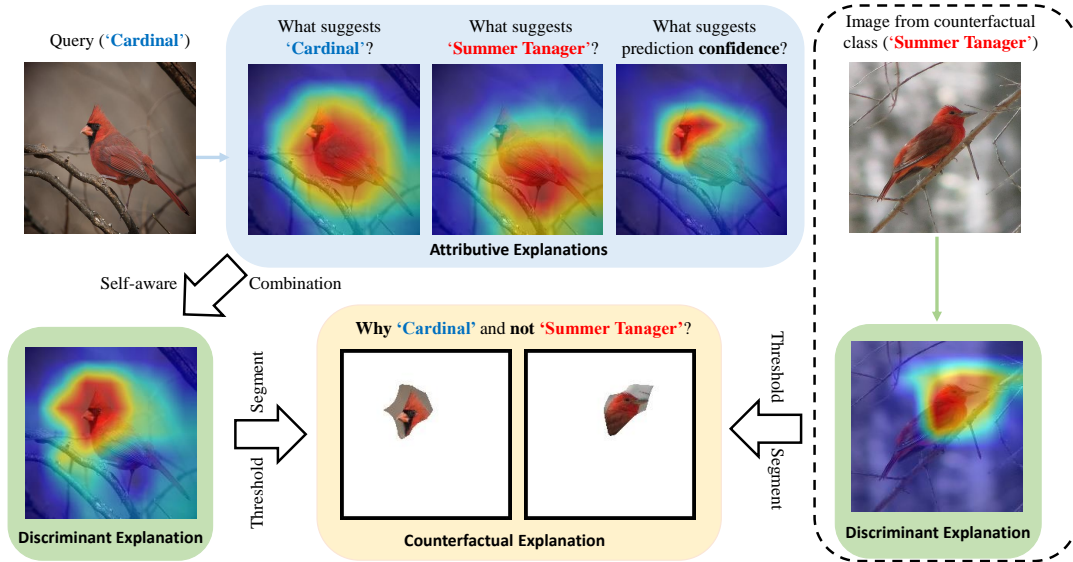
**XAI for computer vision:** Many variants of XAI have been proposed in the literature. For computer vision, explanations can be based on concepts [116, 117, 118], examples [88, 119, 120, 121], image transformations [122, 115], language [123, 124, 125], etc.



**Figure 3.2:** Left: Illustration of the deliberations made by a human to categorize an ambiguous image. Insecurities are ambiguous regions. Right: Deliberative explanations expose this deliberative process.

Among these, the visualization of saliency maps is a widely used approach [126, 127, 128, 104, 103], which we pursue in this work. XAI methods can also be divided into two groups that depend on the design stage where predictions and explanations are performed. One possibility is to design models to be interpretable [128, 89, 129, 130], another to perform post-hoc analysis on pre-trained models [131, 104, 103]. In this work, we mainly discuss post-hoc methods. Several survey papers [132, 133, 134, 135] provide a more comprehensive review of the field.

**Attributive explanations:** The most popular post-hoc XAI approach to create saliency maps is to rely on attribution functions [99, 95, 136, 101, 104]. These methods produce a heatmap that encodes how much the classifier prediction can be attributed to each pixel or image region. Many attribution functions have been proposed [98, 99, 95, 136, 100, 104, 137]. The most popular approach is to compute some variants of the gradient of the classifier prediction with respect to a chosen layer of the network and then backproject to the input [103, 102]. These techniques tend to work well when the object of the predicted class is immersed in a large background (as in object detection or scene recognition), but



**Figure 3.3:** The derivation of a counterfactual explanation.

are less useful when the image contains the object alone (as in object recognition). In this setting, the heat map frequently covers the whole object, as illustrated in Figure 3.1. This is troublesome since the recognition setting is the best suited for expert domains, where classification requires inspection of image details and discrimination between fine-grained classes. We show that more informative explanations can be obtained with deliberative explanations and counterfactual explanations. In any case, our goal is not to propose a new attribution function, but to introduce a new explanation strategy, deliberative explanations, that visualize network insecurities about the prediction, and a new unified explanation framework, GALORE, that can produce fast deliberative and counterfactual explanations. This framework can be combined with any of the visualization approaches above, and leverages an additional attribution function for confidence scores.

**Contrastive and counterfactual explanations:** Counterfactual explanations have a long history in machine learning [138, 139] and have been extensively studied for tabular data [140, 123, 141]. In computer vision, they have only received attention in the recent past [115, 142, 143]. Two main approaches have emerged. Natural language (NL)

methods attempt to produce a textual explanation understandable by humans [142, 125, 110]. Since image to text translation is still a difficult problem, full blown NL explanations tend to target specific applications, like self driving [144]. More robust systems tend to use a limited vocabulary, e.g. a set of image attributes [125, 142]. Beyond the a priori definition of a vocabulary (e.g. attributes), these methods require training data for each vocabulary term, and training of the classifier to produce this side information. To avoid these difficulties, most explanation methods rely instead on visualizations.

While the ideas proposed in this work could be extended to NL, we consider only visual explanations. In this area, counterfactual explanations transform an image of class  $A$  so as to elicit its classification into the counter class  $B$  [145, 123, 146, 147, 148, 149]. The simplest example are adversarial attacks [108, 145], which optimize perturbations to map an image of class  $A$  into class  $B$ . However, these perturbations usually push the perturbed image outside the boundaries of the space of natural images. Generative methods have been proposed to address this problem, computing large perturbations that generate realistic images [146, 150, 151, 152]. This is guaranteed by the introduction of regularization constraints, auto-encoders, or GANs [153]. However, because realistic images are difficult to synthesize, these approaches have only been applied to simple, MNIST or CelebA [154] style, datasets and domains that do not require expertise [150, 152, 122]. StyleEx [148] is a recent example, leveraging a GAN to produce the explanations. This, however, requires training on large-scale data, which is not a necessity for other methods. A more plausible alternative is to exhaustively search the space of features extracted from a large collection of images, to find replacement features that map the image from class  $A$  to  $B$  [115]. While this has been shown to perform well on fine-grained datasets, exhaustive search is too complex for interactive applications.

**XAI Evaluation:** Explanations are frequently evaluated through human-in-the-loop experiments that measure their consistency with human intuition [155, 103, 108, 156] or

evaluate if explanations improve user performance on some task [115]. It is also possible to assemble a dataset to generate human-driven ground-truth explanations [157]. An alternative approach is automated evaluation, using a proxy task without human participation. A typical example is to erase or add features and observe how the model predictions change [158, 159, 160, 161]. Another is localization, where regions of features deemed important by the explanation are compared to regions deemed intuitive for classification by humans [103, 162]. Another component of the evaluation of explanations is to test their robustness via sanity checks [163, 164, 165, 166]. In this work, we introduce a quantitative protocol for the evaluation of both deliberative and counterfactual visual explanations, which includes sanity checks.

**Self-awareness:** Self-aware systems have some ability to measure their limitations or predict failures. This includes out-of-distribution detection [167, 168, 169, 170] or open set recognition [171, 172, 173, 174], where classifiers are trained to reject non-sensical images, adversarial attacks, or images from classes on which they were not trained. All these problems require the classifier to produce a confidence score for image rejection. The most popular solution is to guarantee that the posterior class distribution is uniform, or has high entropy, outside the space covered by training images [175, 176]. This, however, is not sufficient for deliberative explanations, which have to precisely characterize the ambiguity of image regions, or counterfactual explanations, which require precise confidence scores for classes  $A$  and  $B$ . These explanations are more closely related to realistic classification [177], where a classifier must identify and reject examples that it deems too difficult to classify.

### 3.3 A Unified View of Explainable AI

In this section, we discuss the different types of explanations implemented by the proposed GALORE framework. The detailed computations required to produce the



explanations are discussed in Section 3.4.

### 3.3.1 Attributive Explanations

Attributive explanations identify pixels responsible for a classifier prediction. This is intuitive but prone to generate explanations that are too generic. For example, when asked “why is an object a truck?” an attributive system would answer “because it has wheels, a hood, seats, a steering wheel, a flatbed, head and tail lights, and rearview mirrors,” i.e. generate a list of all the truck parts. After all, all parts are responsible for the ‘truck’ label. The problem is that, while insightful, the explanation does not inform on what distinguishes the truck from, for example, a car. The explanation for ‘car’ would share all components other than the flatbed.

Similarly, visual attributive explanations tend to highlight all pixels of objects in the predicted class. This is sensible for coarse grained classification, e.g. ‘birds’ vs ‘cats,’ but not for fine-grained, e.g. the CUB birds dataset [178] from which the images of Figures 3.1, 3.2 and 3.3 were taken. On this dataset, where most images contain a single bird, methods like Grad-CAM [103] (used in these examples) produce heatmaps that 1) cover most of the bird, and 2) vary little across classes of largest posterior probabilities, leading to very uninformative explanations. In this work, we seek better explanations for the fine-grained setting.

### 3.3.2 Deliberative Explanations

In this setting, visual concepts differ in subtle ways. There are frequently two or more classes of very similar appearance, and the classification can be quite ambiguous. This is illustrated in both Figures 3.1 and 3.2, which present several similar birds, difficult to differentiate for a layperson. Due to this ambiguity, even an expert could reasonably oscillate between different interpretations while deliberating about the class to predict. An

extreme example of this process can be observed for visual illusions, such as that depicted in the left of Figure 3.2, where different image regions provide support for conflicting image interpretations. In this example, the image could depict a ‘country scene’ or a ‘face.’ Most humans would consider the two interpretations while deliberating on a final prediction. When asked to explain the latter, they would say something like: “I see a cottage in region A, but region B could be a tree trunk or a nose, and region C looks like a mustache, but could also be a shirt. Since there are sheep in the background, I am going with country scene.” More generally, different regions can provide evidence for two or more distinct predictions and there may be a need to deliberate between multiple classes.

Having access to this deliberative process is important to trust an AI system. For example, in medical diagnosis, a single prediction can appear unintuitive to a doctor, even if accompanied by a heatmap. The doctor’s natural reaction would be to ask “why did you reach that conclusion?” Ideally, instead of simply outputting a predicted label and a heatmap, the AI system should visualize its *deliberations*, producing a list of image regions that support other plausible predictions. We denote these regions as *insecurities*, since they cast doubt on the validity of the predicted label. To accomplish this, we propose a new type of explanations based on heatmaps of network insecurities. These are denoted as *deliberative explanations*, since they visualize the deliberative process of the network.

As illustrated in the right of Figure 3.2, the deliberative explanation provides a list of insecurities (center inset), each consisting of 1) an image region and 2) an *ambiguity*, formed by the pair of classes that led the network to be uncertain about the region. Example images from the ambiguous classes can also be displayed, as shown in the right inset. For example, the first insecurity of Figure 3.2 reflects the fact that the head of the Pelagic Cormorant is similar to those of the Brandt Cormorant and the Common Raven. Hence, this region raises uncertainty about the ‘Pelagic Cormorant’ label predicted by the classifier.

### 3.3.3 Counterfactual Explanations

Returning to the ‘truck’ example, domain experts will likely not be satisfied by the simply listing of all truck parts. Instead, they are likely to request more *precise* explanations, for instance asking the question “Why is it a truck and not a car?” The answer “because it has a flatbed. If it did not have a flatbed it would be a car,” is known as a *counterfactual explanation* [123, 108, 179, 115]. Counterfactual explanations, by supporting a specific query with respect to a *counterfactual* class ( $B$ ), allow expert users to zero-in on a specific ambiguity between two classes, which they already *know* to be plausible predictions. Unlike attributions, these explanations scale naturally with user expertise. As the latter increases, the class and counterfactual class simply become more *fine-grained*. In computer vision, counterfactual explanations are usually implemented as “correct class is  $A$ . Class  $B$  would require changing the image as follows,” where “as follows” is some visual transformation. Possible transformations include image perturbations akin to those used in adversarial attacks [108], image synthesis [122, 149], or replacing image regions by regions of some images in the counter class  $B$ , found by the exhaustive search of a large feature pool [115]. However, image perturbations and synthesis frequently leave the space of natural images, only working on simple non-expert domains, and feature search is too complex for interactive applications.

In this work, we propose the computation of counterfactual explanations by a simple and robust procedure, based on attributions. We start by introducing *discriminant explanations* that, as shown in Figure 3.3, connect attributive to counterfactual explanations. Like attributive explanations, they consist of a single heatmap. This, however, is an attribution map for the *discrimination* of classes  $A$  and  $B$ , attributing high scores to image regions that are informative of  $A$  but not of  $B$ , and high classification confidence, indicating that the discrimination between the two classes is clear and easy to identify. The final *counterfactual explanation* is then composed by two discriminant explanations, with the

roles of  $A$  and  $B$  reversed. It identifies the image regions informative of  $A$  but not  $B$  and the regions informative of  $B$  but not  $A$ .

As illustrated in Figures 3.1 and 3.3, the presentation of these regions side by side allows the user to visualize how the image of  $A$  would need to be changed in order to be classified as  $B$  (and vice-versa). This shows that counterfactual explanations can be seen as a *generalization* of attributive explanations, computed by a *combination* of attribution and confidence prediction methods that is much more efficient to compute than previous methods. In fact, our experiments show that their computation is  $50\times$  to  $1000\times$  faster for popular networks. This is quite important for applications such as machine teaching, where explanation algorithms should operate in real-time, ideally in low-complexity platforms such as mobile devices.

## 3.4 Implementation of GALORE

In this section, we discuss a unified framework for implementation of the explanations discussed above.

### 3.4.1 Explanation Framework

Consider an object recognition system  $\mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$ , mapping images  $\mathbf{x} \in \mathcal{X}$  into classes  $y \in \mathcal{Y} = \{1, \dots, C\}$ , according to a classifier

$$y^* = \arg \max_y h_y(\mathbf{x}), \tag{3.1}$$

where  $\mathbf{h}(\mathbf{x}) : \mathcal{X} \rightarrow [0, 1]^C$  is a  $C$ -dimensional probability distribution with  $\sum_{y=1}^C h_y(\mathbf{x}) = 1$ , usually computed by a convolutional neural network (CNN). The classifier is denoted self-aware if it produces a *confidence score*  $s(\mathbf{x}) \in [0, 1]$ , encoding the strength of its belief

that the image  $\mathbf{x}$  belongs to the predicted class  $y^*$ . The confidence score can be generated by the classifier itself, in which case it is denoted as *self-referential*, or by a complementary network, in which case it is *non-self-referential*. Both the classifier and the confidence score generator are learned from a training set  $\mathcal{D}$  of  $N$  i.i.d. samples  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ , where  $y_i \in \mathcal{Y}$  is the label of image  $\mathbf{x}_i \in \mathcal{X}$ . Classification performance is evaluated on a disjoint test set  $\mathcal{T} = \{(\mathbf{x}_j, y_j)\}_{j=1}^M$ .

In this work, we propose a *GenerAlized expLanatiOn fRamEwork* (GALORE) to unify various visualization-based explanations, accounting for both confidence scores and a set  $\mathcal{C}$  of class labels of interest beyond the prediction  $y^*$ . All GALORE explanations are implemented with a heat map

$$\mathcal{M}(\mathbf{x}, h_{y^*}, \mathcal{C}) = m^\alpha(\mathbf{a}(h_{y^*}(\mathbf{x}))) \cdot \prod_{c \in \mathcal{C}} m^\beta(\mathbf{a}(h_{y^c}(\mathbf{x}))) \cdot m^\gamma(\mathbf{a}(s(\mathbf{x}))), \quad (3.2)$$

where  $\mathbf{a}(\cdot)$  is an attribution function, and  $m^\alpha$ ,  $m^\beta$  and  $m^\gamma$  are three functions that depend on the visualization strategy. Explanations are provided in the form of collections image segments [180, 181, 182] obtained by thresholding the heat map. We next discuss how (3.2) is used to implement different visualization strategies.

### 3.4.2 Attributive Explanations

Attributive explanations visualize how strongly the prediction  $y^*$  is attributed to different regions of image  $\mathbf{x}$  [98, 99, 95, 136, 100]. They are obtained from (3.2) by setting  $m^\alpha(x) = x$ ,  $m^\beta(x) = m^\gamma(x) = 1$ , leading to heat map

$$\mathcal{A}(\mathbf{x}, y^*) = \mathbf{a}(h_{y^*}(\mathbf{x})). \quad (3.3)$$

The attribution function  $\mathbf{a}(\cdot)$  is usually applied to a tensor of activations  $\mathbf{F} \in \mathbb{R}^{W \times H \times D}$  of spatial dimensions  $W \times H$  and  $D$  channels, extracted at some layer of a deep network with

$\mathbf{x}$  at the input. While many attribution functions have been proposed, they are usually some variant of the gradient of  $h_{y^*}(\mathbf{x})$  with respect to  $\mathbf{F}$ . This results in an attribution map where the amplitude of  $\mathcal{A}_{ij}(\cdot)$  encodes the attribution of the prediction to each entry  $i, j$  along the spatial dimensions of  $\mathbf{F}$ . Two attributive heatmaps of an image of a “Cardinal” with respect to predictions “Cardinal” and “Summer Tanager,” are shown in the top row of Figure 3.3.

### 3.4.3 Self-aware Attributive Explanations

Attributive explanations can be extended to account for confidence scores by setting  $m^\gamma(x) = x$ . In this case, the attributive explanation becomes

$$\mathcal{A}(\mathbf{x}, y^*) = \mathbf{a}(h_{y^*}(\mathbf{x})) \cdot \mathbf{a}(s(\mathbf{x})). \tag{3.4}$$

Large heat map entries indicate regions that not only contribute to the prediction but also make the classifier confident about it. When compared to standard attributive explanations, the self-aware version emphasizes more class-specific regions. In experiments, we will see that these regions usually cover the attributes discriminant for the predicted classes, providing a sharper and more convincing explanation for the classifier prediction.

### 3.4.4 Deliberative Explanations

A deliberative explanation consists of a set of  $Q$  insecurities  $\{(\mathbf{r}_q, a_q, b_q)\}_{q=1}^Q$  that provide insight on the reasoning performed by the classifier to reach prediction  $y^*$ . Each insecurity is a triplet  $(\mathbf{r}, a, b)$ , where  $\mathbf{r}$  is the segmentation mask of a region responsible for classifier uncertainty, and  $(a, b)$  an ambiguity composed by a pair of class labels. Altogether, the insecurity shows that the network is insecure as to whether the image region defined by  $\mathbf{r}$  should be attributed to class  $a$  or  $b$ . Note that none of  $a$  or  $b$  has to be the prediction

$y^*$ , although this could happen for one of them. In Figure 3.2,  $y^*$  is the label “Pelagic Cormorant,” and appears in insecurities 2, 5, and 6, but not on the remaining. This reflects the fact that certain parts of the bird could actually be shared by many classes.

Insecurities are generated by first identifying the set  $\mathbb{C} = \{y^1, \dots, y^K\}$  of the  $K$  classes  $y$  of largest posterior probability  $h_y(\mathbf{x})$ . A *candidate class ambiguity set*  $\mathbb{A} = \binom{\mathbb{C}}{2}$  is then created with all class pairs in  $\mathbb{C}$ . For each ambiguity  $(a, b) \in \mathbb{A}$ , an *ambiguity map* is computed using (3.2) with  $\mathcal{C} = \{a, b\}$ ,  $m^\alpha(x) = 1$ ,  $m^\beta(x) = m^\gamma(x) = x$ , and  $s(\mathbf{x})$  replaced with  $1 - s(\mathbf{x})$ ,

$$\mathcal{I}(\mathbf{x}, \mathcal{C}) = \mathbf{a}(h_a(\mathbf{x})) \cdot \mathbf{a}(h_b(\mathbf{x})) \cdot \mathbf{a}(1 - s(\mathbf{x})). \quad (3.5)$$

Using as self-awareness score the complement of the belief in the prediction assigns larger scores to regions where the prediction is most ambiguous, reflecting the difficulty of the classifier decision.  $\mathcal{I}_{i,j}$  is large only when location  $(i, j)$  is deemed difficult to classify (large difficulty attribution  $\mathbf{a}(1 - s(\mathbf{x}))_{i,j}$ ) and this difficulty is due to large attributions to both classes  $a$  and  $b$ . The ambiguity map is thresholded to obtain the segmentation mask

$$\mathbf{r}\{a, b\}(\mathbf{x}) = \mathbb{1}_{\mathcal{I} > T}, \quad (3.6)$$

where  $\mathbb{1}_{\mathcal{S}}$  is the indicator function of set  $\mathcal{S}$  and  $T$  a threshold. The ambiguity  $(a, b)$  and the mask  $\mathbf{r}\{a, b\}(\mathbf{x})$  form an *insecurity*.

### 3.4.5 Counterfactual Explanations

While attributive and deliberative explanations assume a passive user, counterfactual explanations assume an interactive user who poses questions. Given image  $\mathbf{x}$  and prediction  $y^*$ , the user asks why not counterfactual class  $y^c \neq y^*$ . A popular counterfactual explanation approach is to use an image  $\mathbf{x}^c$  from class  $y^c$  and highlight the differences between  $\mathbf{x}$  and  $\mathbf{x}^c$

by displaying matched bounding boxes on the two images. [115] showed that explanation performance is nearly independent of the choice of  $\mathbf{x}^c$ , i.e. it suffices to use a random image  $\mathbf{x}^c$  from class  $y^c$ .

We adopt a similar strategy in this work, implementing counterfactual explanations as

$$\mathcal{R}(\mathbf{x}, y^*, y^c, \mathbf{x}^c) = (\mathcal{D}(\mathbf{x}, y^*, y^c), \mathcal{D}(\mathbf{x}^c, y^c, y^*)), \quad (3.7)$$

where  $\mathcal{D}(\mathbf{x}, y^*, y^c)$  and  $\mathcal{D}(\mathbf{x}^c, y^c, y^*)$  are *counterfactual heatmaps* for images  $\mathbf{x}$  and  $\mathbf{x}^c$ , respectively. The first map identifies the regions of  $\mathbf{x}$  that are informative of the predicted class but not the counter class while the second identifies the regions of  $\mathbf{x}^c$  informative of the counter class but not of the predicted class. Altogether, the explanation shows that the regions highlighted in the two images are matched: the region of the first image depicts features that *only* appear in the predicted class while that of the second depicts features that *only* appear in the counterfactual class. The counterfactual map of  $\mathbf{x}$  is thresholded to obtain the segmentation mask

$$\mathbf{r}\{y^*, y^c\}(\mathbf{x}) = \mathbb{1}_{\mathcal{D}(\mathbf{x}, y^*, y^c) > T}. \quad (3.8)$$

Similarly, a segmentation mask is generated for  $\mathbf{x}^c$  using

$$\mathbf{r}\{y^c, y^*\}(\mathbf{x}^c) = \mathbb{1}_{\mathcal{D}(\mathbf{x}^c, y^c, y^*) > T}. \quad (3.9)$$

Figure 3.3 illustrates the construction of a counterfactual explanation with two discriminant explanations.

To compute the heatmaps of (3.7), [115] proposed to exhaustively compare all combinations of features in  $\mathbf{x}$  and  $\mathbf{x}^c$ , which is expensive. We propose a much simpler and more effective procedure that leverages a new class of attributive explanations, denoted as



*discriminant* and defined as in (3.2), with  $m^\alpha(x) = m^\gamma(x) = x$ ,  $\mathcal{C} = \{y^c\}$ , and  $m^\beta(\mathbf{a}(\cdot))$  the complement of  $\mathbf{a}(\cdot)$ . i.e.

$$m^\beta(\mathbf{a}(\cdot))_{i,j} = \max_{i,j} \mathbf{a}_{i,j} - \mathbf{a}_{i,j}, \quad (3.10)$$

leading to heatmap

$$\mathcal{D}(\mathbf{x}, y^*, y^c) = \mathbf{a}(h_{y^*}(\mathbf{x})) \cdot m^\beta(\mathbf{a}(h_{y^c}(\mathbf{x}))) \cdot \mathbf{a}(s(\mathbf{x})). \quad (3.11)$$

This is large only at locations  $(i, j)$  that contribute strongly to the prediction of class  $y^*$  but little to that of class  $y^c$ , and where the discrimination between the two classes is easy, i.e. the classifier is confident. This, in turn, implies that location  $(i, j)$  is strongly specific to class  $y^*$  but not specific to class  $y^c$ , which is the essence of the counterfactual explanation.

Discriminant explanations have commonalities with both attributive and counterfactual explanations. Like counterfactual explanations, they consider both the prediction  $y^*$  and counterfactual class  $y^c$ . Like attributive explanations, they compute a single attribution map  $\mathcal{D}(\cdot, \cdot)$ . The difference is that this map *attributes the discrimination between the prediction  $y^*$  and counter  $y^c$  class* to regions of  $\mathbf{x}$ , identifying pixels strongly informative of class  $y^*$  but uninformative of class  $y^c$ . Figure 3.3 shows how these explanations benefit from the fact that the self-awareness attribution map is usually much sharper than the other two maps. This is critical to identify the object details that differentiate the two classes.

### 3.5 Implementation

Table 3.1 summarizes how GALORE produces different visualization-based explanations, including different types of attributive, deliberative, and counterfactual explanations. All explanations are obtained by combinations of attribution maps and classification

**Table 3.1:** Implementation of different explanation strategies under the GALORE framework.

explanation	heatmap	$m^\alpha(\mathbf{a})$	$m^\beta(\mathbf{a})$	$m^\gamma(\mathbf{a})$	$\mathcal{C}$	$s(\mathbf{x})$
Attributive	$\mathcal{A}(\mathbf{x}, y^*)$	$\mathbf{a}$	1	1	None	None
Self-aware attributive	$\mathcal{A}(\mathbf{x}, y^*)$	$\mathbf{a}$	1	$\mathbf{a}$	None	$s(\mathbf{x})$
Deliberative	$\mathcal{I}(\mathbf{x}, \mathcal{C})$	1	$\mathbf{a}$	$\mathbf{a}$	$\{a, b\}$	$s(\mathbf{x}) \leftarrow 1 - s(\mathbf{x})$
Counterfactual	$\mathcal{R}(\mathbf{x}, y^*, y^c, \mathbf{x}^c)$	$\mathbf{a}$	$\max_{i,j} \mathbf{a}_{i,j} - \mathbf{a}_{i,j}$	$\mathbf{a}$	$\{y^c\}$	$s(\mathbf{x})$

confidence scores using (3.2). In this section, we discuss how these are computed.

### 3.5.1 Attribution Maps

Given a feature tensor  $\mathbf{F}(\mathbf{x})$  in some deep network layer, attribution map  $\mathbf{a}_{i,j}(h_y(\mathbf{x}))$  quantifies how the activations  $\mathbf{F}_{i,j}(\mathbf{x})$  at locations  $(i,j)$  contribute to prediction  $y$ . This could be either a class prediction or the prediction of a confidence score. In this section, we make no distinction between the two, simply denoting  $p(\mathbf{x}) = g_p(\mathbf{F}(\mathbf{x}))$ , where  $g$  is the mapping from activation tensor  $\mathbf{F}$  into prediction vector  $g(\mathbf{F}) \in [0, 1]^P$ . For class predictions  $P = C$ , the prediction  $p$  is a class  $y$ , and  $g_p(\mathbf{F}(\mathbf{x})) = h_y(\mathbf{x})$ . For confidence predictions  $P = 1$ , the prediction is a confidence score, and  $g_p(\mathbf{F}(\mathbf{x})) = s(\mathbf{x})$ .

GALORE is compatible with any attribution function in the literature [98, 103, 126, 95, 136, 104]. One of the most popular class of such functions is that of gradient-based attributions [98, 136, 103], which are derived from  $\nabla g_p(\mathbf{F}(\mathbf{x}))$  and  $\mathbf{F}(\mathbf{x})$ , i.e. have the form  $q([\nabla g_p(\mathbf{F}(\mathbf{x}))]_{i,j}, \mathbf{F}_{i,j}(\mathbf{x}))$  for some function  $q$ . Our implementation uses the vanilla gradient based function of [98], which computes the dot-product of the partial derivatives of prediction  $p$  with respect to activations  $\mathbf{F}(\mathbf{x})$  by these activations,

$$\mathbf{a}_{i,j}^p = [\nabla g_p(\mathbf{F})]_{i,j}^T \mathbf{F}_{i,j}. \quad (3.12)$$

Here we omit the dependency on  $\mathbf{x}$  for simplicity.

This is compared to two more complex attribution functions, integrated gradient (InteGrad) [136] and GradCAM [103]. InteGrad is based on the Riemman approximation of the integral of the gradient  $\nabla g_p$  along a linear path from a reference  $\mathbf{F}^0$  to the observed

activation tensor  $\mathbf{F}$ ,

$$\mathbf{a}_{i,j}^p = \left( \sum_{k=1}^Q [\nabla g_p(\mathbf{F}) \Big|_{\mathbf{F}^0 + \frac{k}{Q} \times (\mathbf{F} - \mathbf{F}^0)}]_{i,j} \cdot \frac{1}{Q} \right)^T (\mathbf{F}_{i,j} - \mathbf{F}_{i,j}^0), \quad (3.13)$$

where  $Q$  is the number of steps in the approximation and set to 50. The reference  $\mathbf{F}^0$  is defined by the user and often chosen to be the image that induces zero activation. Unlike (3.12), which only uses the partial derivative at activation  $\mathbf{F}_{i,j}(\mathbf{x})$ , InteGrad computes the average gradient along the linear path from  $\mathbf{F}^0$  to  $\mathbf{F}$ . Grad-CAM [103] assigns a unique weight per activation channel  $k$ , which is the spatial mean of the activations of this channel

$$\mathbf{a}_{i,j}^p = \text{ReLU} \left( \sum_k w_k \mathbf{F}_{i,j,k} \right), \quad (3.14)$$

where  $w_k = \frac{1}{W \times H} \sum_{i,j} \frac{\partial g_p(\mathbf{F})}{\partial \mathbf{F}_{i,j,k}}$ . In our implementation, the attribution maps of (3.12), (3.13), (3.14) are normalized to  $[0, 1]$  by min-max normalization, i.e. subtracting the minimum value and dividing by the maximum.

GALORE is also compatible with non gradient-based attribution functions [180, 183, 104]. In experiments, we present comparisons to score-CAM [104], a representative of these methods. Like Grad-CAM, its attribution map is a weighted sum of the activation maps but the weight  $w_k$  of (3.14) is not derived from gradients, involving forwarding computations only. We omit the details for brevity.

### 3.5.2 Confidence Scores

Beyond attribution maps, GALORE is compatible with many classification confidence scores. We consider three scores of different characteristics. The *softmax score* [113]

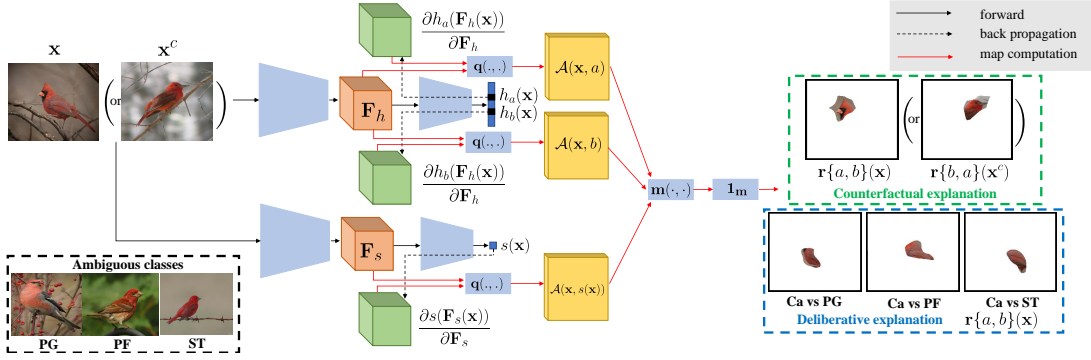


Figure 3.4: GALORE explanation architecture.

is the largest class posterior probability

$$s^s(\mathbf{x}) = \max_y h_y(\mathbf{x}). \quad (3.15)$$

It is computed by adding a max pooling layer to the network output. The *certainty score* is the complement of the normalized entropy of the softmax distribution [114],

$$s^c(\mathbf{x}) = 1 + \frac{1}{\log C} \sum_y h_y(\mathbf{x}) \log h_y(\mathbf{x}). \quad (3.16)$$

Its computation requires an additional layer of log non-linearities and average pooling. These two scores are self-referential. We also consider the non-self-referential *easiness score* of [177],

$$s^e(\mathbf{x}) = 1 - s^{hp}(\mathbf{x}) \quad (3.17)$$

where  $s^{hp}(\mathbf{x})$  is computed by an external predictor  $\mathcal{S}$ , which predicts the difficulty of classifying each example and is trained jointly with the classifier.  $\mathcal{S}$  is implemented by a network  $s^{hp}(\mathbf{x}) : \mathcal{X} \rightarrow [0, 1]$  whose output is a sigmoid unit.

### 3.5.3 Network Implementation

Figure 3.4 shows a network implementation of (3.2). Given a query image  $\mathbf{x}$  of class  $y^*$ , a user-selected counter class  $y^c \neq y^*$ , a predictor  $h_y(\mathbf{x})$ , and a confidence predictor  $s(\mathbf{x})$  are used to produce the explanation. Note that  $s(\mathbf{x})$  can share weights with  $h_y(\mathbf{x})$  (self-referential) or be separate (non-self-referential).  $\mathbf{x}$  is forwarded through the network, generating activation tensors  $\mathbf{F}_h(\mathbf{x})$ ,  $\mathbf{F}_s(\mathbf{x})$  in pre-chosen network layers and predictions  $h_a(\mathbf{x})$ ,  $h_b(\mathbf{x})$ ,  $s(\mathbf{x})$ , which depend on the explanation strategy. For deliberative explanations, the predictions are classes  $a$ ,  $b$  from the candidate ambiguities set. For counterfactual explanations, they are  $h_{y^*}(\mathbf{x})$ ,  $h_{y^c}(\mathbf{x})$ ,  $s(\mathbf{x})$ . The attributions of  $a$ ,  $b$  and  $s(\mathbf{x})$  to  $\mathbf{x}$ , i.e.  $\mathcal{A}(\mathbf{x}, a)$ ,  $\mathcal{A}(\mathbf{x}, b)$ ,  $\mathcal{A}(\mathbf{x}, s(\mathbf{x}))$  are then computed with (3.12), (3.13), or (3.14), which reduce to a backpropagation step with respect to the desired layer activations and a few additional operations. These attributions can also be computed by other non-gradient-based functions. Finally, the attributions are combined with (3.5) or (3.11). Thresholding the resulting heatmap with (3.6) or (3.8) produces the deliberative explanation  $\mathbf{r}\{a, b\}(\mathbf{x})$  or discriminant explanation  $\mathbf{r}\{y^*, y^c\}(\mathbf{x})$ . For counterfactual explanations, the network is simply applied to  $\mathbf{x}^c$  to compute  $\mathbf{r}\{y^c, y^*\}(\mathbf{x}^c)$ .

## 3.6 Evaluation

Explanations can be difficult to evaluate, since ground truth is usually not available. Two major classes of evaluation strategies have been proposed.

### 3.6.1 User Experiments

One possibility is to perform Turk experiments, e.g. measuring whether humans can predict a class label given a visualization, or identify the most trustworthy of two models that make identical predictions from their explanations [103]. We use a similar strategy

for deliberative explanations, by measuring whether, given an insecurity produced by the explanation algorithm, humans can predict the associated ambiguities. For counterfactual explanations, we use instead a machine teaching setting, testing whether the explanation helps humans distinguish different classes. While these strategies directly measure how intuitive the explanations appear to humans, they require subject experiments that are somewhat cumbersome to perform and difficult to replicate.

### 3.6.2 Proxy Tasks

A second evaluation strategy uses a proxy task, such as localization [102, 103] on datasets with object bounding boxes. While this is much easier to implement, there is usually no groundtruth for regions of importance to the classification of an image. We overcome this problem by leveraging datasets annotated<sup>2</sup> with parts and attributes. Specifically, where the  $k^{th}$  part of an object of class  $c$  is annotated with a semantic descriptor  $\phi_c^k$  containing the attributes present in this class. For example, in a bird dataset, the “eye” part can have color attribute values “green,” “blue,” “brown,” etc. The descriptor is a probability distribution over these values, characterizing the variability of attribute values of the part per class. Explanation ground-truth is derived from attribute distributions, as described next.

#### Deliberative explanations

For deliberative explanations, we define insecurities as *ambiguous parts*, namely object parts common to multiple object classes or scene parts (e.g. objects) shared by scene classes. This reduces evaluation to insecurity localization.

For binary explanations, the similarity between classes  $a$  and  $b$  according to part

---

<sup>2</sup>Note that part and attribute annotations are only required to evaluate the accuracy of insecurities, not to compute the visualizations. These require no annotation.

$k$  is defined as  $\alpha_{a,b}^k = \gamma(\phi_a^k, \phi_b^k)$ , where  $\gamma$  is a dataset dependent similarity measure. This reflects the *strength of the ambiguity* between classes  $a$  and  $b$ , declaring as ambiguous parts that have similar attribute distributions under the two classes. To generate ground-truth, the values of  $\alpha_{a,b}^k$  are computed for all parts  $\mathbf{p}_k$  and class pairs  $(a, b)$ . The  $M$  triplets  $\mathcal{G}^d = \{(\mathbf{p}_i, a_i, b_i)\}_{i=1}^M$  of largest similarity in  $\mathcal{G} = \{(\mathbf{p}_i, a_i, b_i)\}_{i=1}^{C \times C \times K}$  are selected as insecurity ground-truth, where  $K$  is the total number of parts.

Given this groundtruth, two metrics are used to evaluate the quality of the explanations, depending on the nature of part annotations. For datasets where parts are labelled with a single location (usually the geometric center of the part), i.e.  $\mathbf{p}_i$  is a point, the quality of segment  $\mathbf{r}\{a, b\}(\mathbf{x})$  is computed by precision (P) and recall (R). Here,  $P = \frac{J}{|\{k | \mathbf{p}_k \in \mathbf{r}\}|}$ ,  $R = \frac{J}{|\{i | (\mathbf{p}_i, a_i, b_i) \in \mathcal{G}, a_i = a, b_i = b\}|}$  and  $J = |\{i | \mathbf{p}_i \in \mathbf{r}, a_i = a, b_i = b\}|$  is the number of ground-truth parts included in the insecurities that compose the explanation. Precision-recall curves are produced by varying the threshold  $T$  of (3.6). For datasets where parts have segmentation masks, the quality of  $\mathbf{r}\{a, b\}(\mathbf{x})$  is computed by the intersection over union (IoU) metric  $\text{IoU} = \frac{|\mathbf{r} \cap \mathbf{p}|}{|\mathbf{r} \cup \mathbf{p}|}$ , where  $\mathbf{p} = \{\mathbf{p}_i | (\mathbf{p}_i, a_i, b_i) \in \mathcal{G}^d, a_i = a, b_i = b\}$ .

## Counterfactual explanations

For counterfactual explanations, where the goal is to localize a region predictive of class  $A$  but unresponsive of class  $B$ , groundtruth is assembled by identifying parts with attributes specific to  $A$  that do not appear in  $B$ . This enables the evaluation of counterfactual explanations as a class-specific part localization problem.

For two-class explanations, where  $\alpha_{a,b}^k$  measures the similarity between two classes according to part  $k$ , a small  $\alpha_{a,b}^k$  indicates that part  $k$  discriminates between the two classes. To generate ground-truth, the  $N$  parts of smallest similarity in  $\mathcal{G}$ ,  $\mathcal{G}^c = \{(\mathbf{p}_i, a_i, b_i)\}_{i=1}^N$  are selected as counterfactual ground-truth.

For two-class counterfactual explanations, evaluation is based on the precision-recall



and IoU metrics used for deliberative explanations. On datasets with point-based ground truth, evaluation is based on precision and recall of the generated counterfactual regions. On datasets with mask-based ground truth, the IoU is used.

We also define a metric that captures the semantic consistency of two segments,  $\mathbf{r}\{a,b\}(\mathbf{x})$  and  $\mathbf{r}\{b,a\}(\mathbf{x}^c)$ , by calculating the consistency of the parts included in them. This is denoted as the part IoU (PIoU),

$$\text{PIoU} = \frac{|\{k | (\mathbf{p}_k, a, b) \in \mathbf{r}\{a,b\}(\mathbf{x})\} \cap \{k | (\mathbf{p}_k, b, a) \in \mathbf{r}\{b,a\}(\mathbf{x}^c)\}|}{|\{k | (\mathbf{p}_k, a, b) \in \mathbf{r}\{a,b\}(\mathbf{x})\} \cup \{k | (\mathbf{p}_k, b, a) \in \mathbf{r}\{b,a\}(\mathbf{x}^c)\}|}. \quad (3.18)$$

This metric provides a fair comparison of different explanations if their counterfactual regions have the same size. Region size is controlled by  $T$  in (3.8) and (3.9).

User expertise has an impact on counterfactual explanations. Beginner users tend to choose random counterfactual classes, while experts tend to pick counterfactual classes similar to the true class. Hence, explanation performance should be measured for the two user types. In this work, users are simulated by choosing a random counterfactual class  $b$  for beginners and the class predicted by a small CNN for advanced users. Class  $a$  is the prediction of the classifier used to generate the explanation, which is a larger CNN.

### Attributive explanations

For attributive explanations, ground-truth consists of parts with unique attributes, present in the ground truth class and lacking in all other classes. However, it is frequently impossible to find a part whose attributes appear in a single class. Hence, we randomly select  $L$  classes from  $\mathcal{Y} \setminus \{y^*\}$ , to create a label set  $\mathcal{L} = \{y_1, \dots, y_L\}$  and use the evaluation metrics discussed for counterfactual explanations.

## 3.7 Experiments

In this section we discuss an experimental evaluation of the explanations generated by GALORE.

### 3.7.1 Experimental Setup

**Datasets:** Experiments were performed on the CUB200 [178] and ADE20K [184] datasets. CUB200 [178] is a densely-labeled dataset of fine-grained bird classes, annotated with parts. 15 part locations (points) are annotated including back, beak, belly, breast, crown, forehead, left/right eye, left/right leg, left/right wing, nape, tail and throat. Attributes are defined and assigned to each part according to [178]. ADE20K [184] is a fine-grained scene image dataset with more than 1000 scene categories and segmentation masks for 150 objects. In this case, objects are seen as scene parts and each object has a single attribute, which is its probability of appearance in a scene. Both datasets were subject to standard normalizations. All results are presented on the standard CUB200 test set and the official validation set of ADE20K.

**Networks:** VGG16 [185] is the most popular architecture in the explanation literature. Unless otherwise noted, it is used for all visualizations. It is also compared to the ResNet-50 [1] and AlexNet [186]. All predictors are trained by standard strategies [185, 1, 186, 114, 177]. The last convolutional layer output, widely used in the visualization literature [187, 102, 103], is used to create all explanations.

**Evaluation:** On CUB200, where all semantic descriptors  $\phi_c^k$  are multidimensional, similarities  $\alpha_{a,b}^k$  are computed with  $\gamma(\phi_a^k, \phi_b^k) = e^{-\{\text{KL}(\phi_a^k \parallel \phi_b^k) + \text{KL}(\phi_b^k \parallel \phi_a^k)\}}$  [188], where  $\text{KL}(\cdot \parallel \cdot)$  is the Kullback–Leibler divergence. To generate groundtruth for insecurities and discriminant regions, the set  $\mathcal{G}$  of region and class tuples was divided into two subsets. The size  $M$  of the set of groundtruth insecurities was set to the 20% insecurities  $(\mathbf{p}_i, a_i, b_i)$  of

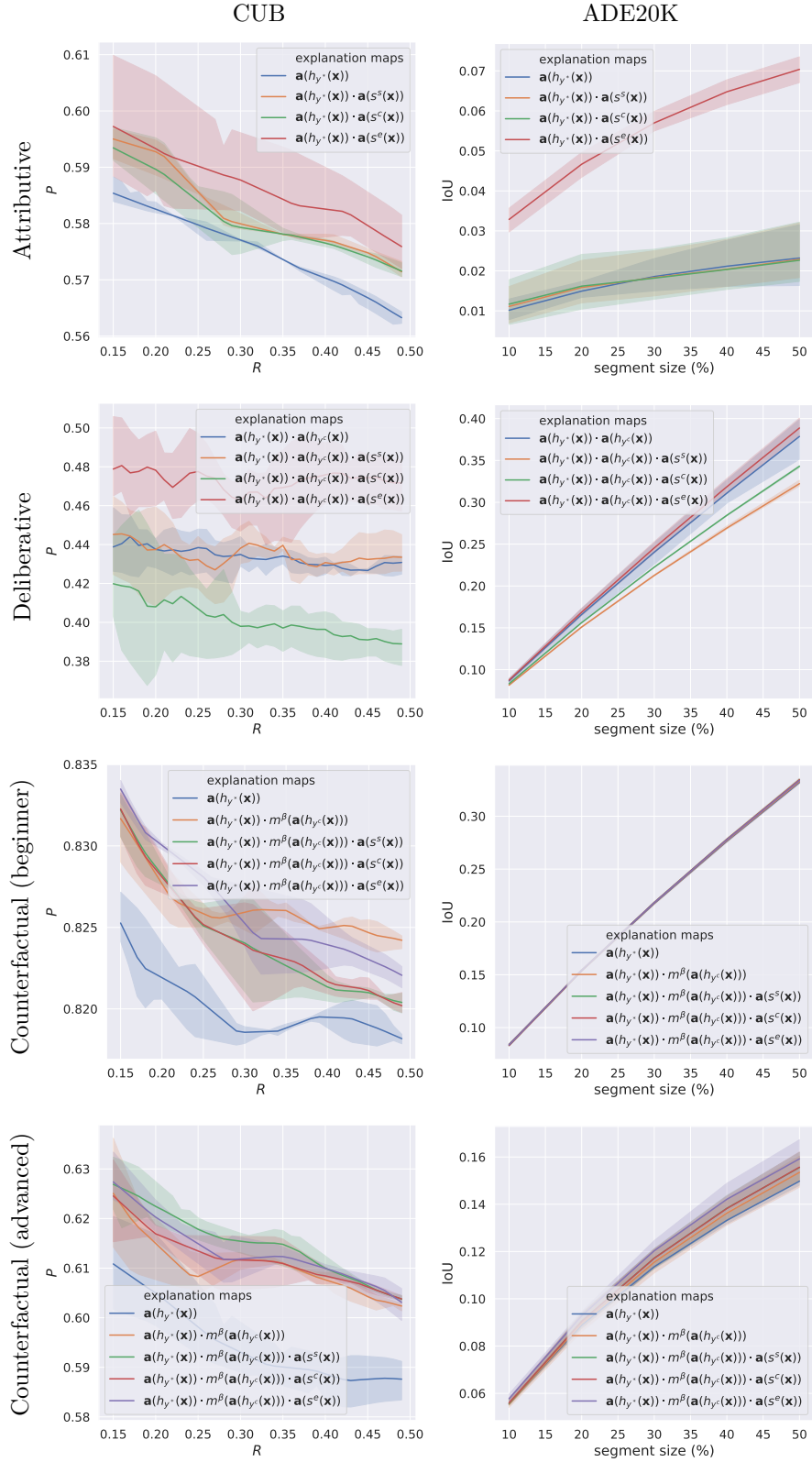
strongest ambiguity. The size  $N$  of the set of discriminant groundtruth regions was set to the remaining 80% parts  $(\mathbf{p}_i, a_i, b_i)$  of smallest similarity. This division reflects the fact that dissimilar parts dominate  $\mathcal{G}$ . Since parts are labelled with points, accuracy is measured with precision and recall.

On ADE20K, the semantic descriptors  $\phi_c^k$  are scalar (where  $k \in \{1, \dots, 150\}$ ) namely the probability of occurrence of part (object)  $k$  in scenes of class  $c$ . This is estimated by the relative frequency with which the part appears in scenes of the class. Only parts such that  $\phi_c^k > 0.3$  are considered. For deliberative explanations, ambiguity strengths are computed with  $\gamma(\phi_a^k, \phi_b^k) = \frac{1}{2}(\phi_a^k + \phi_b^k)$ . This is large when object  $k$  appears very frequently in both classes, i.e. the object adds ambiguity. Due to the sparsity of the matrix of ambiguity strengths  $\alpha_{a,b}^k$ , the number  $M$  of ground-truth insecurities is set to the 1% triplets of strongest ambiguity. On the other hand, counterfactual ground truth consists of the triplets  $(\mathbf{p}_i, a_i, b_i)$  with  $\phi_a^k > 0$  and  $\phi_b^k = 0$ , i.e. where object  $k$  appears in class  $a$  but not in class  $b$ .

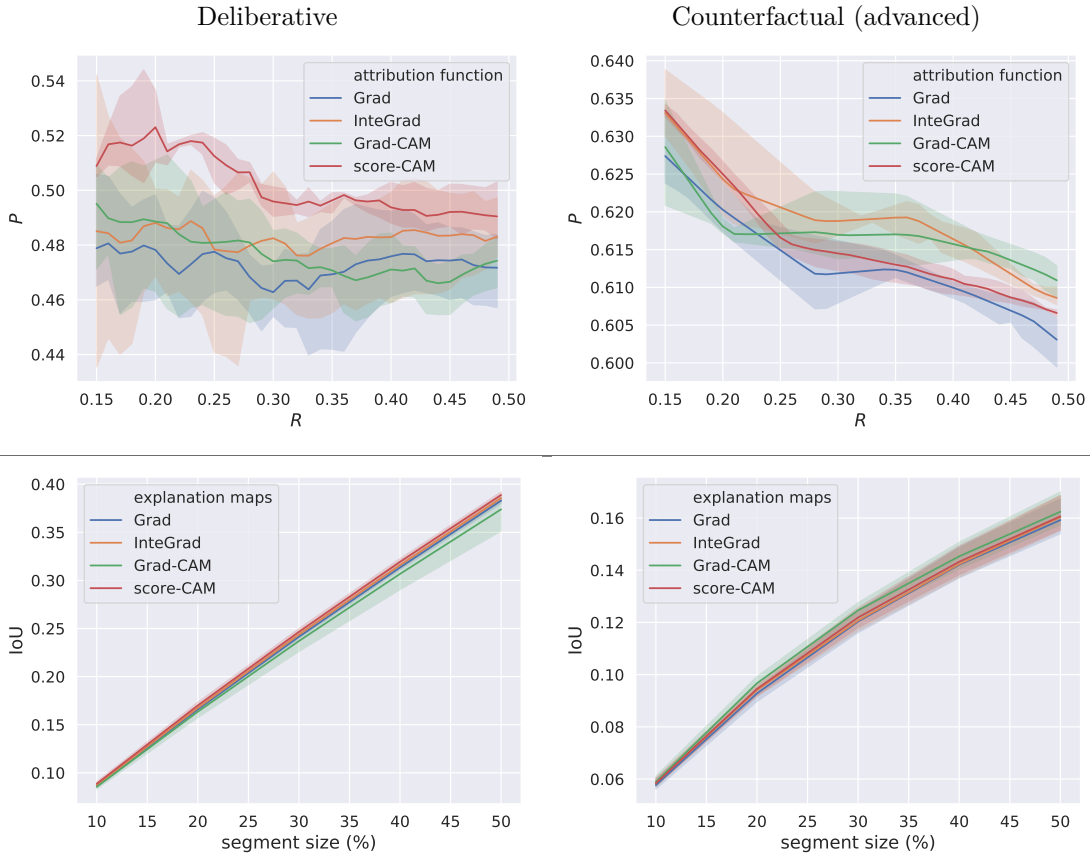
Since deliberative explanations aim to explain examples that are difficult to classify, explanations are produced only for the 100 test images of largest difficulty score on each dataset. The  $K = 5$  top classes are used to produce the class ambiguity set (see Section 3.4.4). In counterfactual explanations, AlexNet predictions [186] are used to mimic advanced users.

### 3.7.2 Ablation Study

**Self-awareness scores:** Figure 3.5 shows the impact of the confidence scores of (3.15)-(3.17) on precision-recall curves (on CUB200) and IoU (on ADE20K) for three explanation strategies. Some conclusions can be drawn. First, self-awareness is useful for all explanations. For attributive explanations, self-awareness attribution functions highlight more class-specific features. For counterfactual explanations, the gains are larger for expert users than for beginners. This is because the counter and predicted classes



**Figure 3.5:** Effect of confidence scores on precision-recall curves and IoU of different GALORE explanations.

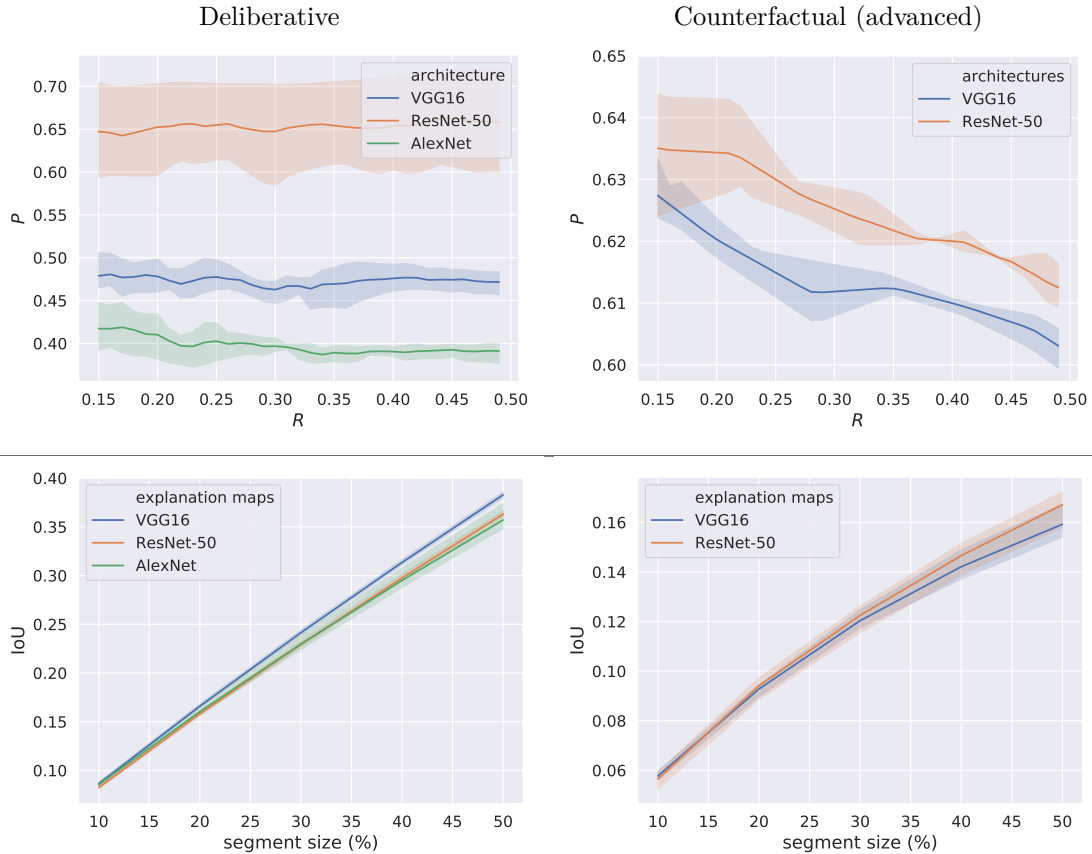


**Figure 3.6:** Impact of attribution function on GALORE explanation performance. Top: precision-recall on CUB200. Bottom: IoU on ADE20K.

are more similar for the former, producing attribution maps that overlap. Second, the easiness score substantially outperforms the remaining scores, for all but counterfactual explanations with beginner users, where counter classes are easy to distinguish. Third, for deliberative explanations, only the easiness score  $s^e(\mathbf{x})$  improves on the baseline. This suggests that self-referential difficulty scores are not always reliable. For this reason, the easiness score is used in the remaining experiments.

**Attribution Function:**<sup>3</sup> GALORE is compatible with any attribution function. Figure 3.6 compares different functions: baseline gradient (‘Grad’), the integrated gradient of [136] (‘InteGrad’), Grad-CAM [103], and score-CAM [104]. For brevity, we only present

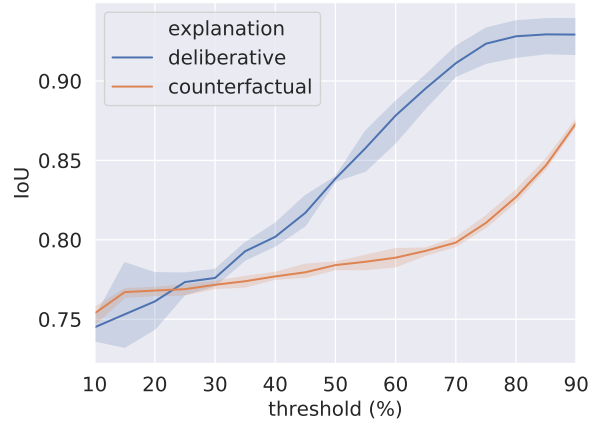
<sup>3</sup>Since no new algorithm is proposed for attributive explanations, ablations are restricted to deliberative and counterfactual explanations in the remainder of the chapter.



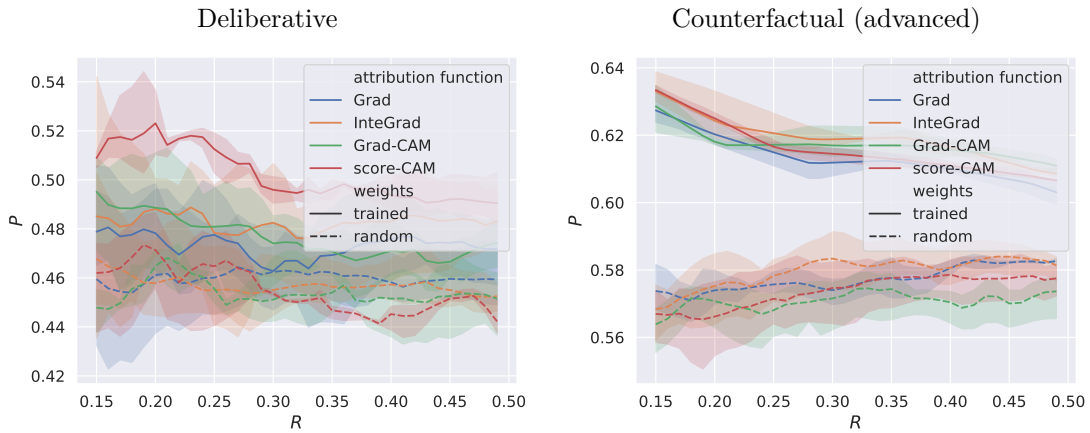
**Figure 3.7:** Impact of network architecture on GALORE explanation performance. Top: precision-recall on CUB200. Bottom: IoU on ADE20K.

deliberative and counterfactual results for advanced users. A few conclusions are possible. First, while the three more complex functions always outperform Grad, the differences are small, especially on ADE20K. This is probably because ADE20K is more difficult (more than 1000 categories and only about 16 examples per category) than CUB200 (200 categories and 26 examples per category). Second, while GALORE benefits from advanced attribution functions, there is little difference between InteGrad, Grad-CAM and score-CAM.

**Network Architectures:** Figure 3.7 compares the explanations produced by ResNet-50, VGG16 and AlexNet. For counterfactual explanations, only the former two are compared because AlexNet is used to simulate the users. On CUB200, ResNet-50 has the



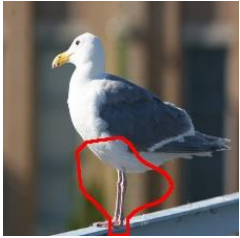


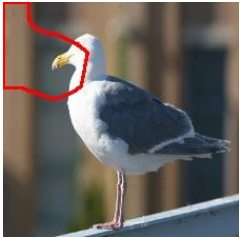


**Figure 3.8:** Robustness of GALORE to image shifts on CUB200.









**Figure 3.9:** Precision-recall of GALORE explanations obtained with pre-trained and random weights on CUB200.

best performance. Interestingly, although ResNet-50 and VGG16 have similar classification performance on these two datasets, the ResNet segments are much more accurate than those of VGG16. This suggests that the ResNet architecture uses more intuitive, i.e. human-like, deliberations. On ADE20K, where the classification task is harder ( $< 60\%$  mean accuracy), there is no clear difference between the three architectures.

Glaucous Gull

Insecurity	Ambiguity		
<p><b>Class:</b> Glaucous gull</p> 	<p><b>Class:</b> California gull</p> 	<p><b>Class:</b> Herring gull</p> 	<p><b>Shared Part:</b></p> <ul style="list-style-type: none"> <li>• Leg color is buff;</li> <li>• Belly color is white and pattern is solid;</li> </ul>
<p><b>Class:</b> Glaucous gull</p> 	<p><b>Class:</b> Western gull</p> 	<p><b>Class:</b> Glaucous gull</p> 	<p><b>Shared Part:</b></p> <ul style="list-style-type: none"> <li>• Bill shape is hooked;</li> <li>• Forehead color is white;</li> </ul>

Black Tern

Insecurity	Ambiguity		
<p><b>Class:</b> Black tern</p> 	<p><b>Class:</b> Artic tern</p> 	<p><b>Class:</b> Elegant tern</p> 	<p><b>Shared Part:</b></p> <ul style="list-style-type: none"> <li>• Tail shape is forked;</li> <li>• Tail pattern is solid;</li> </ul>
<p><b>Class:</b> Black tern</p> 	<p><b>Class:</b> Elegant tern</p> 	<p><b>Class:</b> Forsters tern</p> 	<p><b>Shared Part:</b></p> <ul style="list-style-type: none"> <li>• Wing color is white;</li> <li>• Wing shape is long;</li> <li>• Wing pattern is solid;</li> </ul>

**Figure 3.10:** Deliberative explanations produced by GALORE for two images from CUB.








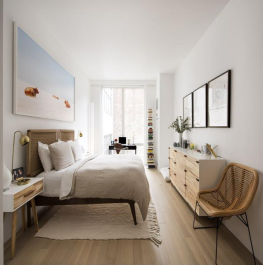





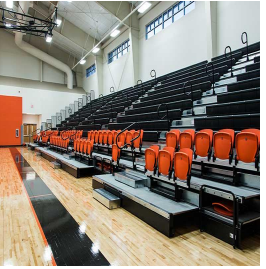
Insecurity	Ambiguity		
<p><b>Class: Plaza</b></p> 	<p><b>Class: Hacienda</b></p> 	<p><b>Class: Mosque</b></p> 	<p><b>Shared Part:</b> Building Edifice</p>
<p><b>Class: Bedroom</b></p> 	<p><b>Class: living room</b></p> 	<p><b>Class: Bedroom</b></p> 	<p><b>Shared Part:</b> Wall Floor Ceiling Window</p>
Insecurity	Ambiguity		
<p><b>Class: Junk pile</b></p> 	<p><b>Class: Barnyard</b></p> 	<p><b>Class: Vege Garden</b></p> 	<p><b>Shared Part:</b> Soil Tree Grass</p>
<p><b>Class: misc</b></p> 	<p><b>Class: auditorium</b></p> 	<p><b>Class: bleachers</b></p> 	<p><b>Shared Part:</b> Wall Floor Light Chair</p>

Figure 3.11: Deliberative explanations produced by GALORE for four images from ADE20K.

### 3.7.3 Sanity Checks

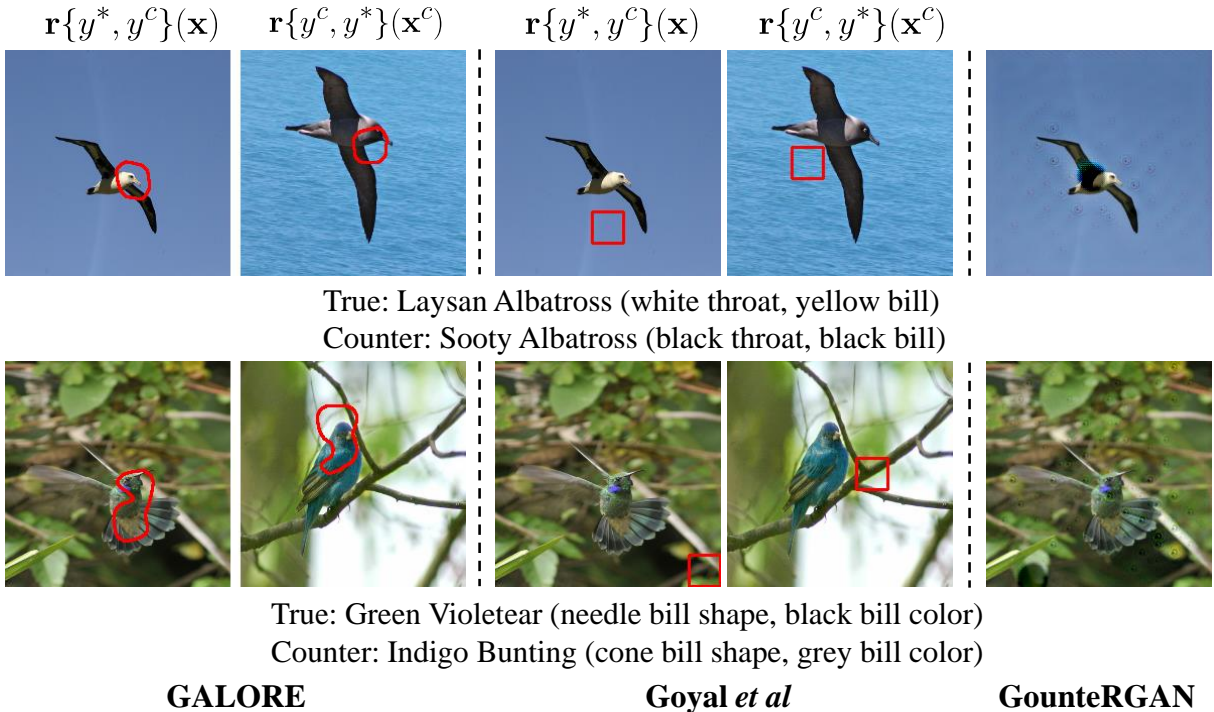
Recent works have shown that attribution maps can be sensitive to data shifts and model variance [165, 166]. We conducted two basic sanity checks for all visualizations: a data shift check and a model parameter randomization test. Data shift checks [165] test the robustness of the explanation to input shifts. For this, test images were randomly translated by 1 to 10 pixels along four directions. The resulting insecurities and counterfactual segments were compared to those obtained without translations, by measuring the similarity (IoU) between segments. The average IoU across all segments and examples is shown in Figure 3.8 as a function of the threshold  $T$ . While these are plots for the ‘easiness-Grad-VGG’ configuration, they are typical. The average IoU is almost always above 75% showing that the explanations of GALORE are robust to image shifts. Parameter randomization tests [166] compare the explanation of well-trained and random initialized models. Similar outputs indicate that the explanation method is insensitive to model parameters, which is undesirable. Figure 3.9 shows that all attribution functions passed the sanity check, since pre-trained models always outperformed random initialization. This was especially true for score-CAM and the differences were larger for counterfactual explanations.

### 3.7.4 Visualizations

Figure 3.10 shows two examples of deliberative explanations of two insecurities each. The top of the figure shows the insecurities of the classifier for an image of a ‘Glaucous gull’. The top insecurity covers the leg/belly region, which is a region of ambiguity with classes ‘California gull’ and ‘Herring gull’ that also have leg color ‘buff’, belly color ‘white’, and belly pattern ‘solid’. The lower insecurity covers the bill/forehead region of the gull, due to an ambiguity between the ‘Glaucous gull’ and the ‘Western gull’ with whom the ‘Glaucous gull’ shares a ‘hooked’ bill shape and a ‘white’ colored forehead. The bottom of

the figure shows insecurities for a ‘Black tern,’ due to a tail ambiguity with ‘Artic’ and ‘Elegant’ terns and a wing ambiguity with ‘Elegant’ and ‘Forsters’ terns. Figure 3.11 shows single insecurities from four images of ADE20K. In all cases, the insecurities correlate with regions of attributes shared by different classes. This shows that deliberative explanations unveil truly ambiguous image regions, generating intuitive insecurities that help understand network predictions. Note, for example, how the visualization of insecurities tends to highlight classes that are semantically very close, such as the different families of gulls or terns and class subsets such as ‘plaza’, ‘hacienda’, and ‘mosque’ or ‘bedroom’ and ‘living room’. All of this suggests that the deliberative process of the network correlates well with human reasoning.

Figure 3.12 shows two examples of counterfactual visualizations on CUB200. The regions selected in the query and counter class image are shown in red. For CounterGAN [152], the generated explanatory images are shown. The true  $y^*$  and counter  $y^c$  class are shown below the images and followed by the ground truth discriminative attributes for the image pair. Note how GALORE explanations identify semantically matched and class-specific bird parts on both images. For example, the throat and bill that distinguish Laysan from Sooty Albatrosses. This feedback enables a user to learn that Laysans have white throats and yellow bills, while Sootys have black throats and bills. This is unlike the regions produced by [115], also shown in the figure, which sometimes highlight irrelevant cues, such as the background. CounterGAN, only generates some patterns from the counterfactual classes (zoom in for more detail), but not realistic images. This is consistent with the well known difficulty of GANs to translate images across hundreds of fine grained classes. Figure 3.13 presents similar figures for ADE20K, where the proposed explanations tend to identify scene-discriminative objects. For example, that a promenade deck contains objects ‘floor’, ‘ceiling’, ‘sea,’ while a bridge scene includes ‘tree’, ‘river’ and ‘bridge’.



**Figure 3.12:** Counterfactual explanations (true and counter classes shown below each example, ground truth class-specific part attributes in parenthesis).

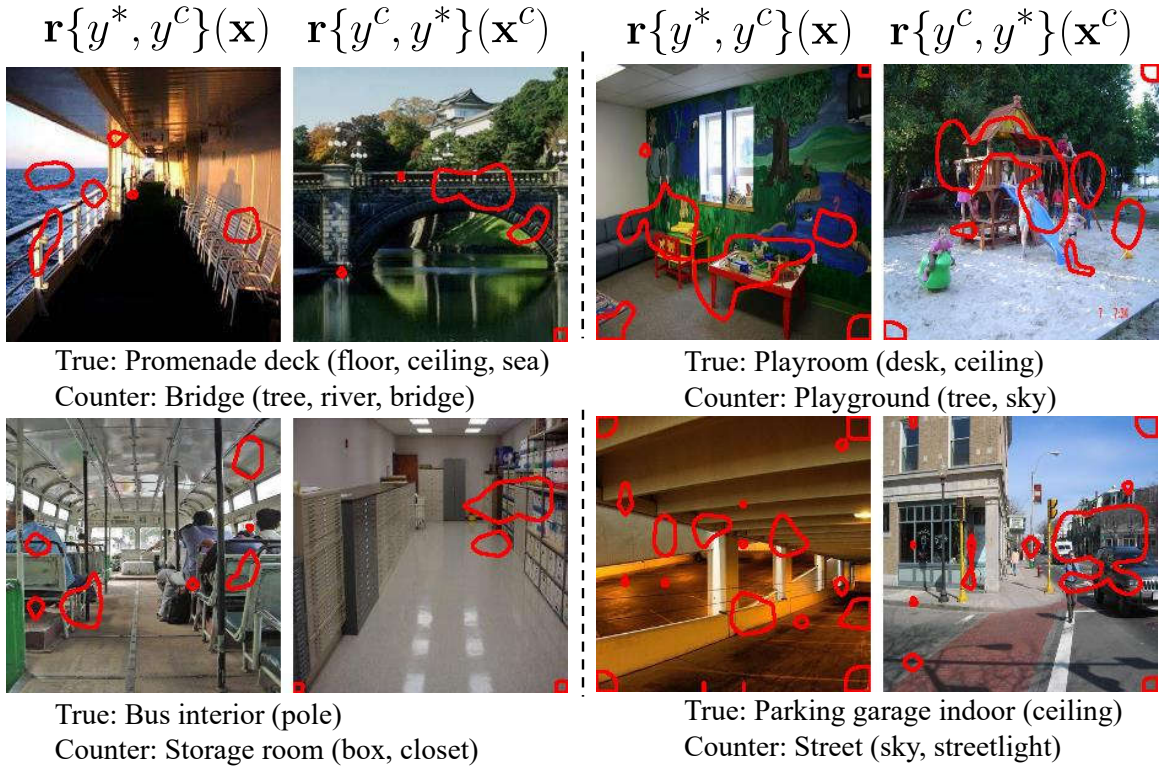
### 3.7.5 Comparison to State of the Art

To the best of our knowledge there have been no previous attempts to produce deliberative explanations. Table 3.2 presents a counterfactual explanation comparison between GALORE, the method of [115] and the CounteRGAN [152], for the two user types considered in this work. For fair comparison, these experiments use the softmax score of (3.15), so that model sizes are equal for both [115] and the proposed approach. The size of the counterfactual region is the receptive field size of one unit ( $\frac{1}{14*14} \approx 0.005$  of image size for VGG16 and  $\frac{1}{7*7} \approx 0.02$  for ResNet-50). This is constrained by the speed of the algorithm of [115], where the counterfactual region is determined by exhaustive feature matching. For CounteRGAN, we guarantee the same region size by thresholding the residual outputs of the generator. In the table, Results are shown as mean(stddev). IPS stands for images per second, implemented on NVIDIA TITAN Xp. Results are omitted

**Table 3.2:** Comparison to the state of the art in counterfactual explanations.

Arch.	Metric	Beginner User				Advanced User			
		Goyal [115]	CounteRGAN [152]	GALORE	Goyal [115]	CounteRGAN [152]	GALORE		
VGG16	R	0.02 (0.01)	0.03 (0.00)	<b>0.05</b> (0.01)	<b>0.05</b> (0.00)	<b>0.05</b> (0.00)	<b>0.05</b> (0.00)		
	P	0.76 (0.01)	0.78 (0.00)	<b>0.84</b> (0.01)	0.56 (0.01)	0.61 (0.00)	<b>0.64</b> (0.01)		
	PIoU	0.13 (0.00)	0.13 (0.00)	<b>0.15</b> (0.00)	0.09 (0.00)	0.12 (0.00)	<b>0.14</b> (0.02)		
	IPS	0.02 (0.00)	-	<b>26.51</b> (0.71)					
ResNet-50	R	0.03 (0.01)	0.06 (0.00)	<b>0.09</b> (0.02)	0.12 (0.01)	<b>0.17</b> (0.00)	0.16 (0.00)		
	P	0.77 (0.01)	0.74 (0.01)	<b>0.81</b> (0.01)	0.57 (0.02)	0.56 (0.00)	<b>0.60</b> (0.01)		
	PIoU	0.18 (0.01)	<b>0.20</b> (0.00)	0.16 (0.01)	<b>0.15</b> (0.00)	0.14 (0.00)	<b>0.15</b> (0.01)		
	IPS	1.13 (0.07)	-	<b>78.54</b> (11.87)					

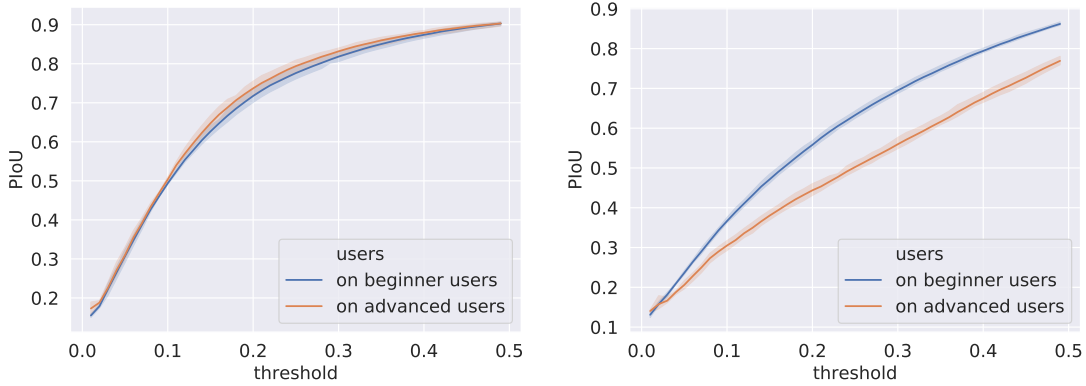




**Figure 3.13:** Counterfactual explanations by GALORE on ADE20K.

for the CounterGAN [152] due to the very long training times it requires.

Several conclusions can be drawn from the table. First, GALORE outperforms [115, 152] for almost all metrics. Second, GALORE is much faster, improving the speed of [115] by 1000+ times on VGG and 50+ times on ResNet. This is because it does not require exhaustive feature matching. These gains increase with the size of the counterfactual region, since computation time is constant for GALORE but exponential on region size for [115]. Third, due to the small size used in these experiments, PIoU is relatively low for all methods. It is, however, larger for GALORE explanations with large gains in some cases (VGG & advanced). Figure 3.14 shows that PIoU can raise to 0.5 for regions of 10% (VGG) or 20% (ResNet) of the image size. This suggests that, for such regions sizes, region pairs have matching semantics.



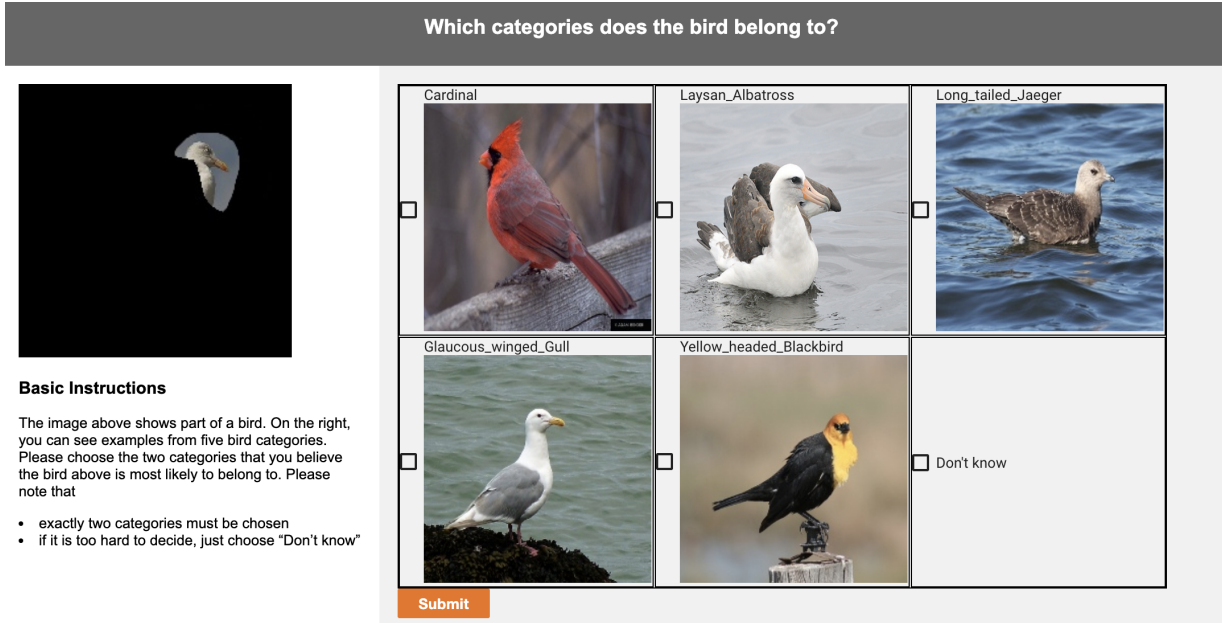
**Figure 3.14:** PIoU of proposed counterfactual explanations as a function of the segmentation threshold on CUB200. Left: VGG16, right: ResNet-50.

## 3.8 Human Studies

### 3.8.1 Insecurity Evaluation

Figure 3.15 shows the interface of the human experiment used to evaluate deliberative explanations on Amazon MTurk. The region of support of the uncertainty is shown on the left and examples from five classes are displayed on the right. These include the two ambiguous classes *a* and *b* found by the explanation algorithm, the “Laysan Albatross” and the “Glaucous Winged Gull”. The Turker is asked to select, among the five classes shown, the two to which the segment on the left is most likely to belong. If these two classes match the ambiguities found by the explanation algorithm the insecurity is considered intuitive. Otherwise, it is not. Turker performance was compared for insecurities generated by the explanation algorithm and randomly cropped regions of the same size.

Turkers agreed amongst themselves on classes *a* and *b* for 59.4% of the insecurities and 33.7% of randomly cropped regions. They agreed with the algorithm for 51.9% of the insecurities and 26.3% of the random crops. This shows that 1) insecurities are much more predictive of the ambiguities sensed by humans, and 2) the algorithm predicts those ambiguities with significant levels of consistency. In both cases, the “Don’t know” rate was



**Figure 3.15:** MTurk interface for human evaluation of deliberative explanations.

around 12%.

### 3.8.2 Application to Machine Teaching

In this section, we first show some preliminary results of applying GALORE to machine teaching. In Chapter 3, we will talk about how to combine GALORE with machine teaching in a more advanced manner.

Goyal *et al.* [115] used counterfactual explanations to design an experiment to teach humans distinguish two bird classes. During a training stage, learners are asked to classify birds. When they make a mistake, they are shown counterfactual feedback of the type of Figure 3.12, using the true class as  $y^*$  and the class they chose as  $y^c$ . This helps them understand why they chose the wrong label, and learn how to better distinguish the classes. In a test stage, learners are then asked to classify a bird without visual aids. Experiments reported in [115] show that this is much more effective than simply telling them whether their answer is correct/incorrect, or other simple training strategies. We





Kentucky Warbler

Setophaga Citrina

**Figure 3.16:** Visualization of machine teaching experiment.

made two modifications to this set-up. The first was to replace bounding boxes with highlighting of the counterfactual regions, as shown in Figure 3.16. We also instructed learners not to be distracted by the darkened regions. Unlike the set-up of [115], this guarantees that they do not exploit cues outside the counterfactual regions to learn bird differences. Second, to verify this, we added two contrast experiments where 1) highlighted regions are generated randomly (without telling the learners); 2) the entire images are lighted. If these produce the same results, one can conclude that the explanations do not promote learning.

We also chose two more difficult birds, the Setophaga Citrina and the Kentucky Warbler (see Figure 3.16), than those used in [115]. This is because these classes have large intra-class diversity. The two classes also cannot be distinguished by color alone, unlike those used in [115]. The experiment has three steps. The first is a pre-learning test, where humans are asked to classify 20 examples of the two classes, or choose a ‘Don’t know’ option. The second is a learning stage, where counterfactual explanations are provided for 10 bird pairs. The third is a post-learning test, where humans are asked to answer

20 binary classification questions. In this experiment, all students chose ‘Don’t know’ in the pre-learning test. However, after the learning step, they achieved 95% mean accuracy, compared to 60% (random highlighted regions) and 77% (entire images lighted) in the contrast settings. These results suggest that the proposed counterfactual explanations can help teach naive humans distinguish categories from an expert domain.

### 3.9 Conclusion

In this work, we have proposed a new framework, GALORE, for visualization-based explanations of deep neural networks predictions. GALORE unifies attributive, counterfactual, and deliberative explanations, aiming to satisfy the requirements of a diverse set of end-users. Attributive explanations visualize how different pixels contribute to a class prediction, deliberative explanations address the “why?” question, and counterfactual explanations the “why not?” question. All explanations are based on a combination of attributions with respect to class predictions and confidence scores. This makes them very efficient to compute, in some cases orders of magnitude faster than the state of the art. We have also introduced an experimental protocol to evaluate explanation accuracy, which sidesteps the difficulty of replicating user experiments. We believe this will facilitate research in the visualization based XAI problem. Both this protocol and human experiments were used to evaluate GALORE on two fine-grained datasets, demonstrating that the explanation results are more accurate than previously possible, intuitive, and correlate with human perception. In this process, we have also validated the importance of self-awareness both to define different explanations and to increase their accuracy. The counterfactual explanation results have shown to be beneficial for machine teaching.

Chapter 3 is, in full, based on the materials as they appear in the publication of “Deliberative Explanations: visualizing network insecurities”, Pei Wang, Nuno Vasconcelos,

In *Advances of Neural Information Processing Systems* (NeurIPS), 2019, and “SCOUT: Self-aware Discriminant Counterfactual Explanations”, Pei Wang, Nuno Vasconcelos, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2020, as well as the material as it appears in the submission of “A Generalized Explanation Framework for Visualization of Deep Learning Model Predictions”, Pei Wang, Nuno Vasconcelos, In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (TPAMI). The dissertation author was the primary investigator and author of these papers.

## Chapter 4

# A Machine Teaching Framework for Scalable Recognition

## 4.1 Introduction

The success of deep learning in computer vision has been largely driven by large-scale datasets. Many breakthroughs, made across various tasks, have benefited from large-scale and well-curated datasets like ImageNet for object recognition [14], COCO for object detection and segmentation [18], Kinetics for action recognition [15], etc. These datasets usually contain common objects, scenes, or actions and thus can be scalably annotated on crowdsourcing platforms such as Amazon Mechanical Turk (MTurk) [19]. When this is possible, we say that *learning is scalable*. However, this is usually not the case for expert domains, such as biology or medical imaging. While data collection can still be easy in these domains, annotations require highly specialized and domain-specific knowledge. For example, while it is easy to crawl the web or deploy cameras in the wild to collect a large number of animal images, it is usually expensive to recruit the biologists or taxonomists needed to label them. The resulting lack of large annotated datasets hampers the application of deep learning to expert domains. For example, the largest existing bird dataset, NAbirds, only contains about 48k instances [189]. Even the recent and largest biological dataset, iNaturalist, contains only about 850k instances [85]. This is smaller than ImageNet, proposed about 10 years ago, and pales in comparison to the largest datasets of everyday objects, e.g. Open Images with 9M images [190].

Since labeling is difficult in expert fine-grained domains, scalable learning must take advantage of small expert-labeled datasets and large amounts of unlabeled data. This motivated extensive research on less label-intensive forms of learning, including few-shot learning, transfer learning, semi-supervised learning, and self-supervised learning. For example, models pre-trained on an everyday domain by supervised learning are frequently transferred to a target fine-grained domain by fine-tuning. Another strategy is to learn a good feature extractor by self-supervised learning, which requires no labels, and then fine-tune a classifier at the top of it on a small set of labeled target data. However, these

approaches usually underperform scalable supervised learning. For example, state-of-the-art self-supervised learning with SimCLR [23] underperforms a supervised baseline when only a subset of the samples are labeled, especially on fine-grained domains [30].

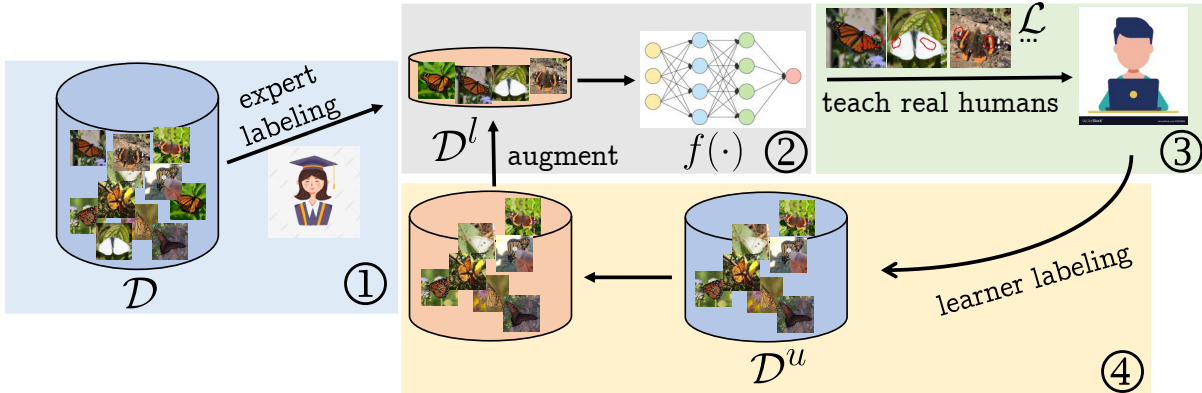
Unlike all these approaches, we pursue the alternative solution of scaling up the process of *data annotation*. While this was a pie in the sky idea in the past, two recent developments now make it promising. First, several crowdsourcing platforms, like Amazon Mechanical Turk, Sama [191], microWorkers [36], or Clickworker [37], have appeared in recent years, making it easier to recruit large numbers of image annotators online. Second, research has been steadily increasing in the area of machine teaching [51, 192, 59], showing potential to develop algorithms capable of teaching these annotators the domain-specific knowledge needed to label expert data. While these developments are promising, there have been so far no efforts to study how they can be combined into a complete framework for scalable learning. Typically, machine teaching papers only evaluate the accuracy of the labeling produced by the annotators taught by their algorithms. While this is informative, it does not fully address the scalable learning problem, which also includes the design of deep learning systems using those annotations. This raises an additional set of questions, such as what quality must the labels have to guarantee effective deep learning performance, how can the machine teaching algorithms achieve that quality, and whether noisy label learning algorithms [193, 194] have a role in the process.

In this work, we address these questions in the context of scalable learning of recognition systems, which we denote as *scalable recognition*. We propose a new *Machine teaching fraMewORk for scAlaBLe rEcognition* (MEMORABLE) in fine-grained expert domains, illustrated in Figure 4.1. A large raw dataset ( $\mathcal{D}$ ) is first collected for a target fine-grained task, e.g. by deploying cameras in the wild or crawling archived medical images in a hospital database. A small subset  $\mathcal{D}^l \subset \mathcal{D}$  and  $|\mathcal{D}^l| \ll |\mathcal{D}|$  is then labeled by experts. Machine teaching is next used to teach non-experts, e.g. Amazon MTurk workers, how

to label for the target categories. The unlabeled data  $\mathcal{D}^u = \mathcal{D}/\mathcal{D}^l$  is finally labeled by these humans and the complete dataset used to train an image recognition system. To identify critical areas of this framework, we perform an initial study with simulated noisy annotations. This shows that the accuracy of the machine teaching plays a significant role in the accuracy of the final recognition system. We then hypothesize that better machine teaching performance can be achieved by introducing explanations in the machine teaching algorithm. State-of-the-art machine teaching algorithms [56, 57, 61] tend not to use explanations. Although there is literature doing [59], it tends to rely on attributive explanations [103, 102] that do not take into account the student predictions. To address this problem, we propose the addition of counterfactual explanations to machine teaching.

Counterfactual explanations [195, 196] take into account both ground-truth labels and student predictions, highlighting image regions that are most discriminant of student mistakes. They are thus most instructive for humans to learn from their errors. Furthermore, because the explanatory feedback varies according to the student’s prediction, they naturally adjust to the level of competence of the student. We seek to leverage all these benefits by introducing a generalization of the recent MaxGrad machine teaching algorithm [197], denoted *Counterfactual MaxGrad* (CMaxGrad), which is endowed with counterfactual explanations. Experiments show that this algorithm both achieves state-of-the-art machine teaching performance and enables significant scalable recognition gains for the MEMORABLE framework. The latter is itself shown to outperform other scalable recognition strategies, such as semi-supervised learning. It is also shown that deep learning systems trained with MEMORABLE can leverage noisy label training schemes with surprising effectiveness.

The contributions of the work are summarized as 1) a study of the importance of labeling accuracy for the accuracy of scalable recognition; 2) the MEMORABLE framework to solve the fine-grained scalable recognition problem, by leveraging crowdsourcing platforms



**Figure 4.1:** The proposed MEMORABLE framework for large-scale recognition in fine-grained domains.

and machine teaching algorithms; 3) the new CMaxGrad machine teaching algorithm that introduces counterfactual explanations into machine teaching; and 4) new benchmarks, based on two challenging datasets, for the evaluation of scalable recognition.

## 4.2 Related Work

**Crowdsourcing platforms** There are two types of crowd sourcing platforms. They provide expert and non-expert annotation services. Amazon Mechanical Turk [19] is a widely known and representative one. It has been making it easy to require simple annotation tasks of significantly huge size to a large pool of workers. Although Amazon Turk has been broadly used, most of the workers are non-expert for a specific target expertise task like fine-grained annotation. For example, they can help annotate “dog” and “cat”, but hard to do “California Gull” and “Western gull”. The lack of prior knowledge of a specific domain makes it hard to satisfy the requirement of fine-grained expert domain labeling. The similar platforms include Sama [191], microWorkers [36], Clickworker [37], etc. They all provide similar services just with slight differences. A comprehensive discussion of them can be found in [198].



Another type of crowdsourcing platform can give expertise annotation service. Citizen scientist is a typical one [189]. It is non-profit and people in this platform are non-professional scientists or enthusiasts in a particular domain. They contribute annotations with the understanding that their expertise, experience and passion in a domain of interest. Although it makes it feasible to do expert labeling, there are some problems. Because of non-profits, it is hard to guarantee the quality of their results and guarantee that they are all responsible. This is different from Amazon Turk where if the annotation results are assessed badly by the requester, the worker would not get the payment. The second problem is that the active user number is small, especially on some minor domains. So it is hard to meet the large-scale annotation requirement. In this work, we use Amazon Turk, but unlike the common usage, a short course is introduced preceding the annotation. The worker is trained first and then annotates. This alleviates the problems of both types.

**Semi-supervised and self-supervised learning** Semi-supervised learning describes a class of algorithms that seek to learn from both unlabeled and labeled samples, typically assumed to be sampled from the same or similar distributions. Limited to the space, we refer to [199] for an extensive survey and [27] for up-to-date development.

Self-supervised learning (SSL) refers to learning methods in which the model is explicitly trained with supervisory signals that are generated from the data itself by leveraging some pretext tasks. The pretext tasks can be predictive tasks, generative tasks, contrasting tasks, or a combination of them. SSL can benefit almost all types of downstream tasks, e.g. semi-supervised learning, that can also be used to evaluate the quality of features learned by self-supervised learning [23, 200, 201]. Literature [202, 203, 204] is recommended for an extensive overview.

**Counterfactual explanations** Given an image of class  $A$  and a user-specified counterfactual class  $B$ , counterfactual explanations produce an explanation to answer “why the prediction is  $A$  but not  $B$ ” [145, 146, 147, 125, 142]. In computer vision, the explanations

are usually given by visualizations. Two main approaches to these explanations have emerged. The first group is based on an image transformation that elicits the classification as  $B$  [145, 146, 147]. The simplest example is adversarial attack [108, 145], which optimize perturbations to map an image of class  $A$  into class  $B$ . However, adversarial perturbations usually push the perturbed image outside the boundaries of the space of natural images. A more plausible alternative is to exhaustively search the space of features extracted from a large collection of images, to find replacement features that map the image from class  $A$  to  $B$  [196]. However, exhaustive search is too complex for interactive applications. Another form is optimization-free but produces a pair of segments on two images from ground truth class and counterfactual class [195]. These segments cover the class-discriminant regions. Its generation is much faster and we use it in our work.

**Machine teaching** Machine teaching is a broad area. The goal is to select a small number of data from a large set so that this small set can efficiently teach a student. The student can be either a network model or a real human. Because this work mainly talks about the latter, we recommend [51, 192] for the reader about the network-oriented machine teaching. For real-human machine teaching, a typical strategy is to first model humans as a network model and then select a teaching sequence universally used for human teaching. In this process, most of the previous literature simulates human students based on the assumption that they have limited capacity or are otherwise sub-optimal learners [55, 61, 58]. This is intuitive but not optimal in the crowdsourcing context, which has been discussed in [197]. The latter is subject to an optimal student assumption that the students will try their best to complete the assigned tasks. Another direction of real-human machine teaching is to think about how to incorporate the explanation into the teaching process because it is straightforward that explanations are helpful for digesting the knowledge easily [59, 60, 80]. A representative work [59] merges the attribution map into the example selection and feedback stage of teaching. When the learner makes a mistake,

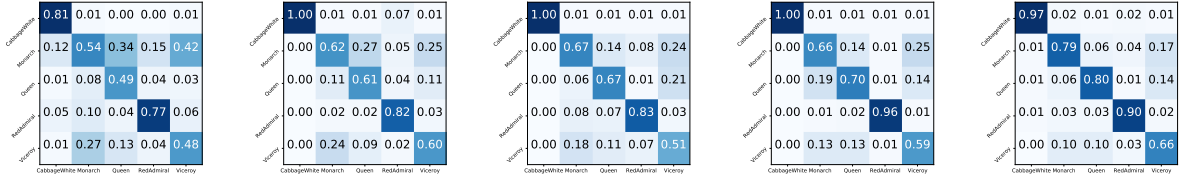
a heatmap [102] that highlights the regions that contribute to the correct class is shown. This, to a certain extent, provides some explanations but can not adapt to the learner’s choice. Counterfactual explanations were simply associated with random selected images to evaluate their qualities in [195, 196], but there is no special machine teaching algorithm involved and the evaluation is only on simple binary classification tasks. Tropel [205] lets workers identify positive/negative images with respect to a given query image, to train a detector. This is unlike a counterfactual explanation for teaching, where the counter class is an incorrect label chosen by the worker. The latter more directly provides the worker with feedback regarding mistakes. Also, there is no image-based explanation in Tropel. In this work, we attempt to include the counterfactual explanation into the machine teaching, an explanation that explicitly indicates the class-discriminant between correct class and mis-chosen class. The experiments show that this is more helpful.

## 4.3 The MEMORABLE Framework

In this section we introduce the MEMORABLE framework.

### 4.3.1 Machine Teaching

We consider the problem of  $C$ -class classification on expert domains where data collection is easy but annotation is difficult. For example, while biologists routinely deploy camera traps in the wild [206] or underwater [207], the labeling of the resulting images by professional taxonomists is quite expensive. The goal is to train classifiers from large datasets, i.e. scalable recognition. A practical solution is semi-supervised learning. A large set of images  $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{M+N}$  is first collected and a small subset  $\mathcal{D}^a = \{\mathbf{x}_i\}_{i=1}^M$ , where  $M \ll N$  labeled by experts. This results in a labeled dataset  $\mathcal{D}^l = \{(\mathbf{x}_i, y_i)\}_{i=1}^M$  where  $y_i$  is the label of  $\mathbf{x}_i$ . A classifier  $f$  is then learned from the semi-supervised dataset  $\mathcal{D}^s = \mathcal{D}^u \cup \mathcal{D}^l$ ,



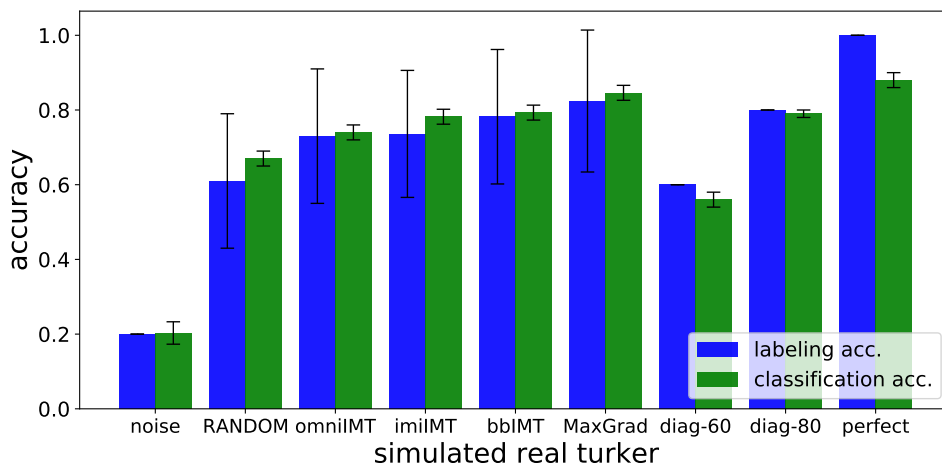
(a) RANDOM [59] (b) omniIMT [56] (c) imiIMT [56] (d) bbIMT [57] (e) MaxGrad [197]

**Figure 4.2:** Confusion matrices for human annotators trained by different machine teaching algorithms on Butterflies dataset.

where  $\mathcal{D}^u = \mathcal{D} - \mathcal{D}^l$  is the set of unlabeled images. The performance of  $f$  is finally evaluated on a testing set  $\mathcal{T}$ . While various semi-supervised learning algorithms exist [27, 199], their performance is frequently inferior to supervised learning. This gap can be bridged by labeling the data  $\mathcal{D}^u$  on a crowd-sourcing platform, such as Amazon MTurk. This, however, is impossible for data from domains, e.g. animal taxonomies, on which MTurk annotators have no expertise.

MEMORABLE addresses this problem by leveraging the labeled dataset  $\mathcal{D}^l$  to *teach* MTurk annotators to label the images in  $\mathcal{D}^u$ . As shown in Figure 4.1, this is done in several steps. A classifier  $f$  is first trained, either by semi-supervised learning on  $\mathcal{D}^l \cup \mathcal{D}^u$ , or supervised learning on  $\mathcal{D}^l$ . This classifier is then leveraged to design a *teaching set*  $\mathcal{L} \subset \mathcal{D}^l$  of  $L \ll M$  images for training MTurk annotators. Several machine teaching algorithms have been proposed to extract an optimal teaching set from  $\mathcal{D}^l$  [59, 55, 197]. Finally, MTurk annotators are trained by practicing on the teaching set  $\mathcal{L}$ . This usually consists of an introductory step where they are shown one (or a few) images of each class, and an iterative step where they attempt to classify images in  $\mathcal{L}$  and receive feedback on their mistakes. When this process is completed, the trained MTurkers are finally asked to label  $\mathcal{D}^u$  and the classifier is retrained.

While various works have addressed individual components of this framework, e.g. by proposing different machine teaching algorithms [59, 55, 197] or semi-supervised learning techniques [27, 199], we are aware of no studies on the effectiveness of the entire scalable



**Figure 4.3:** Labeling and classification accuracies of simulated turkers.

recognition architecture. Two questions, in particular, seem quite relevant. First, how does the accuracy of the trained MTurkers affect the accuracy of scalable recognition? Second, how can machine teaching algorithms be enhanced to improve MTurker accuracy?

### 4.3.2 How Important is Annotator Accuracy?

Since the training of MTurkers is not perfect, labels can be noisy. In general, the human-labeled dataset  $\mathcal{D}^u$  is noisier than if labeled by experts. This begs the question of how accurate must the trained MTurkers be for machine teaching to be useful. To determine this, we perform a set of experiments with simulated “noisy MTurkers.” Given an unlabeled dataset  $\mathcal{D}^u$ , for which the ground-truth labels  $Y$  are known to us but unavailable to the algorithms, we assign to each image a noisy label  $Y'$ , according to a confusion matrix  $\mathbf{M}$ , where  $m_{ij} = P(Y' = i | Y = j)$ . More precisely, given ground truth label  $y = j$ , a class label  $y'$  is sampled from the distribution  $[m_{1j}, \dots, m_{Cj}]$ .

The resulting noisy labeled dataset  $\mathcal{D}^n$  is used to train a classifier  $f$ . By comparing the accuracy of  $f$  to that of a classifier  $g$  trained on the ground truth dataset, it is possible to determine the effect of MTurker annotation noise on the final classification performance.

By varying the matrix  $\mathbf{M}$ , it is possible to analyze how the latter depends on the quality of the annotators. To enable these comparisons, we propose two metrics. The first is the **labeling accuracy**

$$\text{ACC}^l = \frac{\sum_{i=1}^C m_{ii}}{\sum_{i=1}^C \sum_{j=1}^C m_{ij}}. \quad (4.1)$$

This is a number in  $[0, 1]$ , equal to 1 when there is no labeling noise. The second is the **classification accuracy**, measured by average accuracy on the testing set  $\mathcal{T}$  of classifiers  $f$  trained on  $\mathcal{D}^l \cup \mathcal{D}^n$ .

To investigate the effects of the confusion matrix  $\mathbf{M}$  on classifier accuracy, we considered nine different matrices. The first five were estimated from real MTurker data. Annotators were trained with several machine teaching algorithms from the literature, chosen to reflect the spectrum of training effectiveness. The weakest performance was implemented with the RANDOM [59, 197] procedure, where annotators are taught with a randomly chosen teaching set  $\mathcal{L}$ . Stronger performances were implemented with omniIMT [56], imiIMT [56], bbIMT [57] as well as the state-of-the-art MaxGrad machine teaching algorithm [197]. As can be seen in Figure 4.2, the latter four produce much more accurate annotators than the former. The next four matrices are hand-crafted models of annotator quality. The first is a “chance level” annotator, i.e.  $m_{ij} = 1/C, \forall i, j$ . The next two are models that mimic matrices estimated on MTurk. They are denoted as diag-60 and diag-80, and have diagonal elements of 0.6, 0.8, respectively, and uniform non-diagonal values. diag-60 approximates RANDOM and diag-80 approximates MaxGrad. The final model is a perfect annotator with a diagonal matrix  $\mathbf{M}$  of entries 1.

Figure 4.3 summarizes the result of this experiment, enabling several interesting observations. First, the labeling accuracies of diag-60 and diag-80 do match those of RANDOM and MaxGrad, respectively. However, the same does not hold for the associated classification accuracies. In fact, one of the most interesting observations of the figure is how the hand-crafted matrices have much weaker classification accuracy than those learned

from MTurker data. In particular, the classification accuracy is always higher than the labeling accuracy for the MTurk matrices, but the reverse holds for their models.

A closer inspection of the confusion matrices shows that those estimated from human annotators do not have a uniform distribution for the annotation errors. While the diagonal value may not be 1, there is usually a dominant class for mistakes, i.e. the second probability tends to be larger than the remaining. This is likely to simplify the learning of the classifier, since it is mostly faced with label noise between pairs of classes, rather than all. The ensuing insight is that, beyond errors, it also matters what type of errors are made by the annotators. Informative labeling errors, between a few classes, lead to much better classifiers than uninformative, uniformly distributed, ones. Note that the differences in classification accuracy are substantial, with the MTurk-trained classifiers outperforming the model-trained classifiers by 5 – 10%.

Having said this, a second observation is that the *accuracy of the machine teaching algorithm does matter*. For example, both MaxGrad and diag-80 produced better classifiers than RANDOM and all methods produced very large gains over the chance annotator. Comparing machine teaching algorithms, it is clear that recognition accuracy increases with labeling accuracy. Finally, it can be observed that there is an upper bound on the required annotator accuracy. In fact, the perfect annotator produces classifiers that are only marginally better than those of MaxGrad. This is quite interesting, suggesting that current machine teaching algorithms already are a viable solution for classifier training. We note, however, that this is an experiment based on five classes. For large  $C$ , the differences are likely to be more significant. This is left for future research.

### 4.3.3 The Role of Explanations

A machine teaching algorithm aims to select the teaching set  $\mathcal{L}$  from  $\mathcal{D}^l$  that maximizes student labeling accuracy. Traditional algorithms [55, 56] present the images

in  $\mathcal{L}$  to the student, displaying the ground truth label as feedback when the latter makes a mistake. While this can suffice for coarse-grained classification, it is not ideal for most expert domains, where classification tends to be fine-grained. In this case, the differences between categories can be imperceptible to the untrained eye. Without further hints, it can be quite hard for non-experts to learn the target concepts. [59] addressed the problem with the EXPLAIN algorithm, which introduced attributive explanations into machine teaching. These are explanations based on a saliency map that highlights regions contributing to the classifier prediction [103, 102]. By directing student attention to features important for the classification, these explanations can enhance teaching. However, more recent methods, such as bbIMT [57], imiIMT [56], or MaxGrad [197] achieve better results than EXPLAIN without explanations.

In this work, we seek to add explanations to the state-of-the-art MaxGrad algorithm [197]. We note, however, that a limitation of attributive explanations, such as those of EXPLAIN, is the lack of user-specific interaction. At each teaching iteration, the feedback provided by these explanations is always the correct label and the corresponding attribution map. Since the class predicted by the student is not considered in the explanation, the latter does not necessarily address the student’s difficulties. Better feedback should take the student prediction into account. This is the definition of counterfactual explanations [195, 196], which address the question: “why is the class predicted by the student incorrect?” We next introduce an enhanced version of MaxGrad that leverages counterfactual explanations.

## 4.4 Counterfactual MaxGrad (CMaxGrad)

Counterfactual explanations can provide detailed student feedback during the retraining step when, given the query image  $\mathbf{x}^t$  of ground-truth label  $y^t$ , the student predicts



a counterfactual class  $y^c \neq y^t$ . An example is shown in Figure 4.4 for the Butterflies dataset, where  $y^t = \text{'Viceroy'}$  and  $y^c = \text{'Monarch'}$ . The explanation first samples an image  $\mathbf{x}^c$  from  $y^c$ , and then produces a visualization of the form: “The correct label is  $y^t$ . If the correct label were  $y^c$ , the circled region of  $\mathbf{x}^t$  should look like the circled region of  $x^c$ .” Mathematically, this reduces to a function

$$\mathcal{C}(\mathbf{x}^t, y^t, y^c, \mathbf{x}^c) = (\mathbf{c}^c(\mathbf{x}^t), \mathbf{c}^t(\mathbf{x}^c)), \quad (4.2)$$

where  $\mathbf{c}^c(\mathbf{x}^t)$  and  $\mathbf{c}^t(\mathbf{x}^c)$  are counterfactual heatmaps or segments for images  $\mathbf{x}^t$  and  $\mathbf{x}^c$  respectively. They highlight image regions of features discriminant for the two classes. In Figure 4.4, these are the presence/absence of a line that crosses the radial wing lines of the two butterflies, and the different configurations of white spots. This explanation allows the student to quickly learn what to look for in order to distinguish the two classes. Since the counterfactual class was selected by the student, the process quickly provides the student with *precise* feedback on how to differentiate between the classes that most *confuse* them.

To include counterfactual explanations on MaxGrad, we propose the following generalization.

1. counterfactual maps are generated for all pairs of queries and counterfactual examples in the labeled dataset  $\mathcal{D}^l$ . This results in the explanation set  $\mathcal{E} = \{\mathbf{c}^{y^c}(\mathbf{x}_i) | y^c \neq y_i\}_{i=1, c=1}^{M, C}$ . This is a pre-processing step, performed before machine teaching takes place.
2. teaching set  $\mathcal{L}^t$  is augmented with a *counterfactual set*  $\mathcal{C}^t$  that includes counterfactual images and heatmaps.
3. during training, at iteration  $t$  the teacher selects an image  $\mathbf{x}^t$  from  $\mathcal{D}^l - \mathcal{L}^{t-1}$ . The student then makes a prediction  $y = f^t(\mathbf{x}^t)$ . For the reasons discussed below, this is

---

**Algorithm 2 CMaxGrad**


---

**Input** Data  $\mathcal{D}^l = \{(\mathbf{x}_i, y_i)\}_{i=1}^M$ , max iter.  $T$ ,  $\alpha$  and  $\beta$ ,  $\mathcal{E} = \{\mathbf{c}^{y^c}(\mathbf{x}_i) | y^c \neq y_i\}_{i=1, c=1}^{M, C}$ .

- 1: **Initialization:**  $\mathcal{L}^0 \leftarrow \emptyset$ ,  $\mathcal{C}^0 \leftarrow \emptyset$ ,  $f^1$ ,  $\mathcal{D}^0 \leftarrow \mathcal{D}^l$ ,  $\mathcal{E}^0 \leftarrow \mathcal{E}$
- 2: **for**  $t = \{1, \dots, T\}$  **do**
- 3:   compute  $\xi(\mathbf{x}_i)$  for all examples in  $\mathcal{D}^{t-1}$  and  $\xi(\mathbf{c}^{f^t(\mathbf{x}_i)}(\mathbf{x}_i))$  for all examples in  $\mathcal{E}^{t-1}$
- 4:   select  $\mathbf{x}^t = \arg \max_{\{\mathbf{x}_i \in \mathcal{D}^l - \mathcal{L}^{t-1}\}} \xi_c(\mathbf{x}_i, \mathbf{c}^{f^t(\mathbf{x}_i)}(\mathbf{x}_i); \alpha)$
- 5:   select  $\mathbf{x}^{t,c} = \arg \max_{\{\mathbf{x}_i \in \mathcal{D}^l - \mathcal{L}^{t-1} | y_i = f^t(\mathbf{x}^t)\}} \xi_c(\mathbf{x}_i, \mathbf{c}^{y^t}(\mathbf{x}_i); \beta)$
- 6:   teaching and explanation sets update:  $\mathcal{L}^t \leftarrow \mathcal{L}^{t-1} \cup \{\mathbf{x}^t\}$ ,  $\mathcal{C}^t \leftarrow \mathcal{C}^{t-1} \cup \{\mathbf{x}^{t,c}, \mathbf{c}^{f^t(\mathbf{x}^t)}(\mathbf{x}^t), \mathbf{c}^{y^t}(\mathbf{x}^{t,c})\}$
- 7:   student update:  $f^{t+1} = f^*(\mathcal{L}^t \cup \mathcal{C}^t)$
- 8:    $\mathcal{D}^t \leftarrow \mathcal{D}^{t-1} \setminus \{\mathbf{x}^t, \mathbf{x}^{t,c}\}$ ,  $\mathcal{E}^t \leftarrow \mathcal{E}^{t-1} \setminus \{\mathbf{c}^{f^t(\mathbf{x}^t)}(\mathbf{x}^t), \mathbf{c}^{y^t}(\mathbf{x}^{t,c})\}$
- 9: **end for**

**Output**  $\mathcal{L}^t$

---

always incorrect, i.e.  $y = y^c \neq y^t$ , A counterfactual image  $\mathbf{x}^{t,c}$  is selected from class  $y^c$  and the counterfactual maps  $(\mathbf{c}^c(\mathbf{x}^t), \mathbf{c}^t(\mathbf{x}^{t,c}))$  are retrieved from  $\mathcal{E}$ . The teaching set is then augmented into  $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \{\mathbf{x}^t\}$  and the counterfactual set into  $\mathcal{C}^t = \mathcal{C}^{t-1} \cup \{\mathbf{x}^{t,c}, \mathbf{c}^c(\mathbf{x}^t), \mathbf{c}^t(\mathbf{x}^{t,c})\}$ . The student is finally updated with  $f^{t+1} = f^*(\mathcal{L}^t \cup \mathcal{C}^t)$ .

In MaxGrad, the image  $\mathbf{x}^t$  selected by the teacher is the one that maximizes a score  $\xi(\mathbf{x})$  representative of the classification difficulty posed by image  $\mathbf{x}$  to the student model  $f^t$ . Since this score is the negative classification margin  $\xi(\mathbf{x})$  of the image  $\mathbf{x}$  under  $f^t$ , there is always at least one image that the student cannot classify correctly in  $\mathcal{D}^l - \mathcal{L}^{t-1}$  (otherwise the training would be complete). Hence, the resulting student prediction is incorrect, i.e. a counterfactual class  $y^{t,c}$ .

However, in the counterfactual setting, image selection must also account for the counterfactual heatmaps  $(\mathbf{c}^c(\mathbf{x}^t), \mathbf{c}^t(\mathbf{x}^{t,c}))$ . For this, we propose a *counterfactual margin score*

$$\xi_c(\mathbf{x}, \mathbf{c}^y(\mathbf{x}); \alpha) = \alpha \xi(\mathbf{x}) + (1 - \alpha) \xi(\mathbf{c}^y(\mathbf{x})), \quad (4.3)$$

where  $\alpha \in [0, 1]$  is a hyperparameter that weighs the contribution of images and counter-

factual regions. Note that this supports scores based on the margin of the whole image ( $\alpha = 1$ ), the counterfactual region ( $\alpha = 0$ ) or both. This leads to the following procedure for the selection of the image  $\mathbf{x}^t$  to augment the teaching set. For each image  $\mathbf{x}_i \in \mathcal{D}^l - \mathcal{L}^{t-1}$ , the counterfactual class is identified as  $f^t(\mathbf{x}_i)$  and the heatmap  $\mathbf{c}^{f^t(\mathbf{x}_i)}(\mathbf{x}_i)$  retrieved from  $\mathcal{E}$ . The teacher then selects the image of largest score, i.e.

$$\mathbf{x}^t = \underset{\{\mathbf{x}_i \in \mathcal{D}^l - \mathcal{L}^{t-1}\}}{\operatorname{arg\,max}} \quad \xi_c(\mathbf{x}_i, \mathbf{c}^{f^t(\mathbf{x}_i)}(\mathbf{x}_i); \alpha), \quad (4.4)$$

to add to the teaching set  $\mathcal{L}^{t-1}$ .

The image  $\mathbf{x}^{t,c}$  of the counterfactual class  $y^{t,c}$  is then chosen with the same criterion among the images in the counterfactual class, i.e.

$$\mathbf{x}^{t,c} = \underset{\{\mathbf{x}_i \in \mathcal{D}^l - \mathcal{L}^{t-1} | y_i = f^t(\mathbf{x}^t)\}}{\operatorname{arg\,max}} \quad \xi_c(\mathbf{x}_i, \mathbf{c}^{y^t}(\mathbf{x}_i); \beta), \quad (4.5)$$

where  $y^t$  is the label of  $\mathbf{x}^t$ . The teaching set  $\mathcal{L}^t$  and the counterfactual set  $\mathcal{C}^t$  are then updated with  $\mathbf{x}^t$  and  $(\mathbf{x}^{t,c}, \mathbf{c}^{f^t(\mathbf{x}^t)}(\mathbf{x}^t), \mathbf{c}^{y^t}(\mathbf{x}^{t,c}))$ , respectively, and the student updated with  $f^{t+1} = f^*(\mathcal{L}^t \cup \mathcal{C}^t)$ . This requires training a classifier with both images and image regions, derived from the counterfactual heatmaps. In our implementation, counterfactual regions are converted to images by simply thresholding the heatmaps and setting the pixels outside the counterfactual region to the average image color. The resulting images are then added to  $\mathcal{C}^t$ . We note, however, that this is not done on human teaching experiments, where subjects are shown whole images, as demonstrated in Figure 4.4. The overall procedure is summarized in Algorithm 2 and denoted CMaxGrad.

## 4.5 Evaluation of Student Teaching

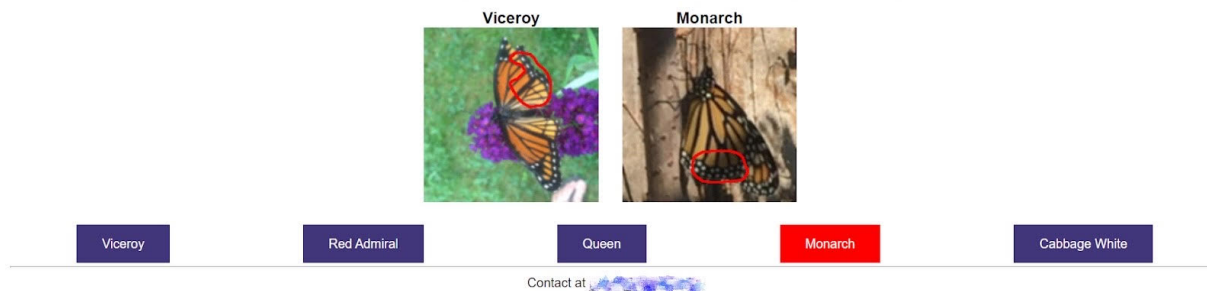
We start by evaluating the accuracy of the labels produced by students trained with CMaxGrad. Following [59], we consider both simulated and real students. Note that in this section, we do not talk about the scalable learning problem, and only focus on the evaluation of CMaxGrad by existing common protocol.

**Dataset** We used two recent machine teaching benchmark datasets: Butterflies and Chinese Characters [59]. These are more challenging than binary classification or synthetic datasets used in earlier work [55, 60, 80], because they are both fine-grained multi-class datasets of real images from expert domains. Both datasets have large intra-class diversity, e.g. due to different handwriting styles, and large inter-class similarity. Butterflies has five butterfly species sampled from iNaturalist [85], with 1544 training and 386 testing samples. Chinese Characters consists of three similar Chinese characters, with 568 training and 143 testing examples. These training-testing split on both cases follows [59]. The data is accessible in [87]. Both datasets were subject to standard normalizations. Training images were first randomly resized to  $224 \times 224$  and then randomly flipped, whereas testing images were first resized to  $256 \times 256$  and then center-cropped to  $224 \times 224$ . All images were also first converted to  $[0.0, 1.0]$  from  $[0, 255]$  and then normalized by subtracting the mean  $[0.485, 0.456, 0.406]$  and dividing by the standard deviation  $[0.229, 0.224, 0.225]$  of each RGB color channel. The teaching set is selected from the training set and the method is evaluated on the testing set.

**Network** The same as [59], the pre-trained ResNet-18 [1] on ImageNet is used to simulate the student. This is equivalent to assuming a student that starts from a good generic understanding of image classification. The student learners are trained 10 epochs by gradient descent with batch size equal to  $|\mathcal{L}^t|$  and weight decay of  $1e - 4$ . The learning rate is set to  $1e - 4$  with 0.9 momentum. Counterfactual explanations are generated by a ResNet-18 pre-trained in ImageNet and fine-tuned on the target training set.

Questions Answered: 8 out of 20

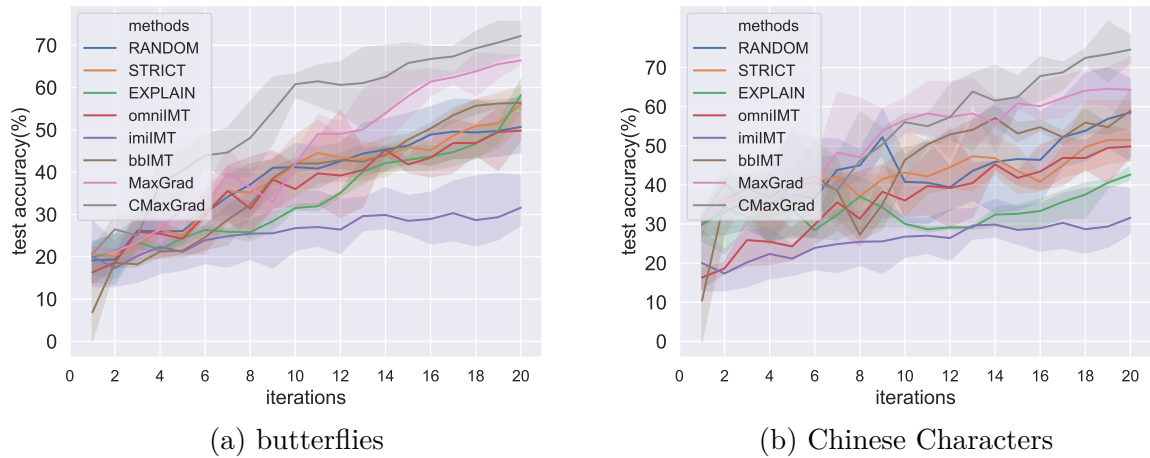
You selected Monarch. The correct answer is Viceroy. A hint might be helpful: The right image is a Monarch. If the left image is Monarch, the circled region should look like the circled region of the right image.



**Figure 4.4:** Interface. When the teaching image is “Viceroy” but the worker selected “Monarch”, the shown feedback will be given.

**Teaching** For fair comparison with other methods [56, 55, 59, 57], novel sets of size  $\tau = 1$  were used in all experiments, i.e. a single example is selected per iteration. All experiments use a teaching set of 20 examples, selected from the training set and tested on the testing set. Counterfactual maps were generated with the recent SCOUT algorithm [195]. Counterfactual regions were extracted by setting the segment size parameter to 5% of the image area. The parameters  $\alpha, \beta$  of (4.4) and (4.5), respectively, were set to  $\alpha = \beta = 0.5$  after cross-validation.

The MTurk experiments basically followed the setting of [59, 55], using 40 workers per dataset. The teaching process consists of two phases, teaching and testing. Before teaching, workers were shown a brief introduction of the teaching set-up, illustrating how our web-based teaching interface works. In the teaching stage, they were shown a sequence of 20 images. At each iteration, they were asked to select a category from a list of candidate options (five for butterflies and three for Characters), and received feedback declaring their choice ‘Correct’ or ‘Incorrect,’ as well as the true class. For CMaxGrad experiments, counterfactual explanations were presented additionally as in Figure 4.4, when their choices



**Figure 4.5:** Test set accuracy of simulated students as a function of teaching iterations (teaching example number).

are incorrect. Upon this, learners had to wait for a minimum of 2 seconds before proceeding to the next iteration. After teaching, 20 randomly selected test images were assigned to each learner, who was asked to classify them. These random images were different per learner and no feedback was provided as they were classified. In real learner evaluation, we require that workers be masters to do our tasks. Additionally, we require non-Chinese speaker on Chinese Characters dataset experiments. Each turker is paid \$1 for the teaching task.

#### 4.5.1 On the Simulated Learners

Again, we start with evaluations on simulated learners, i.e. a classifier. It can be seen that CMaxGrad significantly improves on MaxGrad further, especially on Butterflies. This suggests the importance of counterfactual explanations.

**Table 4.1:** Test set labeling accuracy, mean (std), of MTurkers. Methods with superscript “\*” represent our implementations. Values are presented by mean(std).

	Butterflies	Chinese Char.
RANDOM [59]	65.20	47.05
STRICT [55]	65.00	51.51
EXPLAIN [59]	68.33	65.44
omniIMT* [56]	70.07 (18.30)	64.36 (19.58)
imiIMT* [56]	72.70 (17.63)	64.46 (23.72)
bbIMT* [57]	76.09 (18.05)	64.37 (19.57)
MaxGrad	80.33 (19.76)	81.89 (12.93)
CMaxGrad	<b>84.10</b> (18.24)	<b>84.63</b> (20.18)

## 4.5.2 On the Real Learners

Table 4.1 reports the test accuracies of workers trained with different methods from the literature. Obviously, counterfactual explanations enabled a significant improvement in the accuracy of the MTurk student labels, even on a stronger baseline (MaxGrad). The improvement is even slightly higher than that of EXPLAIN on a weaker RANDOM baseline, on Butterflies. The latter used attributive explanations [103] to enhance the teaching. This in turn indicates that the counterfactual explanations are more suitable for attributive explanations.

## 4.6 Evaluation of Scalable Recognition

In this section, we evaluate the performance of the complete architecture of Figure 4.1.

**Dataset** Because there is no benchmark for the evaluation of scalable fine-grained recognition in expert domains, we created two such benchmarks. The first is based on the Butterflies dataset. The first 300 training samples (according to the dataset order<sup>1</sup>)

<sup>1</sup>[https://github.com/macaodha/explain\\_teach](https://github.com/macaodha/explain_teach)



**Figure 4.6:** Sample images of Gull dataset.

compose the expert labeled dataset  $\mathcal{D}^l$ , and the remaining 1,244 the unlabeled dataset  $\mathcal{D}^u$  to be annotated by Mturkers. The testing set is used for evaluation. The second benchmark is from an even more fine-grained and thus difficult task, based on the recognition of five gull categories: “California Gull”, “Glaucous winged Gull”, “Heermann Gull”, “Ring billed Gull” and “Western Gull”. An example image from each class is shown in Figure 4.6. These classes were chosen because they are the overlapping classes of two widely used bird datasets, CUB200 [178] and NAbirds [189]. The images from the CUB training set (150 instances) serve as expert-labeled dataset  $\mathcal{D}^l$  whereas those from NAbirds serve as unlabeled dataset  $\mathcal{D}^u$  (431 instances). The CUB testing set (149 instances) is used for evaluation.

**Network** A ResNet-18 is used as classifier. Explanations are generated by two models, each specific to one dataset. Because two of the butterfly categories are in ImageNet, the ResNet-18 is initialized from scratch for the Butterflies dataset. The Gull dataset has no overlap with ImageNet and is more challenging. Since the network trained from scratch on this dataset performs only slightly better than chance level ( $\approx 30\%$ ), the network is initialized with the model pre-trained on ImageNet.

**Platform** All experiments were conducted on Amazon Mechanical Turk. Each MTurker received a teaching set of 20 examples, chosen by MaxGrad or CMaxGrad, and was then requested to label 30 images randomly sampled from the unlabeled set. This



**Table 4.2:** Test accuracy comparison with mean (std). The lower group shows our results whereas the upper other literature.

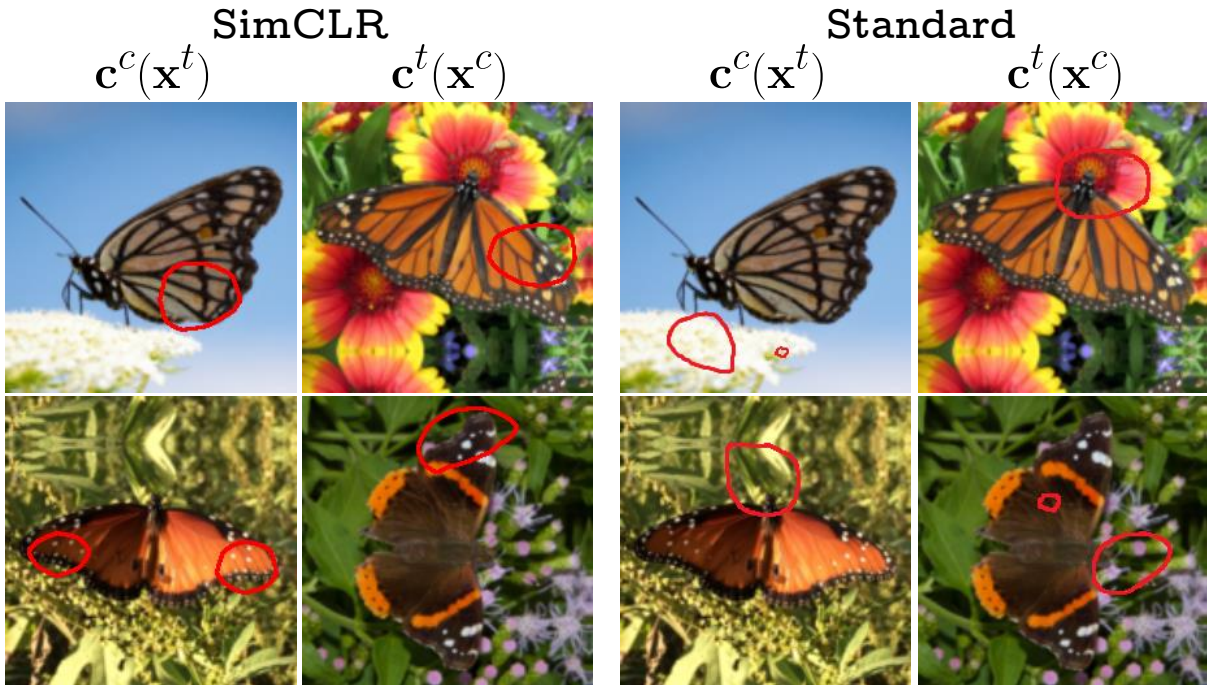
	Butterflies	Gull
Supervised baseline	59.4 (1.3)	58.3 (0.6)
Pseudo-Label [26]	64.7 (1.1)	58.7 (1.4)
SimCLR [23]	76.9 (0.4)	53.0 (0.8)
MaxGrad	74.7 (0.7)	56.3 (0.9)
CMaxGrad	77.5 (0.5)	60.2 (1.1)
CMaxGrad+SimCLR	78.2 (0.2)	59.7 (0.7)
MaxGrad+DivideMix	78.6 (1.2)	59.9 (1.2)
CMaxGrad+DivideMix	81.2 (1.1)	61.7 (1.5)
CMaxGrad+SimCLR+DivideMix	83.4 (0.7)	61.2 (1.1)

number was chosen so as to avoid the danger of worker fatigue and frustration possible with larger jobs. Three labelings were collected per example and their majority vote was chosen as the final label. If the labels were distinct, we chose one randomly.

**Baselines** MEMORABLE was compared to a number of scalable recognition baselines, whose results are shown in the top part of Table 4.2. “Supervised” refers to vanilla supervised learning on the expert-labeled dataset  $\mathcal{D}^l$ . Pseudo-Label [26], a semi-supervised learning method, first trains with supervision on  $\mathcal{D}^l$ , then iteratively improves the performance by self-labeling the unlabeled examples in  $\mathcal{D}^u$  and training on the pseudo labels. SimCLR [23] is a representative semi-supervised learning method. The feature extractor is first trained on  $\mathcal{D}^l \cup \mathcal{D}^u$  with contrastive loss and a top classifier is finetuned on  $\mathcal{D}^l$ .

#### 4.6.1 Comparison with the State of the Art

The second part of Table 4.2 presents results of MEMORABLE, using MaxGrad or CMaxGrad. Both obtain comparable or better results in general. When compared to MaxGrad, the counterfactual explanations produced by CMaxGrad enable substantial



**Figure 4.7:** Comparison of counterfactual explanations generated by different models. Two examples are shown. Top: true class is “Viceroy” and counter class “Monarch”; bottom: true class is “Queen” and counter class “Red Admiral”.

better classification accuracies, e.g. a gain of about 4% on Gull.

## 4.6.2 Enhancements

We next explore if MEMORABLE can benefit from semi-supervised training of the classifier and noisy label training.

**By training on unlabeled data** There is another strategy to train the classifier that produces the counterfactual explanations. Instead of supervised training the classifier on the expert-labeled dataset  $\mathcal{D}^l$ , semi-supervised learning on  $\mathcal{D}^l$  and  $\mathcal{D}^u$  is experimented. For the latter, we adopted the SimCLR [23] contrastive learning algorithm. This is denoted with “+SimCLR” in Table 4.2. On Butterflies, there is a 0.7% improvement but a few drop on Gull. This is consistent with the performance of the classifier. Figure 4.7 shows

examples of counterfactual regions selected by the two versions of CMaxGrad. While those produce with SimCLR cover body parts, the supervised model sometimes has difficulty localizing the class-discriminant regions, perhaps due to its lower classification accuracy.

**By Noisy label training** Since the labels produced by MTurkers are noisy, further performance improvements can in principle be accrued by training the final classifier with noisy label learning algorithms [193, 194, 208]. The bottom part of Table 4.2 shows results obtained with the state of the art DivideMix method [193]. Somewhat surprisingly, DivideMix was always able to improve results significantly. Note that even the combination CMaxGrad+DivideMix outperformed the best baseline by 3 – 5% on these datasets. When further combined with SimCLR-based explanations, the gains were of about 6% on Butterflies. This suggests that even when the MTurker labels are incorrect they are informative of the true class, as discussed in Section 4.3.2. It also shows that MEMORABLE is a viable alternative to scalable recognition, especially in expert domains.

## 4.7 Conclusion

In this work, we proposed the MEMORABLE framework for scalable recognition in fine-grained expert domains. This is based on the novel CMaxGrad machine teaching algorithm. It is designed starting from a new MaxGrad machine teaching algorithm derived from the optimal student assumption and leverages counterfactual explanations to account for student predictions during the teaching process. We have demonstrated their effectiveness on both synthetic and human student teaching experiments. We have also conducted the first studies of machine teaching in the context of the entire scalable recognition pipeline. It was shown that both CMaxGrad and MEMORABLE achieve superior results to existing solutions to their respective problems. It could be argued that comparing MEMORABLE to previous scalable recognition methods is unfair, since it

leverages additional resources in the form of crowdsourcing. While this is true, we argue that crowdsourcing platforms are now very accessible and dataset labeling is a one-time cost. This must be weighed against the benefits of a better dataset that, as shown by the recent computer vision history, is a gift that keeps on giving.

Chapter 4 is, in full, based on the material as they appear in the publication of “A Machine Teaching Framework for Scalable Recognition”, Pei Wang, Nuno Vasconcelos, In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2021. The dissertation author was the primary investigator and author of this paper.

## Chapter 5

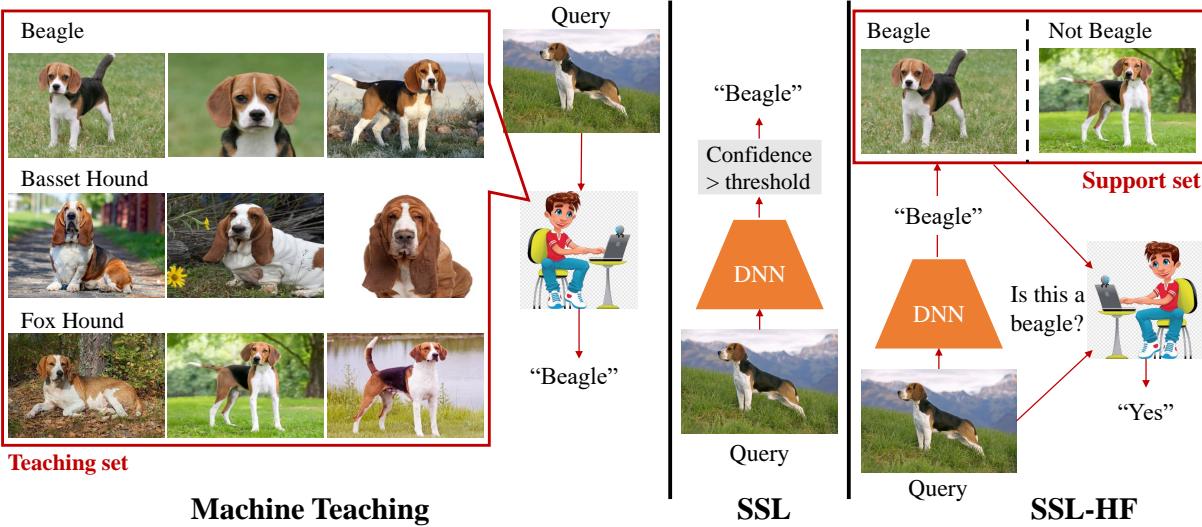
# Towards Professional Level Crowd Annotation of Expert Domain Data

## 5.1 Introduction

While deep learning enabled tremendous advances in image recognition, high recognition performance is still difficult to achieve in expert domains, such as biological or medical imaging, due to two challenges. First, these problems involve fine-grained classes, which differ by subtle visual attributes, such as the dogs of Figure 5.1. Second, large annotated datasets are difficult to produce for specialized areas of biology or medicine, where image labeling requires expert knowledge, which can be too expensive or infeasible at scale. This makes it difficult to train models as strong as those available for non-expert domains, where model training can benefit from millions, or even billions, of labeled examples. To address this challenge, we consider the problem of how to leverage crowd-source platforms to provide professional level crowd annotations for expert domain data, which is denoted as *PrOfeSsional lEvel cRowd* (POSER) annotation.

Since the difficulty is lack of annotator expertise, one route to POSER annotation is to rely on machine teaching algorithms [59, 35, 209]. As illustrated in the left of Figure 5.1, a small teaching set annotated by an expert is used to teach crowd-source workers to discriminate the various classes. The scalability of crowd sourcing platforms is then leveraged to assemble a large labeled dataset [35]. While machine teaching is surprisingly effective for problems of small class cardinality, it is difficult to teach crowd-workers a large number of classes. This is partly because they are averse to complicated training procedures and partly because it relies on short-term memory, which has limited capacity [210, 58]. Hence, machine teaching is restricted to small teaching sets, typically less than 20 images, and a few classes, typically five [59, 35, 209].

The POSER combination of expert domain data and crowd-sourcing also creates challenges to most *human-in-the-loop* schemes in the crowd-source annotation literature. These schemes are usually based on active learning (AL) techniques [75, 211], which assume an oracle that produces a *ground-truth* label per example. To minimize the number of



**Figure 5.1:** Different approaches to the labeling of a query image.

labelling iterations and labelling cost, AL typically selects the *hardest* examples in the dataset to be labelled. However, this strategy is misguided for POSER annotation, where *noisy annotators* are inevitable and the oracle assumption is severely violated. Since hard examples are precisely those where workers make most mistakes, their selection *maximizes labeling noise*. Hence, while AL has achieved success in domains where crowd-source workers are experts, e.g. everyday objects, it is not effective for expert domains.

In this work, we consider an alternative formulation, inspired by semi-supervised learning (SSL) methods [28, 212, 213, 214, 215] where a classifier trained on labelled data produces pseudo-labels for unlabeled examples. These labels are then accepted or rejected by thresholding a classification score, as illustrated in the middle of Figure 5.1. We refer to this process as *pseudo-label filtering*. Accepted labels are added to the training set, the classifier retrained, and the process repeated. SSL has been shown successful for datasets of everyday objects [28, 212, 213, 216], such as CIFAR [34], STL-10 [217], SVHN [218], or ImageNet [14] but frequently collapses in expert domains, even under-performing supervised baselines trained on the small labeled dataset [29, 35, 30]. This is due to the increased difficulty of finer-grained classification, and the well known inability of deep learning to

produce well calibrated confidence scores [32, 33].

While SSL, by itself, does not solve POSER annotation, its strategy of choosing the *easier* examples (higher classification confidence) is more suitable for the noisy POSER annotators than the hardest example strategy of AL. Furthermore, the major SSL weakness - poor pseudo-label filtering - can be significantly improved upon by using humans to filter pseudo-labels. This suggests solving the POSER annotation problem with the *SSL with human filtering* (SSL-HF) approach at the right of Figure 5.1. Unlike machine teaching, where workers are image classifiers, POSER annotation is framed as an SSL problem where they become *filters that verify the pseudo-labels produced by the classifier for unlabeled images*. This has the critical benefit of framing the annotator operation as an *instantaneous low-shot learning problem*, which does not require prior training.

In SSL-HF, given a query image and its pseudo-label (‘Beagle’), the annotator is presented with a small support set containing both positive (‘Beagle’ class) and negative (other classes) images. The annotator then simply declares if they agree with the pseudo-label, based on the similarity of the query image to the support set examples. Due to the well-know ability of humans for confidence calibration [41], this label filtering procedure is much more accurate than that of SSL, enabling POSER annotation with high accuracy. Furthermore, because the filtering is by visual similarity, the labeling is *implicit*, i.e. the annotator does not even need to know the ‘Beagle’ class. Hence, there is no need to teach annotators a priori, eliminating the short-term memory constraints of machine teaching. Together, these properties enable the ultimate goal of POSER annotation: accurate crowd-sourced annotation of expert datasets with large numbers of classes. The main insight behind SSL-HF is to leverage the well known low-shot learning ability of humans [39, 40, 38] to enable annotators to filter labels even in domains where they are not expert. On the other hand, SSL-HF can benefit from the introduction of vision modules that enhance this low-shot learning ability. We have observed that the main difficulty posed by expert



domains is that, when the differences between support set examples are fine-grained, annotators may not know which object details to base their decision on. To address this problem, we introduce deliberative visual explanations [219], which visualize image regions of ambiguity between class pairs, and tailor these explanations to the SSL-HF setting.

Overall, this work makes five contributions. First, we introduce the SSL-HF framework for POSER annotation. This leverages the example selection strategy of SSL, which is more robust to noisy annotations than those of previous human-in-the-loop solutions based on AL. Second, we propose an implementation, where the classifier suggests a label for the image and a support set of a few positive and close-negative examples. This leverages the human ability to perform both classification and confidence estimation with high accuracy in the low-shot setting. Third, to maximize the accuracy of the human filtering of pseudo-labels, the support set is complemented with explanations that visualize the most ambiguous regions for the classifier. Fourth, we present experiments showing that SSL-HF significantly outperforms SSL, AL, and machine teaching approaches to POSER annotation and that explanations enhance these gains. Finally, to minimize the development cost of POSER annotation methods, we introduce an evaluation protocol based on simulated human labeling. We believe that these contributions establish a new research direction at the intersection of human-in-the loop and fine-grained classification, which is important for the advancement of deep learning in expert domains.

## 5.2 Related Work

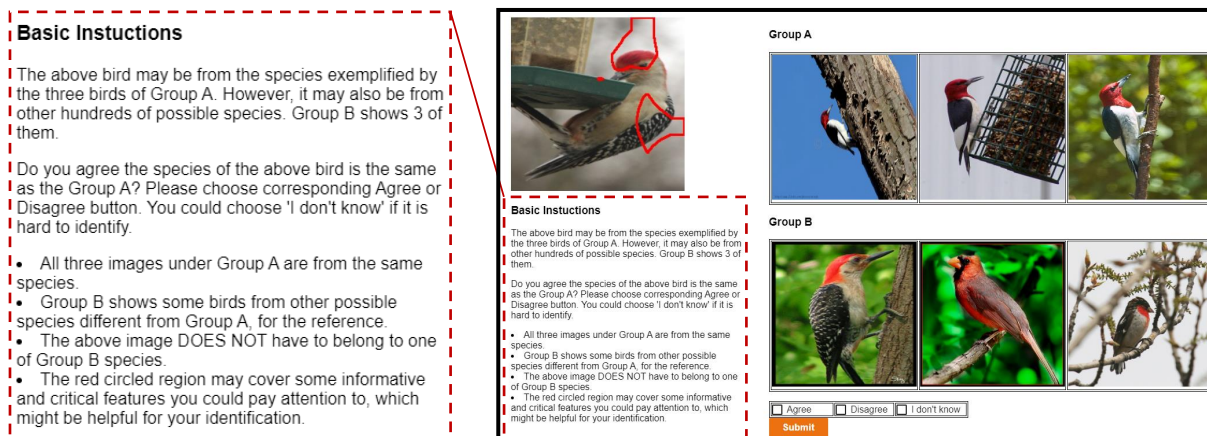
The problem of fine-grained classification with scarce labeled data can be addressed with various approaches.

**Crowd sourcing:** Crowd-source labeling has been critical for the success of deep learning. However, platforms like MTurk are not suitable for expert domain data, due to the lack

of expert annotators. [189] introduced a tool to collect large-scale fine-grained datasets with crowd annotators who are passionate and knowledgeable about a specific domain. However, this is still a much smaller scale of annotation than MTurk. An alternative is to teach MTurk workers using machine teaching algorithms, but these are only applicable to problems of low class cardinality [35]. Our work is partly inspired by [205], who develops a crowdsourcing system for binary detection by asking online workers to select images similar to a target image, from a large pool. However, it is difficult to search a large number of candidates. [220] introduces active learning, only forwarding ‘hard’ examples for human labeling. However, in domains where workers are not experts, they frequently select false positives. Our work aims to extend these approaches to multi-class classification and increase the robustness to the errors of lay annotators.

**Semi-supervised learning (SSL):** SSL methods can be broadly divided into representation learning and pseudo-labelling. Representation learning methods learn a backbone using the unlabeled data and self-supervision. A linear classifier is then learnt on the small labeled dataset [23, 221]. For pseudo-labelling methods [26], a model learned from the labelled data is used to generate pseudo-labels for the unlabeled data. These are then used to improve the model using supervised losses. Pseudo-label methods have achieved better results in SSL challenges [222]. Two popular approaches are self-training [26, 223] and consistency-based learning [28, 212]. For self-training, [224] introduced a self-paced scheme, where high-confidence examples are labeled first and lower confidence examples later on. [225] proposes to replace hard with soft pseudo-labels. While [226, 227] have demonstrated some success for medical images, SSL is still relatively under-explored for the fine-grained classes typical of expert domains. In fact, studies show, that for fine-grained data, SSL frequently under-performs a supervised baseline trained only on the labelled data [29, 35, 30].

**Active learning (AL):** AL uses an acquisition function to select the most useful samples



**Figure 5.2:** Interface (right black box) used in SSL-HF

to label in the unlabeled dataset, so as to minimize labelling cost. Standard AL assumes ground-truth labels produced by an oracle [75, 211]. However, oracle-like annotators are very expensive in expert domains. On crowd source platforms, where noise annotations are inevitable, the oracle assumption is unrealistic. A few papers have considered acquisition functions for noisy oracles [228], post-hoc denoising layers to overcome annotation noise [229], or theoretical results on statistical consistency and query complexity in the presence of noise [230]. However, these works either assume coarse-grained data, simulated noise, or both. We focus on real expert domains with noisy annotators and show, experimentally, that AL methods perform poorly with noisy labels.

**Machine teaching (MT):** MT is a broad research problem [192, 51, 60, 35], which includes the task of leveraging machines to teach humans expert domain knowledge for data labelling. Existing approaches can be grouped into plain [55, 61, 209] or explanations-enhanced [60, 231, 35], depending on whether they use explanations. Motivated by the success of the latter, we introduce deliberative explanations [219] as an aid to the human filtering now proposed.

### 5.3 Challenges of POSER annotation

In this section, we formalize the fine-grained expert domain annotation problem. Previous representative methods are also recapped so as to motivate SSL-HF, which is introduced in the next section.

**Challenges:** Very large datasets annotated on scalable crowd-sourcing platforms, such as MTurk, are critical to the success of deep learning. Their assembly relies on lay annotators, from around the globe, to minimize cost. However, in expert domains, such as biological or medical imaging, annotation requires experts, which is very expensive and unfeasible at scale. While it is typically not difficult to collect a large image dataset  $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{M+N}$ , only a small subset  $\mathcal{D}^a = \{\mathbf{x}_i\}_{i=1}^M$ , where  $M \ll N$  can be realistically labeled. This results in a labeled dataset  $\mathcal{D}^l = \{(\mathbf{x}_i, y_i)\}_{i=1}^M$  where  $y_i$  is the label of  $\mathbf{x}_i$ , and an unlabeled dataset  $\mathcal{D}^u = \mathcal{D} - \mathcal{D}^a$ . Usually, expert domain problems also involve a large number  $C$  of fine-grained classes. Intra-class variation, due to factors like object pose, can easily exceed inter-class variation. The goal is to label  $\mathcal{D}^u$ . Annotation quality can be evaluated by the performance of a classifier  $f$  trained on it together with  $\mathcal{D}^l$ .

**SSL:** SSL is a fully automated approach, where a classifier  $f$  trained on  $\mathcal{D}^l$  generates pseudo-labels  $\hat{y} = f(\mathbf{x})$  and confidence scores  $\sigma(\mathbf{x})$  for each  $\mathbf{x} \in \mathcal{D}^u$ . As shown in the middle of Figure 5.1, pseudo-labels are then filtered by confidence score thresholding, i.e. checking that  $\sigma(\mathbf{x}) > \theta$ . Images from  $\mathcal{D}^u$  that survive this test are added to  $\mathcal{D}^l$ , pseudo-labels accepted as labels, and the process iterated. There are, however, two main difficulties. First, since  $\mathcal{D}^l$  is originally small,  $f$  is not accurate. Second, deep networks produce poorly calibrated confidence scores. Since the two effects compound, pseudo-labels are not trustworthy. It is particularly difficult to propagate labels across images of the same class that are not visually similar to those in  $\mathcal{D}^l$ , e.g. new object poses.

**Human in the loop:** An alternative is to use human-in-the-loop annotation, which iterates between human labeling of images and model training. The challenge is to identify

the images  $\mathbf{x} \in \mathcal{D}^u$  most informative for learning  $f$ , to reduce human labelling effort. This is usually addressed with AL, which is similar to SSL but samples images based on a hardness score  $h(\mathbf{x})$  produced by  $f$  and uses humans as label oracles. In the crowd-source setting, these methods are useful for domains where workers are experts, e.g. everyday objects, but unsuitable for expert domains, where hard examples elicit the most labeling mistakes by workers, violating the oracle assumption.

**Machine Teaching:** In MT, the classifier  $f$  is first trained on  $\mathcal{D}^l$ . A MT algorithm then designs a course, composed of images  $\mathcal{L} \subset \mathcal{D}^l$ , for teaching workers to recognize the  $C$  target classes. The workers trained with  $\mathcal{L}$  then label  $\mathcal{D}^u$ . The classifier  $f$  is finally re-trained on  $\mathcal{D}$ . The process can be iterative, by giving annotators an ‘I don’t know’ (IDK) option and growing  $\mathcal{D}^l$  over steps of machine teaching and human labeling. Since the annotators do not have to be experts, crowd-sourcing platforms can be leveraged for scalability. However, most MT algorithms only support a small number of classes.

## 5.4 SSL with Human Filtering

In this section we introduce the SSL-HF approach.

### 5.4.1 Motivation

Overall, the annotation of large datasets  $\mathcal{D}^u$  in expert domains creates several problems. On one hand, crowd workers cannot be trusted or taught to be good image classifiers. In these domains, label noise is inevitable. This prevents the use of classical human-in-the-loop solutions based on AL, which equate humans to oracles. On the other, fully automated SSL algorithms cannot be trusted to filter pseudo-labels. While SSL accounts for noisy labels, the pseudo-labels produced by  $f$  are usually too poor to enable progress. To address these problems we propose a combination of SSL and human-in-the-

loop, by using humans to *filter* pseudo-labels produced by  $f$ . This inherits the robustness of SSL to noisy labels but leverages the much superior human classification accuracy to filter pseudo-labels.

The SSL-HF process is illustrated in Figure 5.1. Given a query image  $\mathbf{q} \in \mathcal{D}^u$  and a pseudo-label  $\hat{y}$ , in this case ‘Beagle’, the annotator is asked the question ‘do you agree that image  $\mathbf{q}$  belongs to class  $\hat{y}$ ?’. The annotator then responds with  $p = H(\mathbf{q}, \hat{y})$ , where  $p \in \{\text{‘agree’, ‘disagree’, ‘I don’t know’}\}$  and the ‘I don’t know (IDK)’ option allows the annotator to skip images that are too difficult. The problem is that the annotator may not know the ‘Beagle’ class. To overcome this challenge, we propose two mechanisms.

The first is to ask the question *implicitly*, with respect to a *support set* of images, composed by a set  $\mathcal{S}_{\hat{y}} \in \mathcal{D}^l$  of images from class  $\hat{y}$  (positives) and a set of images  $\mathcal{S}_{\hat{y}}^c \in \mathcal{D}^l$  from classes other than  $\hat{y}$  (negatives). This is illustrated in Figure 5.2, which shows a query of the class ‘Red bellied Woodpecker’ that receives the incorrect pseudo-label ( $\hat{y}$ ) ‘Red headed Woodpecker’. The annotator can visually compare the query to three positives ( $\mathcal{S}_{\hat{y}}$  composed of images of ‘Red headed Woodpecker’), shown as ‘Group A,’ and three negatives ( $\mathcal{S}_{\hat{y}}^c$  composed of images of classes ‘Red bellied Woodpecker’, ‘Cardinal’, ‘Rose breasted Grosbeak’), shown as ‘Group B’.

This formulation of label filtering is similar to the definition of the low-shot recognition problem [20, 232] and leverages the known ability of humans to solve this problem. Rather than having to know all the classes, as in MT, the annotator only has to reason in terms of the visual similarity between query and support set examples. Note that, in the example of the figure, it is almost immediately obvious that the query is not a ‘Cardinal’. A detailed examination then reveals that it is also not a ‘Red headed Woodpecker,’ because its head is not fully red, nor a ‘Rose breasted Grosbeak,’ because it has a white breast. However, this type of analysis can exceed the amount of effort that crowd-source workers are willing to devote to the task. The second mechanism aims to address this problem, by

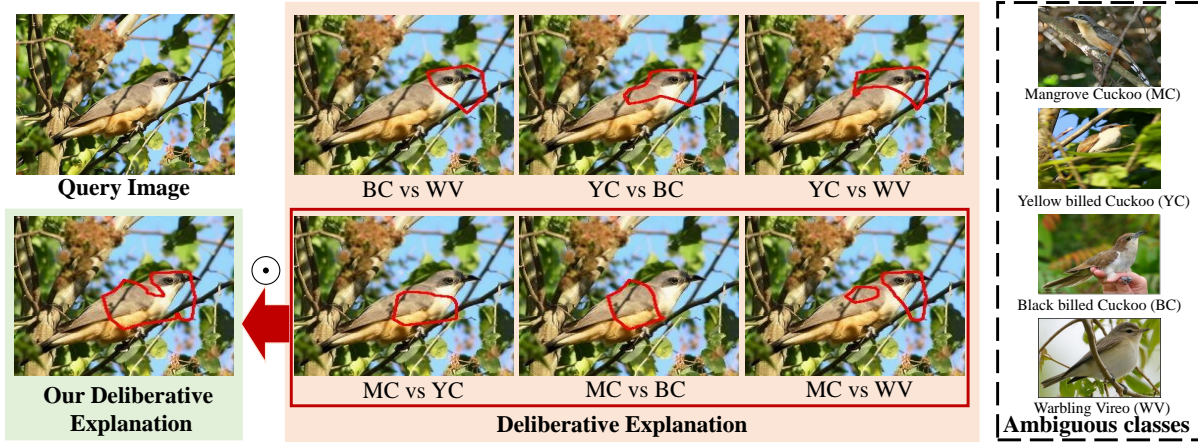
highlighting the image regions of the query most informative for the annotator decision. Namely, the query  $\mathbf{q}$  is enhanced with visual explanations  $\mathbf{m}(\cdot)$  that highlight image regions key to distinguish the positives and negatives in the support set. This is based on deliberative explanations [219] derived from on  $\mathbf{q}$ ,  $\mathcal{S}_{\hat{y}}$ , and  $\mathcal{S}_{\hat{y}}^c$ . In the example of the figure, the explanation highlights the regions of the head and feather texture, which are the most distinctive for the discrimination from the other classes in the support set. Rather than examining the other images in detail, the annotator can then immediately realize that the ‘Red bellied Woodpecker’ shown in the left of Group B is the only bird to have the same feather pattern as the query.

### 5.4.2 Support Set Generation

The two components of the support set are assumed to have the same cardinality,  $|\mathcal{S}_{\hat{y}}| = |\mathcal{S}_{\hat{y}}^c| = K$ . Let  $\mathcal{D}_{\hat{y}}^l$  be the set of examples in  $\mathcal{D}^l$  of ground truth label  $\hat{y}$ . Experimentally, we found no difference between multiple strategies to select the examples of  $\mathcal{S}_{\hat{y}} \subset \mathcal{D}_{\hat{y}}^l$  based on the predicted posterior probability  $f_{\hat{y}}(\mathbf{x})$  of class  $\hat{y}$  given example  $\mathbf{x}$  (see detailed discussion in experiment section). Since randomly selecting  $K$  images from  $\mathcal{D}_{\hat{y}}^l$  to construct  $\mathcal{S}_{\hat{y}}$  was found to be an effective strategy, we use it in the bulk of our experiments.

The assembly of  $\mathcal{S}_{\hat{y}}^c$  is more complex. First, there is a need to decide whether the  $K$  images should come from the same or different classes. We choose to display one image of each of  $K$  classes, to maximize the probability that the true class  $y$  is part of  $\mathcal{S}_{\hat{y}}^c$  when  $\hat{y}$  is incorrect. Next, there is a need to choose the  $K$  classes to display. We select the  $K$  classes other than  $\hat{y}$  of largest probabilities in  $f_{\hat{y}}(\mathbf{q})$ , since these are the most similar to  $\hat{y}$  and thus the potentially most informative for fine-grained class differentiation.

Figure 5.2 illustrates the importance of including  $\mathcal{S}_{\hat{y}}^c$ . In this example, an annotator may not notice that the ‘Red Bellied Woodpecker’ of  $\mathbf{q}$  has a partially red head, while the ‘Red Headed Woodpeckers’ of  $\mathcal{S}_{\hat{y}}$  do not. The inclusion of a ‘Red Bellied Woodpecker’ in



**Figure 5.3:** Deliberative explanation for a query image of a ‘Mangrove Cuckoo’ and simplified explanation used in SSL-HF (green box). Examples from the ambiguous classes are shown on the bottom for illustration only.

$\mathcal{S}_{\hat{y}}^c$  (left image) forces the annotator to realize that there is a class of birds with partially red heads. This makes it clear that  $\mathbf{q}$  does not belong to class  $\hat{y}$ , making the annotators more likely to choose the ‘disagree’ option. In the absence of a fine-grained negative set, these details might be lost, originating a false-positive. Even when  $\mathcal{S}_{\hat{y}}^c$  does not contain images from the groundtruth class  $y$ , the visualization of a diverse set of objects that differ in subtle details is likely to encourage the use of the IDK option whenever  $\hat{y}$  is incorrect.

Given the  $K$  classes that make up  $\mathcal{S}_{\hat{y}}^c$ , it remains to choose one example per class. Similarly to  $\mathcal{S}_{\hat{y}}$ , we have found that random example selection is sufficient.

### 5.4.3 Explanation Generation

A well suited explanation framework for SSL-HF is that of deliberative explanations [219], which highlight the regions that  $f$  finds ambiguous, i.e. likely to belong to more than one class. Formally, a deliberative explanation is a list of insecurities, where an insecurity is a triplet  $(\mathbf{r}, a, b)$ , composed by the segmentation mask  $\mathbf{r}$  of a region of ambiguity between a pair of classes  $(a, b)$ . Figure 5.3 shows an example: a query image  $\mathbf{q}$  of a ‘Mangrove Cuckoo,’ the three most ambiguous classes for  $\mathbf{q}$  (‘Black Billed Cuckoo,’



‘Yellow Billed Cuckoo,’ and ‘Warbling Vireo’), and the deliberative explanation localizing the segments that the classifier deems ambiguous for each pair of classes. We found, however, this to be too much information for the crowd sourcing setting and simplified the explanations as follows. First, there is no need for the explanation to show ambiguities with classes outside the support set. Second, there is no need to even consider ambiguities between pairs of classes in this set, only between the prediction  $\hat{y}$  and the classes in  $\mathcal{S}_{\hat{y}}^c$ . So we only consider the insecurities of  $\mathcal{R} = \{(\mathbf{r}_i, a_i, b_i) | a_i = \hat{y}, b_i \in \mathcal{C}'\}$  where  $\mathcal{C}'$  is the set of  $K$  classes in  $\mathcal{S}_{\hat{y}}^c$ . Finally, instead of showing insecurities separately, we combine them into a single image, by taking the union  $\mathbf{m}(\mathbf{q}) = 1 - \odot_{i=1}^K (1 - \mathbf{r}_i)$ , where  $\odot$  denotes element-wise multiplication and  $\mathbf{r}_i$  is 1 for ambiguous regions and 0 for background. Figure 5.3 shows the result of this operation in the lower left.

#### 5.4.4 Implementation

Human filtering can produce ‘disagree’ or IDK outcomes for the pseudo-label of a particular example. These examples can still be subsequently added to  $\mathcal{D}^l$  if SSL-HF is implemented iteratively. Experimentally, we observed that the human filtering accuracy is positively correlated with the accuracy of the pseudo-labels produced by the classifier  $f$  (see section 5.5.1). Since the accuracy of accepted pseudo-labels determines the performance of  $f$ , there is a positive reinforcement between human filter and classifier accuracy. Hence, best SSL-HF results are usually achieved with a progressive classifier update strategy, where  $\mathcal{D}^l$  grows at each iteration, as unlabeled examples gradually receive labels.

The resulting SSL-HF procedure is summarized in Algorithm 3. At iteration  $t$ , the classifier  $f$  is trained on labeled dataset  $\mathcal{D}^{l,t-1}$ . The classifier is then used to predict labels  $\hat{y}_j$  for each image  $\mathbf{x}_j \in \mathcal{D}^{u,t-1}$ . For examples of high confidence score,  $\sigma(\mathbf{x}_j) > \theta$ , the pseudo-label  $\hat{y}_j = f(\mathbf{x}_j)$  is used to assemble the support set  $\mathcal{S}_{\hat{y}_j}, \mathcal{S}_{\hat{y}_j}^c$ . The human annotator then produces decision  $p = H((\mathbf{x}_j, \hat{y}_j) | \mathcal{S}_{\hat{y}_j}, \mathcal{S}_{\hat{y}_j}^c)$ , where  $p \in \{\text{‘agree,’ ‘disagree,’ IDK}\}$ , that

---

**Algorithm 3 SSL-HF**

---

**Input** Data  $\mathcal{D}^l = \{(\mathbf{x}_i, y_i)\}_{i=1}^M$ ,  $\mathcal{D}^u = \{(\mathbf{x}_j)\}_{j=1}^N$ , #max iteration  $\tau$ , confidence threshold  $\theta$

```
1: Initialization:  $\mathcal{D}^{l,0} \leftarrow \mathcal{D}^l$ ,  $\mathcal{D}^{u,0} \leftarrow \mathcal{D}^u$ ,  $f^0 \leftarrow \arg \min_f \mathcal{R}_{\mathcal{D}^{l,0}}(f)$ ,  $t \leftarrow 1$ .
2: while  $t < \tau$  and empirical risk  $\mathcal{R}_{\mathcal{D}^{l,t}}(f)$  decreases do
3:   for each  $\mathbf{x}_j \in \mathcal{D}^{u,t-1}$  such that  $\sigma(\mathbf{x}_j | f^{t-1}) > \theta$  do { // Data Preparation Loop}
4:      $\hat{y}_j = f^{t-1}(\mathbf{x}_j)$ .
5:     Assemble  $\mathcal{S}_{\hat{y}_j}, \mathcal{S}_{\hat{y}_j}^c$ 
6:   end for
7:    $\mathcal{L}^t \leftarrow \emptyset$ 
8:   for each  $\mathbf{x}_j \in \mathcal{D}^{u,t-1}$  such that  $\sigma(\mathbf{x}_j | f^{t-1}) > \theta$  do { // Crowd Sourcing Loop}
9:      $p_j = H((\mathbf{x}_j, \hat{y}_j) | \mathcal{S}_{\hat{y}_j}, \mathcal{S}_{\hat{y}_j}^c) \in \{\text{agree, disagree, IDK}\}$ 
10:    if  $p_j = \text{agree}$  then
11:       $\mathcal{L}^t = \mathcal{L}^t \cup (\mathbf{x}_j, \hat{y}_j)$ 
12:    end if
13:  end for
14:   $\mathcal{D}^{l,t} \leftarrow \mathcal{D}^{l,t-1} \cup \mathcal{L}^t$ 
15:   $\mathcal{D}^{u,t} \leftarrow \mathcal{D}^{u,t-1} \setminus \mathcal{L}^t$ 
16:  classifier update:  $f^t \leftarrow \arg \min_f \mathcal{R}_{\mathcal{D}^{l,t}}(f)$ .
17:   $t \leftarrow t + 1$ 
18: end while
```

**Output**  $\mathcal{D}^{l,t-1}$ ,  $f^{t-1}$

---

$\mathbf{x}_j$  belongs to class  $\hat{y}_j$ . Examples denoted as ‘agree’ receive the label  $\hat{y}$  and are added to  $\mathcal{D}^l$ . The process is iterated until the empirical risk  $\mathcal{R}_{\mathcal{D}^l}(f)$  of  $f$  on  $\mathcal{D}^l$  does not decrease. While in our implementation, examples are simply selected by thresholding the confidence score, SSL-HF could potentially benefit from more advanced thresholding strategies proposed in the SSL literature, such as dynamic thresholding [233] or a class-specific strategy [216]. In fact, since SSL-HF is an implementation of SSL, it can in principle benefit from any advances on this problem. We leave this for future research.

### 5.4.5 Comparison to Other Methods

When compared to MT solutions, such as MEMORABLE [35], SSL-HF has several benefits. First, filtering labels by comparison to a support set is easier than labeling them

from memory. Second, since there is no need to teach annotators a priori, the labeling experience is more pleasing and much cheaper. Third, because SSL-HF is iterative, a difficult image can be seen by several annotators with several support sets. As common in SSL,  $f$  becomes more capable as the iterations progress. This allows SSL-HF to converge to higher annotation and classifier accuracies. Finally, SSL-HF is applicable to problems with any number of classes while MT is limited to low class cardinalities.

When compared to AL, the main difference is that the SSL-HF annotator is assumed to be noisy. While AL typically samples as query the hardest instance to classify (e.g., an occluded or only partially visible object), SSL-HF samples the image that  $f$  classifies most confidently. Hence, while AL progresses from the labeling of hardest to easiest examples, SSL-HF does the opposite. This is much better suited for noisy annotators, since it avoids the early addition of incorrect labels to the dataset, which can derail  $f$ . When compared to SSL, SSL-HF has the advantage of placing the hardest SSL step, validation of pseudo-labels, on the hands of humans, which are much more competent than any machine learning solution. The downside is the financial cost of the annotations. However, it is now well established that this is not enough to deter the creation of large datasets. We compare the costs of the two approaches in the next section.

It should be noted that while the confidence score of SSL-HF is like those of other SSL methods, it is not decisive for the acceptance of pseudo-labels, which is performed by humans. In fact, in Section 5.5.2 we show that the optimal confidence threshold for SSL-HF is 0.25, much smaller than those typically used in the SSL literature (0.95 for recognition [28] and 0.7 for object detection [234]). This makes SSL-HF less reliant on examples of very high-confidence, allowing a faster convergence. In all our experiments, the entire dataset is labeled in 3-4 iterations. Even without the progressive update (setting  $\theta = 0$ ), SSL-HF has a large gain over competing methods. This is a major benefit of human filtering over plain confidence thresholding.

## 5.5 Experiment

In this section, we demonstrate the effectiveness of SSL-HF. We first introduce the experimental settings, and then show and discuss the results on different set-ups.

**Dataset:** Various fine-grained vision datasets are used: CUB [178], Fungi [29], Butterflies [59] and Gulls [35]. In [29], CUB [178] with 200 bird species is re-organized for Semi-supervised Learning (SSL). The labeled training set has 500 examples from 100 classes (5 examples per class). The unlabeled set has 3,885 in-class examples<sup>1</sup> and 5903 out-class examples by considering the remaining 100 classes of CUB as novel. Fungi has 200 classes, consisting of 4,141 labelled and 13,166 in-class and 64,871 out-class unlabeled images which has 1193 novel classes<sup>2</sup>. This dataset is more difficult because of its long-tailed property. Butterflies and Gulls are two datasets of small class cardinality, with only 5 classes, and 300 (150) labeled images, 1,244 (431) unlabeled images for Butterflies (Gulls). Our results are based on the test sets of [29, 35] with thrice repeated experiments. Both datasets were subject to standard normalizations. Training images were first randomly resized to  $224 \times 224$  and then randomly flipped, whereas testing images were first resized to  $256 \times 256$  and then center-cropped to  $224 \times 224$ . All images were also first converted to  $[0.0, 1.0]$  from  $[0, 255]$  and then normalized by subtracting the mean  $[0.485, 0.456, 0.406]$  and dividing by the standard deviation  $[0.229, 0.224, 0.225]$  of each RGB color channel.

**Network:** For fair comparison with [29, 35], we use ResNet-18 on Butterflies and Gulls, and ResNet-50 on CUB and Fungi if not otherwise stated. The models are pre-trained on ImageNet [14], except for Butterflies where training is from scratch. This follows the setting of [35] because two of the butterfly categories are in ImageNet. We used

---

<sup>1</sup>This number is from the data released on project link <https://github.com/cvl-umass/ssl-evaluation>, which is slightly different from the paper (3,853)

<sup>2</sup>This number is different from 1194 on the project page of <https://github.com/cvl-umass/ssl-evaluation>, because the class ‘Inocybe rimosa’ is repetitively indexed and we fixed this problem.

		Classifier	
		Correct	Incorrect
Human	Agree	TP	FP
	Disagree/IDK	FN	TN
		$R = \frac{TP}{TP+FN}$	$P = \frac{TP}{TP+FP}$
		$\text{Ann Acc} = \frac{TP+TN}{TP+FP+FN+TN}$	

**Figure 5.4:** Confusion matrix for human filter results.

the training setups of [29] on CUB and Fungi<sup>3</sup> and [35] on Butterflies and Gulls<sup>4</sup>. The deliberative explanations and compared Grad-CAM are generated using [219, 103]. We tuned the threshold on the heat map such that 5% image size is remained for visualization, which follows the setting of [219, 35].

**Crowd-sourcing:** Amazon Mechanical Turk is used<sup>5</sup>. The interface is given in Figure 5.2. The per image reward is \$0.01 across all our experiments. We did not limit the maximum number per turker can work on. Statistically, each worker completed 21.1 query image identification tasks on average and the maximum is 135.

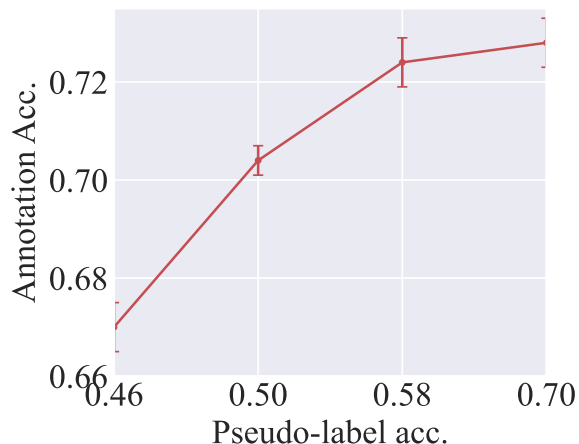
### 5.5.1 Annotation Performance

We performed a study of annotation performance on the fine-grained birds CUB dataset [178], following the SSL setup of [29]. Figure 5.4 defines the confusion matrix of human annotators and statistics such as precision (P), recall (R), and annotation accuracy (Ann Acc). Annotation performance depends on a complex interplay between the quality of the pseudo-labels produced by the classifier  $f$  and the hardness of the examples to annotate. Several experiments were performed to gain insight on this interplay.

<sup>3</sup><https://github.com/cvl-umass/ssl-evaluation>

<sup>4</sup><https://github.com/peiwang062/MEMORABLE>

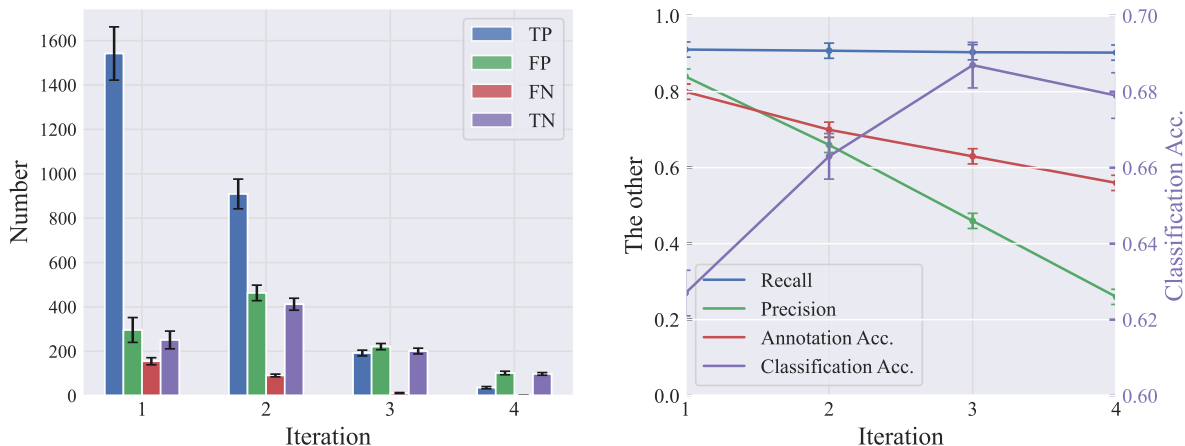
<sup>5</sup><https://www.mturk.com/>



**Figure 5.5:** Human annotation accuracy vs. pseudo label accuracy.

To evaluate how pseudo-label accuracy affects annotator performance, we trained four classifiers of increasing strength (accuracies of 0.46, 0.5, 0.58, 0.7 on  $\mathcal{D}^u$ ), using four labeled datasets  $\mathcal{D}^l$  of increasing size. Figure 5.5 shows the corresponding annotation accuracies on  $\mathcal{D}^u$  after one iteration of SSL-HF. Clearly, human annotation accuracy increases with the accuracy of the pseudo-labels. This shows that there is benefit in improving the classifier, i.e. the SSL component is important. It also justifies the progressive update of  $f$  in Algorithm 3.

We then investigated how example hardness varies with the SSL-HF iteration and how this affects annotator performance. Figure 5.6 left shows how the confusion matrix of the annotation evolves across four SSL-HF iterations. While true positives dominate in the first iteration, this is no longer true by the 3rd, suggesting that the images remaining to label after each iteration are harder. While  $\mathcal{D}^l$  grows with iteration, the newly accepted examples are noisier. The right of the figure shows the impact on annotation P, R, and Ann Acc as well as the accuracy of the classifier  $f$ . The three metrics of annotation performance decrease, confirming that annotation degrades in later iterations. The model  $f$  reaches the best classification accuracy by the 3rd iteration. Note that this does not contradict Figure 5.5, where the comparison is for the same unlabeled image set. In Figure 5.6, annotation



**Figure 5.6:** Results of each iteration under different metrics.

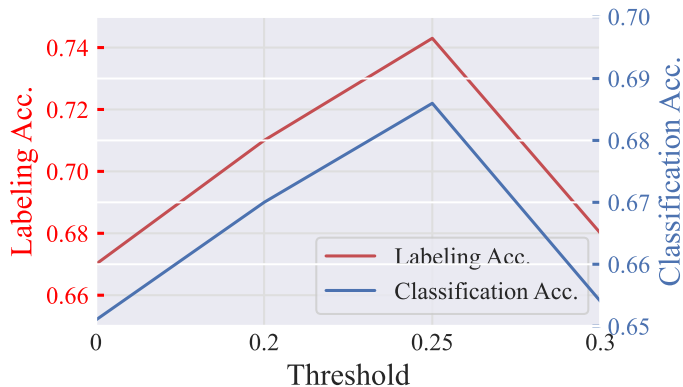
**Table 5.1:** Ablation study for support sets.

	Labeling Acc.	Classification Acc.
A	60.1	59.2
B	66.2	64.1
C	61.1	60.2
D	68.7	65.9
E	74.3	68.6

accuracy declines as the classifier becomes stronger, because the unlabeled data consists of harder instances.

### 5.5.2 Ablation Study

Four different configurations were compared to ablate the mechanisms of section 5.4. In all cases the query is an image. The first two configurations use only text in the support sets. (A) uses the positive support set only, asking turkers if the query image is from class  $\hat{y}$  (replaced with the category name). (B) adds the negative set, displaying the names of  $K$  negative categories. The other two configurations test the importance of including images in the support set. (C) displays the positive support set only and (D) shows the full set of images of the interface of Figure 5.2. None of these experiments use explanations. These



**Figure 5.7:** Ablation study for thresholds.

are added in a final configuration (E), which corresponds to SSL-HF.

Table 5.1 compares the labeling and classification accuracy of all methods, enabling two conclusions. First, without explanations (A to D), it is more important to add a negative support set than example images of the positive set. Note that adding a text-based negative set increases annotator performance by 6%, while adding all images only has an additional gain of 2.5%. The addition of explanations enables a large gain of almost 9%. Second, as expected from the experiments above, improved annotation accuracy leads to better classifiers. Overall, the classifier learned with SSL-HF is almost 10% better than with the simple baseline of A. These results show that the use of negative sets, asking questions implicitly via images, and explanations all contribute to this significant gain. Note how they also demonstrate the importance of SSL-HF for training classifiers in expert domains. For the coarse-grained classification of everyday objects, the baseline of A (“is this a picture of a shoe?”) is sufficient to achieve very high annotation accuracies.

We hypothesize that the good performance of text-based only configurations is due to the fact that, on this dataset, the class name is very strongly aligned with visual features. For example, if a bird does not have a red head, it cannot be a ‘Red Headed Woodpecker.’ When this type of reasoning is not enough, simply adding images has a small gain, because

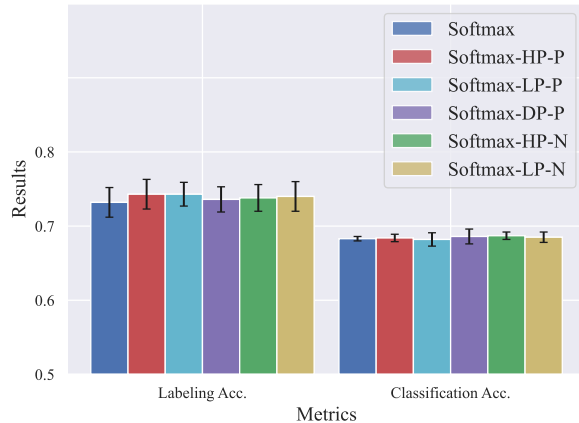


the workers are not sure at what to look for. The addition of explanations allows them to focus on the visual attributes that are important for the classification and reduces this problems. On datasets where class names are not so informative of visual attributes the gains of simply adding images (configurations C D), over the text-only baselines (A B), are likely to be larger.

We also ablated the threshold of confidence scores used to accept labels (step 3 and 8 in Algorithm 3), with the results of Figure 5.7. The optimal threshold, 0.25, is very different from those used for SSL approaches to object recognition (e.g., 0.95 in [28]) and object detection (e.g., 0.7 in [234]). This confirms the claim that human filtering of labels is much more robust than the simple thresholding of confidence scores. Even though most pseudo-labels of low confidence are incorrect, human annotators can still assign the images to the correct class by visually analogy to the examples in the support set, as also demonstrated by the high true positive rates of Figure 5.6. It is only for extremely low values of confidence that the support sets are totally uninformative and human filtering becomes ineffective. In fact, the confidence threshold cannot be too high for SSL-HF, as this leads to the acceptance of only the examples that are relatively easier. Such examples fail to induce improvement of the classifier, which subsequently fails to produce better pseudo labels for the next iteration. In result, the gradual update of the classifier does not happen.

We did a comprehensive ablation study on the support set additionally, including image sampling strategy for creation of support sets sets, support set cardinality, etc.

**Sample choice of the positive support sets** We consider four strategies to select the examples of support set  $\mathcal{S}_{\hat{y}} \subset \mathcal{D}_{\hat{y}}^l$ , based on the predicted posterior probability  $f_{\hat{y}}(\mathbf{x})$  of class  $\hat{y}$  given example  $\mathbf{x}$ . Strategy S1 is to choose the examples of  $K$  highest probabilities  $f_{\hat{y}}(\mathbf{x})$  ('Softmax-HP-P'). These are the easiest to assign to class  $\hat{y}$  and include the most representative class features. Strategy S2 is to choose examples with the  $K$  lowest top-



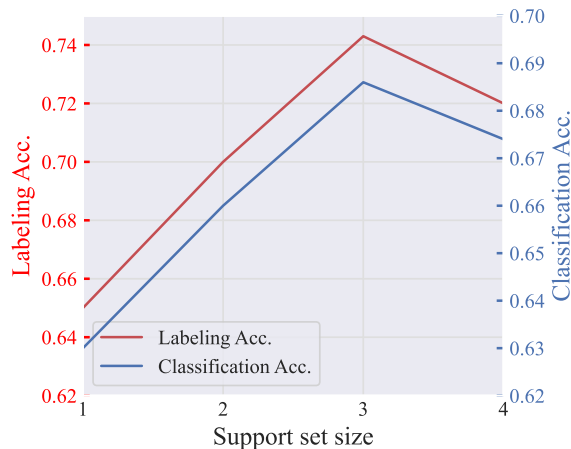
**Figure 5.8:** Result comparison of different support set sample choices.

probability  $f_{\hat{y}}(\mathbf{x})$  (‘Softmax-LP-P’). These are harder and more likely to be outliers for class  $\hat{y}$ , including features that are rarely visible, occlusions, or other variations. Strategy S3 is to select a set of examples with diverse probability  $f_{\hat{y}}(\mathbf{x})$  (‘Softmax-DP-P’). This means the selected examples have more diverse features. Finally, Strategy S4 is to select the examples randomly (‘Softmax’), which is used as baseline. Figure 5.8 compares the results. We have found no big difference between these strategies and just used randomly selection.

**Sample choice of the negative support sets** For  $\mathcal{S}_{\hat{y}}^c$ , similarly to  $\mathcal{S}_{\hat{y}}$ , we experimented with the highest-probability (‘Softmax-HP-N’), lowest top-probability (‘Softmax-LP-N’), and random example, again finding that these strategies make no big difference. Figure 5.8 shows the results as well.

**The size of support sets** The support set size  $K$  is ablated from 1 to 4. Figures 5.9 shows that with just one image both annotation and classification accuracies are weak. Both accuracies improve for larger  $K$  saturating at about  $K = 3$ . This likely reflects the fact that too many images can be distracting or even confusing.

**Explanations** We investigate the importance of explanations, comparing attributive



**Figure 5.9:** Result comparison of different support set sizes.

**Table 5.2:** Ablation study for support sets.

	Lab. Acc.	Cla. Acc.
Softmax	68.7	65.9
Softmax+attributive	71.4	67.1
Softmax+deliberative	74.3	68.6

explanations based on Grad-CAM [103], (‘w Grad-CAM’)<sup>6</sup> and the proposed deliberative explanations (‘w deliberative’), with results on Table 5.2. The baseline ‘Softmax’ is the setting only having the support set, corresponding to the D of Table 5.1. Overall, although Grad-CAM enables a clear improvement, the proposed deliberative explanations have the largest benefit.

### 5.5.3 Comparisons on Crowd-source Platforms

POSER annotation with SSL-HF, using MTurk workers, was compared to various other approaches, on several expert-domain datasets: CUB [178] (100 classes) and Fungi [29] (200), which have large class cardinality, and Butterflies [59] (5) and Gulls [35] (5), which are machine teaching datasets.

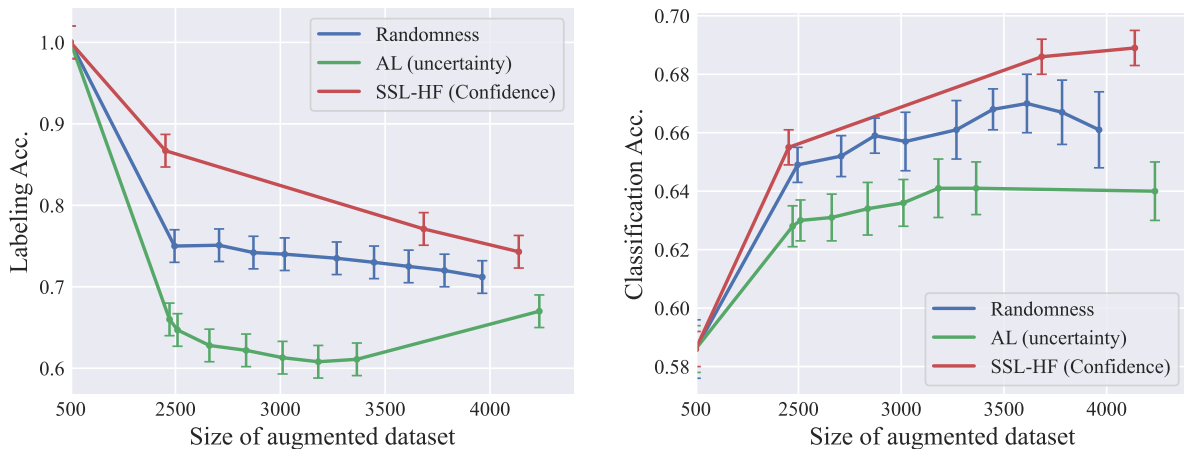
<sup>6</sup>On Grad-CAM experiments, a slightly different description for circled regions for turkers is given, “The circle regions may have some class-specific features, which might be helpful for your identification.”

**Table 5.3:** Classification accuracy (mean(std)) comparison with the state of the art SSL. \* denotes the results are generated by human simulation. Missing std because of no std reported in the original literature.

		In Distribution		Out of Distribution	
		CUB	Fungi	CUB	Fungi
Baseline	Sup. expert (on $\mathcal{D}^l$ )	58.7	53.8 (0.4)	58.7	53.8 (0.4)
Upper bound	Sup. oracle (on $\mathcal{D}^l \cup \mathcal{D}^u$ )	84.5	73.3 (0.1)	84.5	73.3 (0.1)
SSL	MoCo [221]	59.2 (0.6)	55.2 (0.2)	57.9 (0.5)	52.9 (0.3)
	Pseudo-Label [26]	57.0	51.5 (1.2)	59.1	52.4 (0.2)
	Curr. Pseudo-Label [223]	57.3	53.7 (0.2)	59.6	54.2 (0.2)
	FixMatch [28]	53.2	56.3 (0.5)	52.8	51.2 (0.6)
	Self-Training [29]	61.3	56.9 (0.3)	61.4	55.7 (0.3)
POSER	SSL-HF	<b>68.6 (0.6)</b>	<b>60.0 (0.4)</b>	<b>65.0 (0.9)</b>	<b>57.8 (0.5)</b>

**SSL:** Table 5.3 compares SSL-HF to SSL methods on the benchmarks of [29, 35]. These include both an in-distribution, where unlabeled data and labeled data are from the same class space, and an out-of distribution, where unlabeled data has novel classes, setting. The importance of data annotations is reflected by the large gap between supervised learning from  $\mathcal{D}^l$  (expert labeled dataset) and  $\mathcal{D}^l \cup \mathcal{D}^l$  (upper bound, fully labeled) for all datasets. However, vanilla SSL is of little help, since all methods have little to no gain over learning from  $\mathcal{D}^l$  alone. This is unlike POSER annotation with SSL-HF, which achieves significant gains over expert annotation. The gains can be as high as 10% for in-distribution and 6% for out-of distribution data.

**AL:** SSL-HF was compared to AL and random example selection. These were implemented with Algorithm 3, by replacing the function used to select examples in steps 3 and 8. For AL [235, 236, 237],  $\sigma(\mathbf{x}_j | f^{t-1})$  was replaced by an entropy-based acquisition function [237], which forwards images of high classification uncertainty to the turkers. For random selection it was replaced by a sample from a uniform distribution in  $[0, 1]$ . Figure 5.10 shows how annotation and classification accuracy vary with the amount of data from  $\mathcal{D}^u$  that is labeled. SSL-HF is always the best method and AL the worst, even worse than random. This



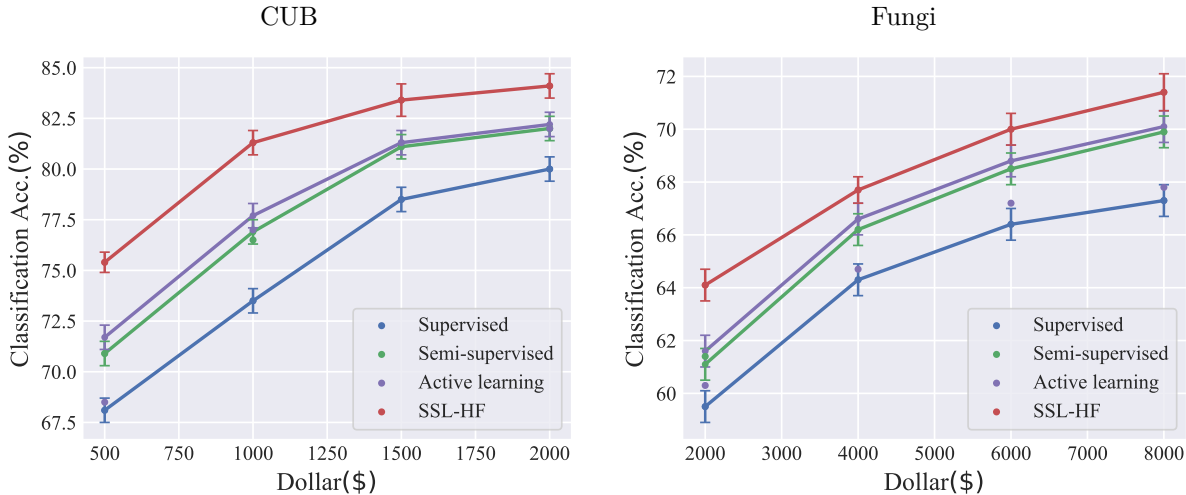
**Figure 5.10:** Accuracy comparison under different thresholding strategies

**Table 5.4:** Labeling and classification accuracy (mean(std)) comparison.

		Labeling Acc.		Classification Acc.	
		Butterflies	Gulls	Butterflies	Gulls
Baseline	Sup. expert (on $\mathcal{D}^l$ )	-	-	58.7	53.8 (0.4)
Upper bound	Sup. oracle (on $\mathcal{D}^l \cup \mathcal{D}^u$ )	-	-	84.5	73.3 (0.1)
MT	MEMORABLE [35]	<b>77.1</b> (1.2)	68.3 (1.8)	<b>77.5</b> (0.5)	60.2 (1.1)
SSL-HF	SSL-HF	73.6 (0.8)	<b>74.1</b> (0.5)	73.0 (0.7)	<b>63.3</b> (0.5)

confirms our claims that the selection of hard examples performed by AL is not suitable for the noisy annotators of POSER annotation.

**MT:** Table 5.4 compares SSL-HF with a state of the art MT algorithm [35]. These experiments are restricted to the small class cardinality datasets supported by MT. They confirm the previous observation that higher labeling accuracy leads to higher classification accuracy. Regarding relative performance, the results are mixed, with better results for [35] in Butterflies and for SSL-HF in Gulls. This is explained by the fact that Butterflies is not as fine-grained as Gulls, a fact confirmed by the higher classification accuracies of the former. In result, Gull classes are harder to commit to short-term memory and the annotation performance of MT degrades. Note that while SSL-HF is slightly inferior to MT



**Figure 5.11:** The trade-off comparison of supervised/SSL/SSL-HF.

**Table 5.5:** Classification accuracy of simulated experiments with different  $R$ , and Two-sample two-tailed T test. Mean(std)/t-score, using p-value of 0.05.

$R$	Simulated					Real 3885
	400	500	1000	1500	2000	
Accuracy/t-score	69.8(0.4)/2.95	69.6(0.4)/2.40	69.2(0.3)/1.58	68.9(0.4)/0.72	68.5(0.3)/-0.26	68.6(0.6)
Conclusion	Reject	Accept	Accept	Accept	Accept	

on the easier dataset, it achieves similar annotation accuracy on the two datasets. This suggests that visual reasoning in terms of support sets and visual explanations is quite robust, unlike the memorization required by MT. This and the scalability of SSL-HF with class cardinality make SSL-HF a clearly better overall solution.

### 5.5.4 Comparisons by Human Simulation

**Protocol:** Crowd source experiments are difficult to replicate and expensive. Hence, there is a benefit to simulated evaluation protocols that facilitate algorithmic development. These should mimic human annotations as closely as possible. Following [35], we propose a simulated protocol to evaluate SSL-HF, based on estimates of the confusion matrix of Figure 5.4, obtained on a small dataset.  $R$  examples are sampled from  $\mathcal{D}^u$ , forwarded to

human annotators, the confusion matrix is computed and used to simulate the annotators for the remaining unlabeled examples. Given a new example  $(\mathbf{x}, y)$  and pseudo-label  $\hat{y}$ , a random number ( $p$ ) is sampled from a uniform distribution in  $[0, 1]$ . If  $\hat{y} = y$ , the human decision is simulated as ‘Agree’ when  $p < \frac{TP}{TP+FN}$  and ‘Disagree/IDK’ otherwise. If  $\hat{y} \neq y$ , ‘Agree’ is declared when  $p < \frac{FP}{FP+TN}$  and ‘Disagree/IDK’ otherwise.

To determine how many examples  $R$  are needed to produce a realistic confusion matrix, we performed a two-sample two-tailed T test comparing the classification accuracies of human and simulated labeling. Table 5.5 lists statistics for different values of  $R$ . The null-hypothesis is that the underlying population means are the same. The t-scores are computed for a p-value of 0.05, and the null-hypothesis is accepted for all  $R \geq 500$ . This suggests that simulation is a very economical alternative to user experiments.

**Cost-accuracy trade-off:** We used simulation to compare supervised learning from  $\mathcal{D}^l$ , SSL-HF, SSL, and AL with respect to the trade-off between classifier accuracy and annotation cost (dollars). These experiments are too expensive to perform on MTurk, due to the need to explore various points along the trade-off.

For supervised and SSL methods, the entire labeling budget is spent on expert annotations. SSL-HF and AL split labels between experts and crowd source workers. These annotations have very different costs. For workers, we assume the rate of \$0.01 per image, used in all experiments above and customary on MTurk. The cost of an expert is harder to determine and can vary significantly with the application area, e.g. doctors tend to be more expensive than botanists. We tried to identify a lower bound for the cost, in a domain of mild expertise. For this, we asked MTurkers to take a survey, declaring if they were specialists on birds or fungi. To answer the survey, they were shown 3 images of birds or fungi. Those who felt confident about their ability to do the classification, were then asked the expected per image reward, for labeling images from 100 candidate classes. Four options were given:  $< \$0.1$ ,  $\$0.1 - \$0.5$ ,  $\$0.5 - \$1.0$ , and  $> \$1$ . We gathered 5 results for

birds and 3 for fungus. One person chose \$0.5 – \$1.0 and all others chose  $> \$1$ , showing that the task is considered difficult. We thus use \$1 as cost estimate for expert labeling. This can be thought as a lower bound, although it is unrealistically low for many image domains. We then assumed a total dollar budget and determined the number of images labeled by experts and workers. Figure 5.11 shows the plots of cost vs classification accuracy of the different methods on CUB. SSL-HF achieves the best trade-off. For example, its accuracy for a cost of \$800 equals those of SSL for \$1,200 and Supervised for \$1,700.

## 5.6 Conclusion

In this work, we proposed SSL-HF, a new method for crowd source annotation of expert domain data. SSL-HF is a human-in-the-loop SSL method, where crowd-source workers act as pseudo-label filters. To enable annotation by non-experts, classes are specified implicitly, via positive and negative sets of examples and augmented with deliberative explanations, which highlight regions of class ambiguity. This leverages the strong low-shot learning and confidence estimation ability of humans to dramatically improve SSL performance. Experiments have show that SSL-HF significantly outperforms alternatives such as machine teaching or active learning for expert domain problems of large numbers of classes.

Chapter 5 is, in full, based on the material as it appears in the submission of “Towards Crowd-Source Annotation of Expert Domain Data”, Pei Wang, Nuno Vasconcelos, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. The dissertation author was the primary investigator and author of this paper.



## Chapter 6

# Discussion and Conclusion

In this thesis, we have studied the possibility of using crowd-source annotations to deal with data-limited problems on fine-grained expert domains, in order to scale up recognition on them. We have proposed two methods MEMORABLE and SSL-HF, based on machine teaching and human filtering, and demonstrated that explanations can enhance the two methods.

We started by introducing a new machine teaching algorithm MaxGrad which is designed specifically for crowd-sourcing scenarios, based on an optimal student assumption. MaxGrad can teach humans to learn some new concepts of expert domains in a short course. This makes it possible to label more data on crowd-sourcing platforms. Its effectiveness has been demonstrated in experiments for both machine and human learners.

In order to improve the labeling accuracy, explanations have been adopted to benefit the teaching process. A generalized framework, GALORE, has been proposed, which can generate two new types of explanations, deliberative explanations and counterfactual explanations. GALORE complements the existing attributive explanations that only answer “why” question with deliberative explanations to address “why” question by exposing the network insecurities about a prediction, and counterfactual explanations to address “why not” question proposed by end-users toward a network prediction regarding a counterfactual prediction. The produced explanations have been proven useful and consistent with human intuition.

We then introduced our machine teaching method for scalable recognition, MEMORABLE. It is a general framework where humans are regarded as classifiers. Within this framework, we use the handy data to train a neural network, and generate a short course to teach humans the domain knowledge so that they can label more data in order to train a better network. In MEMORABLE, MaxGrad has been combined with counterfactual explanations into an explanation-equipped version CMaxGrad. CMaxGrad and MEMORABLE have been proven effective in teaching humans and scalable recognition,

respectively.

We finally introduced another method SSL-HF by Human Filtering. Instead of classifiers, humans act as filters to select pseudo-labels for unlabeled examples. SSL-HF solved a critical limitation of MEMORABLE. It has the deficiency of scalability of class numbers. It also reduced the workload of annotators when labeling. Extensive experiments have shown that SSL-HF is much better than other methods and is a very promising potential solution for scalable recognition problems. Similar to MEMORABLE, deliberative explanations have been shown helpful to human annotating.

To compare between two methods, in MEMORABLE, humans are taught and expected to master the domain knowledge via a preceding course. On some easier tasks MEMORABLE shows better performances because after training, they have been approaching the experts. However, the job is relatively demanding and can not be scaled up to any class number. SSL-HF simplifies the process to a binary decision task and requires much less learning from the annotators, which makes the labeling task much easier and scalable to any number of classes. Since there is no need to teach annotators before they can start labeling, the process is also more pleasing for the latter and much less expensive. We also note that the two explanations are specifically suitable for two methods. For MEMORABLE, it is based on machine teaching. There is a teaching stage. We selected a small teaching set from the expert-labeled set and we have known the ground truth of the teaching examples. Because the teaching process is interactive where humans can propose a counter class, in this case, the setting exactly fits the formulation of counterfactual explanations in which we have two classes, one is the ground truth and another is the counter. On the contrary, in SSL-HF, there is no teaching part. Humans are regarded as filters to filter the unlabeled examples directly without known the ground truth. In this case, deliberative explanations are more suitable because they do not rely on ground truth labels.

In summary, both MEMORABLE and SSL-HF are the pioneer attempts to solve label-limited expert domain recognition problems, by laypeople on crowd-sourcing platforms, to the best of our knowledge. They have presented priority to other methods. We believe that our solution with crowd-source annotations is a valuable alternative method to scalable recognition and is worth earning more attentions in the community. We hope this thesis can motivate more people to work on this field.

# Bibliography

- [1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, pp. 770–778, 2016.
- [2] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *ICCV*, pp. 10012–10022, 2021.
- [3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *ICLR*, 2020.
- [4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *NIPS*, vol. 28, 2015.
- [5] Z. Cai and N. Vasconcelos, “Cascade r-cnn: Delving into high quality object detection,” in *CVPR*, 2018.
- [6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *ECCV*, pp. 213–229, Springer, 2020.
- [7] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, pp. 3431–3440, 2015.
- [8] G. Lin, A. Milan, C. Shen, and I. Reid, “Refinenet: Multi-path refinement networks for high-resolution semantic segmentation,” in *CVPR*, pp. 1925–1934, 2017.
- [9] J. Gu, H. Kwon, D. Wang, W. Ye, M. Li, Y.-H. Chen, L. Lai, V. Chandra, and D. Z. Pan, “Multi-scale high-resolution vision transformer for semantic segmentation,” in *CVPR*, pp. 12094–12103, 2022.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

- [11] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.
- [12] X. Zhang, H. Luo, X. Fan, W. Xiang, Y. Sun, Q. Xiao, W. Jiang, C. Zhang, and J. Sun, “Alignedreid: Surpassing human-level performance in person re-identification,” *arXiv preprint arXiv:1711.08184*, 2017.
- [13] C. Lu and X. Tang, “Surpassing human-level face verification performance on lfw with gaussianface,” in *AAAI*, 2015.
- [14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, pp. 248–255, Ieee, 2009.
- [15] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, “The kinetics human action video dataset,” *arXiv preprint arXiv:1705.06950*, 2017.
- [16] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun, “Objects365: A large-scale, high-quality dataset for object detection,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8430–8439, 2019.
- [17] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov, *et al.*, “The open images dataset v4,” *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, 2020.
- [18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, pp. 740–755, Springer, 2014.
- [19] <https://www.mturk.com/>
- [20] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.
- [21] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, “Meta-transfer learning for few-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 403–412, 2019.
- [22] J. Xu, H. Le, M. Huang, S. Athar, and D. Samaras, “Variational feature disentangling for fine-grained few-shot classification,” in *ICCV*, pp. 8812–8821, 2021.
- [23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

- [24] I. Misra and L. v. d. Maaten, “Self-supervised learning of pretext-invariant representations,” in *CVPR*, pp. 6707–6717, 2020.
- [25] J.-B. Grill, F. Strub, F. Alché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, *et al.*, “Bootstrap your own latent—a new approach to self-supervised learning,” *NeurIPS*, vol. 33, pp. 21271–21284, 2020.
- [26] D.-H. Lee *et al.*, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in *Workshop on challenges in representation learning, ICML*, vol. 3, 2013.
- [27] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, “S4l: Self-supervised semi-supervised learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1476–1485, 2019.
- [28] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” *arXiv preprint arXiv:2001.07685*, 2020.
- [29] J.-C. Su, Z. Cheng, and S. Maji, “A realistic evaluation of semi-supervised learning for fine-grained classification,” in *CVPR*, pp. 12966–12975, 2021.
- [30] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, “What should not be contrastive in contrastive learning,” *arXiv preprint arXiv:2008.05659*, 2020.
- [31] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, “A closer look at few-shot classification,” *arXiv preprint arXiv:1904.04232*, 2019.
- [32] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *ICML*, pp. 1321–1330, PMLR, 2017.
- [33] Y. Wang, B. Li, T. Che, K. Zhou, Z. Liu, and D. Li, “Energy-based open-world uncertainty modeling for confidence calibration,” in *ICCV*, pp. 9302–9311, 2021.
- [34] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” tech. rep., Citeseer, 2009.
- [35] P. Wang and N. Vasconcelos, “A machine teaching framework for scalable recognition,” in *ICCV*, pp. 4945–4954, October 2021.
- [36] <https://www.microworkers.com/>
- [37] <https://www.clickworker.com/>
- [38] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, “A survey of human-in-the-loop for machine learning,” *Future Generation Computer Systems*, 2022.

- [39] S. Wan, Y. Hou, F. Bao, Z. Ren, Y. Dong, Q. Dai, and Y. Deng, “Human-in-the-loop low-shot learning,” *T-NNLS*, vol. 32, no. 7, pp. 3287–3292, 2020.
- [40] F. Ansari, S. Erol, and W. Sihm, “Rethinking human-machine learning in industry 4.0: how does the paradigm shift treat the role of human learning?,” *Procedia manufacturing*, vol. 23, pp. 117–122, 2018.
- [41] L. Cosmides and J. Tooby, “Are humans good intuitive statisticians after all? rethinking some conclusions from the literature on judgment under uncertainty,” *cognition*, vol. 58, no. 1, pp. 1–73, 1996.
- [42] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, T. Duerig, *et al.*, “The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale,” *arXiv preprint arXiv:1811.00982*, 2018.
- [43] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, “Activitynet: A large-scale video benchmark for human activity understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 961–970, 2015.
- [44] A. Antoniou and A. J. Storkey, “Learning to learn by self-critique,” in *Advances in Neural Information Processing Systems*, pp. 9936–9946, 2019.
- [45] S. Kornblith, J. Shlens, and Q. V. Le, “Do better imagenet models transfer better?,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2661–2671, 2019.
- [46] W. Ying, Y. Zhang, J. Huang, and Q. Yang, “Transfer learning via learning to transfer,” in *International Conference on Machine Learning*, pp. 5085–5094, 2018.
- [47] L. Yu, V. O. Yazici, X. Liu, J. v. d. Weijer, Y. Cheng, and A. Ramisa, “Learning metrics from teachers: Compact networks for image embedding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2907–2916, 2019.
- [48] D. Li, W.-C. Hung, J.-B. Huang, S. Wang, N. Ahuja, and M.-H. Yang, “Unsupervised visual representation learning by graph-based consistent constraints,” in *European Conference on Computer Vision*, pp. 678–694, Springer, 2016.
- [49] J. Sauder and B. Sievers, “Self-supervised deep learning on point clouds by reconstructing space,” in *Advances in Neural Information Processing Systems*, pp. 12942–12952, 2019.
- [50] A. Kolesnikov, X. Zhai, and L. Beyer, “Revisiting self-supervised visual representation learning,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1920–1929, 2019.



- [51] X. Zhu, A. Singla, S. Zilles, and A. N. Rafferty, “An overview of machine teaching,” *arXiv preprint arXiv:1801.05927*, 2018.
- [52] M. Cakmak and M. Lopes, “Algorithmic and human teaching of sequential decision tasks,” in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [53] C. J. Butz, S. Hua, and R. B. Maguire, “A web-based intelligent tutoring system for computer programming,” in *IEEE/WIC/ACM International Conference on Web Intelligence (WI’04)*, pp. 159–165, IEEE, 2004.
- [54] L. Von Ahn, “Duolingo: learn a language for free while helping to translate the web,” in *Proceedings of the 2013 international conference on Intelligent user interfaces*, pp. 1–2, 2013.
- [55] A. Singla, I. Bogunovic, G. Bartók, A. Karbasi, and A. Krause, “Near-optimally teaching the crowd to classify,” in *ICML*, vol. 1, p. 3, 2014.
- [56] W. Liu, B. Dai, A. Humayun, C. Tay, C. Yu, L. B. Smith, J. M. Rehg, and L. Song, “Iterative machine teaching,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2149–2158, JMLR. org, 2017.
- [57] W. Liu, B. Dai, X. Li, Z. Liu, J. M. Rehg, and L. Song, “Towards black-box iterative machine teaching,” *International Conference on Machine Learning*, 2018.
- [58] K. R. Patil, X. Zhu, L. Kopeć, and B. C. Love, “Optimal teaching for limited-capacity human learners,” in *Advances in neural information processing systems*, pp. 2465–2473, 2014.
- [59] O. Mac Aodha, S. Su, Y. Chen, P. Perona, and Y. Yue, “Teaching categories to human learners with visual explanations,” in *CVPR*, pp. 3820–3828, 2018.
- [60] Y. Chen, O. Mac Aodha, S. Su, P. Perona, and Y. Yue, “Near-optimal machine teaching via explanatory teaching sets,” in *International Conference on Artificial Intelligence and Statistics*, pp. 1970–1978, PMLR, 2018.
- [61] E. Johns, O. Mac Aodha, and G. J. Brostow, “Becoming the expert-interactive multi-class machine teaching,” in *CVPR*, pp. 2616–2624, 2015.
- [62] Y. Zhou, A. R. Nelakurthi, and J. He, “Unlearn what you have learned: Adaptive crowd teaching with exponentially decayed memory learners,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2817–2826, 2018.
- [63] E. L. Grigornko, R. J. Sternberg, and M. E. Ehrman, “A theory-based approach to the measurement of foreign language learning ability: The canal-f theory and test,” *The Modern Language Journal*, vol. 84, no. 3, pp. 390–405, 2000.

- [64] S. J. Derry and D. A. Murphy, “Designing systems that train learning ability: From theory to practice,” *Review of educational research*, vol. 56, no. 1, pp. 1–39, 1986.
- [65] N. W. Hatch and J. H. Dyer, “Human capital and learning as a source of sustainable competitive advantage,” *Strategic management journal*, vol. 25, no. 12, pp. 1155–1178, 2004.
- [66] H. A. Simon, “Bounded rationality and organizational learning,” *Organization science*, vol. 2, no. 1, pp. 125–134, 1991.
- [67] M. Saberian and N. Vasconcelos, “Multiclass boosting: Margins, codewords, losses, and algorithms,” *Journal of Machine Learning Research*, vol. 20, no. 137, pp. 1–68, 2019.
- [68] Y. Freund, R. Schapire, and N. Abe, “A short introduction to boosting,” *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.
- [69] M. J. Saberian and N. Vasconcelos, “Multiclass boosting: Theory and algorithms,” in *Advances in Neural Information Processing Systems*, pp. 2124–2132, 2011.
- [70] S. Basu and J. Christensen, “Teaching classification boundaries to humans,” in *AAAI*, 2013.
- [71] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, “Deep knowledge tracing,” in *NIPS*, pp. 505–513, 2015.
- [72] J. Zhu, “Machine teaching for bayesian learners in the exponential family,” in *Advances in Neural Information Processing Systems*, pp. 1905–1913, 2013.
- [73] J. H. Bak, J. Y. Choi, A. Akrami, I. Witten, and J. W. Pillow, “Adaptive optimal training of animal behavior,” in *Advances in neural information processing systems*, pp. 1947–1955, 2016.
- [74] A. N. Rafferty, E. Brunskill, T. L. Griffiths, and P. Shafto, “Faster teaching via pomdp planning,” *Cognitive science*, vol. 40, no. 6, pp. 1290–1332, 2016.
- [75] B. Settles, “Active learning literature survey,” tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.
- [76] D. Wang and Y. Shang, “A new active labeling method for deep learning,” in *2014 International joint conference on neural networks (IJCNN)*, pp. 112–119, IEEE, 2014.
- [77] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.
- [78] G. Hacothen and D. Weinshall, “On the power of curriculum learning in training deep networks,” *ICML*, 2019.

- [79] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
- [80] S. Su, Y. Chen, O. Mac Aodha, P. Perona, and Y. Yue, “Interpretable machine teaching via feature feedback,” 2017.
- [81] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [82] K. Seeliger, M. Fritsche, U. Güçlü, S. Schoenmakers, J.-M. Schoffelen, S. Bosch, and M. Van Gerven, “Convolutional neural network-based encoding and decoding of visual object recognition in space and time,” *NeuroImage*, vol. 180, pp. 253–266, 2018.
- [83] M. Eickenberg, A. Gramfort, G. Varoquaux, and B. Thirion, “Seeing it all: Convolutional network layers map the function of the human visual system,” *NeuroImage*, vol. 152, pp. 184–194, 2017.
- [84] A. A. Zeman, J. B. Ritchie, S. Bracci, and H. O. de Beeck, “orthogonal representations of object shape and category in deep convolutional neural networks and human visual cortex,” *Scientific Reports*, vol. 10, no. 1, pp. 1–12, 2020.
- [85] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, “The inaturalist species classification and detection dataset,” 2018.
- [86] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, “Casia online and offline chinese handwriting databases,” in *2011 International Conference on Document Analysis and Recognition*, pp. 37–41, IEEE, 2011.
- [87] O. Mac Aodha, S. Su, Y. Chen, P. Perona, and Y. Yue, “[https://github.com/macaodha/explain\\_teach/tree/master/data](https://github.com/macaodha/explain_teach/tree/master/data).”
- [88] P. C.-H. Lam, L. Chu, M. Torgonskiy, J. Pei, Y. Zhang, and L. Wang, “Finding representative interpretations on convolutional neural networks,” in *ICCV*, pp. 1345–1354, 2021.
- [89] J. Wang, H. Liu, X. Wang, and L. Jing, “Interpretable image recognition by constructing transparent embedding space,” in *ICCV*, pp. 895–904, 2021.
- [90] M. Nauta, R. van Bree, and C. Seifert, “Neural prototype trees for interpretable fine-grained image recognition,” in *CVPR*, pp. 14933–14943, 2021.
- [91] B. Carter, S. Jain, J. W. Mueller, and D. Gifford, “Overinterpretation reveals image classification model pathologies,” *NeurIPS*, vol. 34, 2021.

- [92] J. Parekh, P. Mozharovskiy, and F. d’Alché Buc, “A framework to learn with interpretation,” *NeurIPS*, vol. 34, 2021.
- [93] A. A. Ismail, H. Corrada Bravo, and S. Feizi, “Improving deep learning interpretability by saliency guided training,” *NeurIPS*, vol. 34, 2021.
- [94] J. V. Jeyakumar, J. Noor, Y.-H. Cheng, L. Garcia, and M. Srivastava, “How can i explain this to you? an empirical study of deep neural network explanation methods,” *NeurIPS*, vol. 33, pp. 4211–4222, 2020.
- [95] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in *ICML*, pp. 3145–3153, JMLR. org, 2017.
- [96] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, “Generating visual explanations,” in *ECCV*, pp. 3–19, Springer, 2016.
- [97] D. A. Melis and T. Jaakkola, “Towards robust interpretability with self-explaining neural networks,” in *NIPS*, pp. 7775–7784, 2018.
- [98] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *arXiv preprint arXiv:1312.6034*, 2013.
- [99] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” *PLoS one*, vol. 10, no. 7, p. e0130140, 2015.
- [100] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, “A unified view of gradient-based attribution methods for deep neural networks,” in *NIPS Workshop*, ETH Zurich, 2017.
- [101] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks,” in *WACV*, pp. 839–847, IEEE, 2018.
- [102] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *CVPR*, pp. 2921–2929, 2016.
- [103] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *ICCV*, pp. 618–626, 2017.
- [104] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, “Score-cam: Score-weighted visual explanations for convolutional neural networks,” in *CVPR*, pp. 24–25, 2020.
- [105] P. Wang and N. Vasconcelos, “A machine teaching framework for scalable recognition,” in *ICCV*, pp. 4945–4954, 2021.

- [106] A. Dhurandhar and K. Shanmugam, “Counterfactual vs contrastive explanations in artificial intelligence,” in *towardsdatascience*, 2020.
- [107] A. Korikov, A. Shleyfman, and C. Beck, “Counterfactual explanations for optimization-based decisions in the context of the gdpr,” in *ICAPS workshop*, 2021.
- [108] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das, “Explanations based on the missing: Towards contrastive explanations with pertinent negatives,” in *NIPS*, pp. 592–603, 2018.
- [109] I. Stepin, J. M. Alonso, A. Catala, and M. Pereira-Fariña, “A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence,” *IEEE Access*, vol. 9, pp. 11974–12001, 2021.
- [110] S. Rathi, “Generating counterfactual and contrastive explanations using shap,” *arXiv preprint arXiv:1906.09293*, 2019.
- [111] T. Tsiligkaridis, “Failure prediction by confidence estimation of uncertainty-aware dirichlet networks,” in *ICASSP*, pp. 3525–3529, IEEE, 2021.
- [112] C. Corbière, N. Thome, A. Saporta, T.-H. Vu, M. Cord, and P. Perez, “Confidence estimation via auxiliary models,” *T-PAMI*, 2021.
- [113] Y. Geifman and R. El-Yaniv, “Selective classification for deep neural networks,” in *NIPS*, pp. 4878–4887, 2017.
- [114] X. Wang, Y. Luo, D. Crankshaw, A. Tumanov, F. Yu, and J. E. Gonzalez, “Idk cascades: Fast deep learning by learning not to overthink,” *arXiv preprint arXiv:1706.00885*, 2017.
- [115] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, “Counterfactual visual explanations,” in *ICML* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 2376–2384, 2019.
- [116] W. Wu, Y. Su, X. Chen, S. Zhao, I. King, M. R. Lyu, and Y.-W. Tai, “Towards global explanations of convolutional neural networks with concept attribution,” in *CVPR*, pp. 8652–8661, 2020.
- [117] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar, “On completeness-aware concept-based explanations in deep neural networks,” *NeurIPS*, vol. 33, pp. 20554–20565, 2020.
- [118] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim, “Towards automatic concept-based explanations,” *NeurIPS*, vol. 32, 2019.
- [119] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, “This looks like that: deep learning for interpretable image recognition,” *NeurIPS*, vol. 32, 2019.

- [120] P. Hase, C. Chen, O. Li, and C. Rudin, “Interpretable image recognition with hierarchical prototypes,” in *HCOMP*, vol. 7, pp. 32–40, 2019.
- [121] B. Kim, R. Khanna, and O. O. Koyejo, “Examples are not enough, learn to criticize! criticism for interpretability,” *NIPS*, vol. 29, 2016.
- [122] P. Rodríguez, M. Caccia, A. Lacoste, L. Zamparo, I. Laradji, L. Charlin, and D. Vazquez, “Beyond trivial counterfactual explanations with diverse valuable explanations,” in *ICCV*, pp. 1056–1065, 2021.
- [123] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual explanations without opening the black box: Automated decisions and the gpdr,” *Harv. JL & Tech.*, vol. 31, p. 841, 2017.
- [124] A. Jacovi, S. Swayamdipta, S. Ravfogel, Y. Elazar, Y. Choi, and Y. Goldberg, “Contrastive explanations for model interpretability,” *arXiv preprint arXiv:2103.01378*, 2021.
- [125] L. Anne Hendricks, R. Hu, T. Darrell, and Z. Akata, “Grounding visual explanations,” in *ECCV*, September 2018.
- [126] K. H. Lee, C. Park, J. Oh, and N. Kwak, “Lfi-cam: Learning feature importance for better visual explanation,” in *ICCV*, pp. 1355–1363, 2021.
- [127] D. Lim, H. Lee, and S. Kim, “Building reliable explanations of unreliable neural networks: locally smoothing perspective of model interpretation,” in *CVPR*, pp. 6468–6477, 2021.
- [128] Y. Wang and X. Wang, “Self-interpretable model with transformation equivariant interpretation,” in *NeurIPS*, vol. 34, 2021.
- [129] M. Bohle, M. Fritz, and B. Schiele, “Convolutional dynamic alignment networks for interpretable classifications,” in *CVPR*, pp. 10029–10038, 2021.
- [130] Z. Huang and Y. Li, “Interpretable and accurate fine-grained recognition via region grouping,” in *CVPR*, pp. 8662–8672, 2020.
- [131] T. Fel, R. Cadène, M. Chalvidal, M. Cord, D. Vigouroux, and T. Serre, “Look at the variance! efficient black-box explanations with sobol-based sensitivity analysis,” *NeurIPS*, vol. 34, 2021.
- [132] E. Tjoa and C. Guan, “A survey on explainable artificial intelligence (xai): Toward medical xai,” *T-NNLS*, vol. 32, no. 11, pp. 4793–4813, 2020.
- [133] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, “Explaining explanations: An overview of interpretability of machine learning,” in *DSAA*, pp. 80–89, IEEE, 2018.

- [134] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, “Explainable ai: A review of machine learning interpretability methods,” *Entropy*, vol. 23, no. 1, p. 18, 2020.
- [135] F. Bodria, F. Giannotti, R. Guidotti, F. Naretto, D. Pedreschi, and S. Rinzivillo, “Benchmarking and survey of explanation methods for black box models,” *arXiv preprint arXiv:2102.13076*, 2021.
- [136] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic attribution for deep networks,” in *ICML*, pp. 3319–3328, JMLR. org, 2017.
- [137] S. Srinivas and F. Fleuret, “Full-gradient representation for neural network visualization,” in *NeurIPS*, 2019.
- [138] D. Lewis, “Counterfactuals and comparative possibility,” in *IFS*, pp. 57–85, Springer, 1973.
- [139] J. Woodward, *Making things happen: A theory of causal explanation*. Oxford university press, 2005.
- [140] M. Pawelczyk, K. Broelemann, and G. Kasneci, “Learning model-agnostic counterfactual explanations for tabular data,” in *WWW*, pp. 3126–3132, 2020.
- [141] R. K. Mothilal, A. Sharma, and C. Tan, “Explaining machine learning classifiers through diverse counterfactual explanations,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 607–617, 2020.
- [142] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata, “Generating counterfactual explanations with natural language,” *arXiv preprint arXiv:1806.09809*, 2018.
- [143] C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud, “Explaining image classifiers by counterfactual generation,” *arXiv preprint arXiv:1807.08024*, 2018.
- [144] T. Deruyttere, S. Vandenhende, D. Grujicic, L. Van Gool, and M.-F. Moens, “Talk2car: Taking control of your self-driving car,” *arXiv preprint arXiv:1909.10838*, 2019.
- [145] A. Van Looveren and J. Klaise, “Interpretable counterfactual explanations guided by prototypes,” *arXiv preprint arXiv:1907.02584*, 2019.
- [146] S. Liu, B. Kailkhura, D. Loveland, and Y. Han, “Generative counterfactual introspection for explainable deep learning,” *arXiv preprint arXiv:1907.03077*, 2019.
- [147] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, “Visualizing deep neural network decisions: Prediction difference analysis,” *ICLR*, 2017.
- [148] O. Lang, Y. Gandelsman, M. Yarom, Y. Wald, G. Elidan, A. Hassidim, W. T. Freeman, P. Isola, A. Globerson, M. Irani, *et al.*, “Explaining in style: Training a gan to explain a classifier in stylespace,” in *ICCV*, pp. 693–702, 2021.

- [149] J. Thiagarajan, V. S. Narayanaswamy, D. Rajan, J. Liang, A. Chaudhari, and A. Spanias, “Designing counterfactual generators using deep model inversion,” *NeurIPS*, vol. 34, 2021.
- [150] Y. Zhao, “Fast real-time counterfactual explanations,” *arXiv preprint arXiv:2007.05684*, 2020.
- [151] R. Luss, P.-Y. Chen, A. Dhurandhar, P. Sattigeri, K. Shanmugam, and C.-C. Tu, “Generating contrastive explanations with monotonic attribute functions,” *arXiv preprint arXiv:1905.12698*, 2019.
- [152] D. Nemirovsky, N. Thiebaut, Y. Xu, and A. Gupta, “CounterGAN: Generating realistic counterfactuals with residual generative adversarial nets,” *arXiv preprint arXiv:2009.05199*, 2020.
- [153] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, pp. 2672–2680, 2014.
- [154] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *ICCV*, December 2015.
- [155] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *NIPS*, pp. 4765–4774, 2017.
- [156] I. Lage, E. Chen, J. He, M. Narayanan, B. Kim, S. Gershman, and F. Doshi-Velez, “An evaluation of the human-interpretability of explanation,” *arXiv preprint arXiv:1902.00006*, 2019.
- [157] M. Yang and B. Kim, “Benchmarking attribution methods with relative feature importance,” *arXiv preprint arXiv:1907.09701*, 2019.
- [158] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, “Evaluating the visualization of what a deep neural network has learned,” *T-NNLS*, vol. 28, no. 11, pp. 2660–2673, 2016.
- [159] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, “Evaluating feature importance estimates,” 2018.
- [160] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, “Towards better understanding of gradient-based attribution methods for deep neural networks,” *arXiv preprint arXiv:1711.06104*, 2017.
- [161] R. C. Fong and A. Vedaldi, “Interpretable explanations of black boxes by meaningful perturbation,” in *ICCV*, pp. 3429–3437, 2017.
- [162] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, “Top-down neural attention by excitation backprop,” *IJCV*, vol. 126, no. 10, pp. 1084–1102, 2018.



- [163] N. Bansal, C. Agarwal, and A. Nguyen, “Sam: The sensitivity of attribution methods to hyperparameters,” in *CVPR*, pp. 8673–8683, 2020.
- [164] C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, and P. K. Ravikumar, “On the (in) fidelity and sensitivity of explanations,” *NeurIPS*, vol. 32, 2019.
- [165] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, “The (un) reliability of saliency methods,” in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 267–280, Springer, 2019.
- [166] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, “Sanity checks for saliency maps,” in *NIPS*, pp. 9505–9515, 2018.
- [167] D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” *ICLR*, 2017.
- [168] J. Yang, H. Wang, L. Feng, X. Yan, H. Zheng, W. Zhang, and Z. Liu, “Semantically coherent out-of-distribution detection,” in *ICCV*, pp. 8301–8309, 2021.
- [169] K. Tang, D. Miao, W. Peng, J. Wu, Y. Shi, Z. Gu, Z. Tian, and W. Wang, “Codes: Chamfer out-of-distribution examples against overconfidence issue,” in *ICCV*, pp. 1153–1162, 2021.
- [170] A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah, “Out-of-distribution detection using union of 1-dimensional subspaces,” in *CVPR*, pp. 9452–9461, 2021.
- [171] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, “Toward open set recognition,” *T-PAMI*, vol. 35, no. 7, pp. 1757–1772, 2012.
- [172] A. Bendale and T. E. Boult, “Towards open set deep networks,” in *CVPR*, pp. 1563–1572, 2016.
- [173] S. Kong and D. Ramanan, “Opengan: Open-set recognition via open data generation,” in *ICCV*, pp. 813–822, 2021.
- [174] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, “Learning placeholders for open-set recognition,” in *CVPR*, pp. 4401–4410, 2021.
- [175] K. Lee, H. Lee, K. Lee, and J. Shin, “Training confidence-calibrated classifiers for detecting out-of-distribution samples,” *ICLR*, 2018.
- [176] D. Hendrycks, M. Mazeika, and T. G. Dietterich, “Deep anomaly detection with outlier exposure,” *ICLR*, 2019.
- [177] P. Wang and N. Vasconcelos, “Towards realistic predictors,” in *ECCV*, 2018.

- [178] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, “Caltech-UCSD Birds 200,” Tech. Rep. CNS-TR-2010-001, California Institute of Technology, 2010.
- [179] T. Miller, “Contrastive explanation: A structural-model approach,” *arXiv preprint arXiv:1811.03163*, 2018.
- [180] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should i trust you?: Explaining the predictions of any classifier,” in *SIGKDD*, pp. 1135–1144, ACM, 2016.
- [181] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, “Network dissection: Quantifying interpretability of deep visual representations,” in *CVPR*, pp. 6541–6549, 2017.
- [182] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Object detectors emerge in deep scene cnns,” *ICLR*, 2015.
- [183] V. Petsiuk, A. Das, and K. Saenko, “Rise: Randomized input sampling for explanation of black-box models,” *arXiv preprint arXiv:1806.07421*, 2018.
- [184] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” in *CVPR*, 2017.
- [185] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [186] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, pp. 1097–1105, 2012.
- [187] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, “Tell me where to look: Guided attention inference network,” in *CVPR*, pp. 9215–9223, 2018.
- [188] D. M. Endres and J. E. Schindelin, “A new metric for probability distributions,” *IEEE Transactions on Information theory*, 2003.
- [189] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie, “Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 595–604, 2015.
- [190] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S. Kamali, M. Mallocci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy, “Openimages: A public dataset for large-scale multi-label and multi-class image classification.,” *Dataset available from <https://storage.googleapis.com/openimages/web/index.html>*, 2017.

- [191] <https://www.sama.com/>
- [192] X. Zhu, “Machine teaching: An inverse problem to machine learning and an approach toward optimal education,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015.
- [193] J. Li, R. Socher, and S. C. Hoi, “Dividemix: Learning with noisy labels as semi-supervised learning,” *ICLR*, 2020.
- [194] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” *arXiv preprint arXiv:1804.06872*, 2018.
- [195] P. Wang and N. Vasconcelos, “Scout: Self-aware discriminant counterfactual explanations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8981–8990, 2020.
- [196] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, “Counterfactual visual explanations,” *ICML*, 2019.
- [197] P. Wang, K. Nagrecha, and N. Vasconcelos, “Gradient-based algorithms for machine teaching,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1387–1396, June 2021.
- [198] G. B. Schmidt and W. M. Jettinghoff, “Using amazon mechanical turk and other compensated crowdsourcing sites,” *Business Horizons*, vol. 59, no. 4, pp. 391–400, 2016.
- [199] X. Yang, Z. Song, I. King, and Z. Xu, “A survey on deep semi-supervised learning,” *arXiv preprint arXiv:2103.00550*, 2021.
- [200] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” *arXiv preprint arXiv:2003.04297*, 2020.
- [201] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” *arXiv preprint arXiv:2006.09882*, 2020.
- [202] L. Jing and Y. Tian, “Self-supervised visual feature learning with deep neural networks: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [203] X. Liu, F. Zhang, Z. Hou, Z. Wang, L. Mian, J. Zhang, and J. Tang, “Self-supervised learning: Generative or contrastive,” *arXiv preprint arXiv:2006.08218*, vol. 1, no. 2, 2020.
- [204] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, “A survey on contrastive self-supervised learning,” *Technologies*, vol. 9, no. 1, p. 2, 2021.

- [205] G. Patterson, G. Van Horn, S. Belongie, P. Perona, and J. Hays, “Tropel: Crowdsourcing detectors with minimal training,” in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 3, 2015.
- [206] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, “Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 25, pp. E5716–E5725, 2018.
- [207] <http://spc.ucsd.edu/>
- [208] H. Song, M. Kim, D. Park, and J.-G. Lee, “Learning from noisy labels with deep neural networks: A survey,” *arXiv preprint arXiv:2007.08199*, 2020.
- [209] P. Wang, K. Nagrecha, and N. Vasconcelos, “Gradient-based algorithms for machine teaching,” in *CVPR*, pp. 1387–1396, 2021.
- [210] G. Giguère and B. C. Love, “Limits in decision making arise from limits in memory retrieval,” *PNAS*, vol. 110, no. 19, pp. 7613–7618, 2013.
- [211] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, “A survey of deep active learning,” *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.
- [212] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” *arXiv preprint arXiv:1905.02249*, 2019.
- [213] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, “Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring,” *arXiv preprint arXiv:1911.09785*, 2019.
- [214] J. T. Springenberg, “Unsupervised and semi-supervised learning with categorical generative adversarial networks,” *arXiv preprint arXiv:1511.06390*, 2015.
- [215] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, “Realistic evaluation of deep semi-supervised learning algorithms,” *NeurIPS*, vol. 31, 2018.
- [216] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, “Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling,” *NeurIPS*, vol. 34, pp. 18408–18419, 2021.
- [217] A. Coates, A. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, JMLR Workshop and Conference Proceedings, 2011.

- [218] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, “Reading digits in natural images with unsupervised feature learning,” in *NIPS workshop*, vol. 2011, p. 5, 2011.
- [219] P. Wang and N. Vasconcelos, “Deliberative explanations: visualizing network insecurities,” in *NeurIPS*, 2019.
- [220] R. T. Mullapudi, F. Poms, W. R. Mark, D. Ramanan, and K. Fatahalian, “Learning rare category classifiers on a tight labeling budget,” in *ICCV*, pp. 8423–8432, 2021.
- [221] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *CVPR*, pp. 9729–9738, 2020.
- [222] J.-C. Su and S. Maji, “The semi-supervised inaturalist challenge at the fgvc8 workshop,” *arXiv preprint arXiv:2106.01364*, 2021.
- [223] P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez, “Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning,” *arXiv preprint arXiv:2001.06001*, 2020.
- [224] O. T. Nartey, G. Yang, J. Wu, and S. K. Asare, “Semi-supervised learning for fine-grained classification with self-training,” *IEEE Access*, vol. 8, pp. 2109–2121, 2019.
- [225] D. Mugnai, F. Pernici, F. Turchini, and A. D. Bimbo, “Soft pseudo-labeling semi-supervised learning applied to fine-grained visual classification,” in *ICPR*, pp. 102–110, Springer, 2021.
- [226] S. K. Asare, F. You, and O. T. Nartey, “A semisupervised learning scheme with self-paced learning for classifying breast cancer histopathological images,” *Computational Intelligence and Neuroscience*, vol. 2020, 2020.
- [227] S. K. Asare, F. You, and O. T. Nartey, “Learning to classify skin lesions via self-training and self-paced learning,” in *BIBM*, pp. 963–967, IEEE, 2020.
- [228] J. Du and C. X. Ling, “Active learning with human-like noisy oracle,” in *ICDM*, pp. 797–802, IEEE, 2010.
- [229] G. Gupta, A. K. Sahu, and W.-Y. Lin, “Learning in confusion: Batch active learning with noisy oracle,” 2019.
- [230] S. Yan, K. Chaudhuri, and T. Javidi, “Active learning from imperfect labelers,” *NIPS*, vol. 29, 2016.
- [231] S. Su, Y. Chen, O. Mac Aodha, P. Perona, and Y. Yue, “Interpretable machine teaching via feature feedback,” 2017.

- [232] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning,” *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.
- [233] Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, and R. Jin, “Dash: Semi-supervised learning with dynamic thresholding,” in *ICML*, pp. 11525–11536, PMLR, 2021.
- [234] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, and P. Vajda, “Unbiased teacher for semi-supervised object detection,” *arXiv preprint arXiv:2102.09480*, 2021.
- [235] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, “Multi-class active learning by uncertainty sampling with diversity maximization,” *IJCV*, vol. 113, no. 2, pp. 113–127, 2015.
- [236] Y. Yang and M. Loog, “Active learning using uncertainty information,” in *ICPR*, pp. 2646–2651, IEEE, 2016.
- [237] A. Holub, P. Perona, and M. C. Burl, “Entropy-based active learning for object recognition,” in *CVPR Workshops*, pp. 1–8, IEEE, 2008.