FROM LABELED TO UNLABELED DATA: UNDERSTAND DEEP VISUAL REPRESENTATIONS UNDER CAUSAL LENS

by

Yuewei Yang

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

_____
Yiran Chen, Supervisor

_____
Leslie M Collins

_____
Henry Pfister

_____
Vahid Tarokh

_____
Carlo Tomasi

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
Duke University

2023

ABSTRACT

FROM LABELED TO UNLABELED DATA: UNDERSTAND DEEP VISUAL REPRESENTATIONS UNDER CAUSAL LENS

by

Yuewei Yang

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

_____
Yiran Chen, Supervisor

_____
Leslie M Collins

_____
Henry Pfister

_____
Vahid Tarokh

_____
Carlo Tomasi

An abstract of a dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
Duke University

2023

Copyright © 2023 by Yuewei Yang
All rights reserved

Abstract

Deep vision models have been highly successful in various computer vision applications such as image classification, segmentation, and object detection. These models encode visual data into low-dimensional representations, which are then utilized in downstream tasks. Typically, the most accurate models are fine-tuned using fully labeled data, but this approach may not generalize well to different applications. Self-supervised learning has emerged as a potential solution to this issue, where the deep vision encoder is pretrained with unlabeled data to learn more generalized representations. However, the underlying mechanism governing the generalization and specificity of representations seeks more understanding. Causality is an important concept in visual representation learning as it can help improve the generalization of models by providing a deeper understanding of the underlying relationships between features and objects in the visual world.

Through works presented in this dissertation, we provide a causal interpretation of the mechanism underlying deep vision models' ability to learn representations in both labeled and unlabeled environments and improve the generalization and the specificity of extracted representations through the interpreted causal factors. Specifically, we tackle the problem from 4 aspects: Causally Interpret Supervised Deep Vision Models; Supervised Learning with Underlabeled Data; Self-supervised Learning with Unlabeled Data; Causally Understand Unsupervised Visual Representation Learning.

Firstly, we interpret the prediction of a deep vision model by identifying causal pixels in the input images via 'inversing' the model weights. Secondly, we recognise the challenges of learning an accurate object detection model with missing labels in the dataset and we address this underlabel data issue by adapting positive-unlabeled learning approach instead of the positive-negative approach. Thirdly, we focus on

improving both generalization and specificity of unsupervised representations based on prior causal relations; Finally, we enhance the stability of the unsupervised representations during the inference by intervening data variables under a well constructed causal framework.

We establish a causal relationship between deep vision models and their input/output for different applications with (partially) labeled data, and strengthen generalized representations through extensive analytical understanding of unsupervised representation learning under various hypothesized causal frameworks.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Yiran Chen, for his invaluable guidance, unwavering support, and boundless knowledge through my doctoral studies. His encouragement and constructive feedback have been critical in shaping my research direction and in guiding me towards excellence.

I would also like to thank the members of my dissertation committee (in alphabetical order), Professor Leslie Collins, Professor Vahid Tarokh, Professor Henry Pfister and Professor Carlo Tomasi, for their insightful feedback and invaluable suggestions during the evaluation of my preliminary thesis and final dissertation.

This Ph.D. study has been an exciting and rewarding journey, thanks to the many individuals who have contributed in various ways. I am grateful for all the talented collaborators: Kevin Liang, Dong Wang, Chenyang Tao, Zhe Gan, Jingwei Sun, Ang Li, and many other collaborators. Without their valuable input, this research would not have been possible.

I extend my heartfelt thanks to my family and friends for their unwavering love and support, and for their encouragement throughout my academic journey. Especially to my dear wife who has supported unconditionally with my life through all these years and I owe my deepest devotion to her.

Contents

Abstract **iv**

Acknowledgements **vi**

List of Tables **xi**

List of Figures **xii**

1 Introduction **1**

1.1 Causally Interpret Supervised Vision Models 2

1.2 Supervised Learning with Underlabeled Data 2

1.3 Unsupervised Learning with Unlabeled Data 3

1.4 Causally Understand Unsupervised Visual Representation Learning . 3

2 Proactive Pseudo-Intervention: Pre-informed Contrastive Learning For Interpretable Vision Models **4**

2.1 Introduction 4

2.2 Background 7

2.3 Proactive Pseudo-Intervention 10

2.3.1 Synthetic causal interventions for contrasts 12

2.3.2 Saliency Weight Backpropagation 14

2.4 Experiments 17

2.4.1 Natural Image Datasets 18

2.4.2 OCT-GA: Geographic Atrophy Classification 20

2.4.3 LIDC-IDRI: Lung Lesions Classification 22

2.5 Conclusions 24

2.6 Appendix 26

2.6.1 Derivation of Convolutional Weight Backpropagation 26
2.6.2 Details on Causal Masking 27
2.6.3 Related Work 27
2.6.4 CUB Experiment Details 32
2.6.5 Geographic Atrophy (GA) Experiment Details 33
2.6.6 LIDC Experiment Details 36
2.6.7 Additional Saliency Map Comparisons 37

3 Object Detection as a Positive-Unlabeled Problem 41

3.1 Introduction 41
3.2 Methods 43
3.2.1 Faster R-CNN 43
3.2.2 PU learning 45
3.2.3 PU learning for object detection 47
3.3 Related work 49
3.4 Experiments 50
3.4.1 Hand-tuning versus estimation of π_p 51
3.4.2 PU versus PN on PASCAL VOC and MS COCO 52
3.4.3 Visual Genome 54
3.4.4 DeepLesion 55
3.5 Conclusion and future work 56
3.6 Appendix 58
3.6.1 Example forgetting in object detection 58
3.6.2 Recall of Region Proposals 60
3.6.3 OpenImages 61

4 Federated Unsupervised Visual Representation Learning via Exploiting General Content and Personal Style **63**

4.1 Introduction 63

4.2 Related work 67

4.3 Federated with Style (FedStyle) 69

4.3.1 Local style extraction 71

4.3.2 Local style infusion for contrastive learning 73

4.3.3 Personalizing with stylized content feature 73

4.3.4 Analysis of style-infused representation 75

4.4 Experiments 76

4.5 Evaluations and Results 77

4.5.1 Ablation studies 80

4.6 Conclusion 83

4.7 Appendix 84

4.7.1 Information on datasets 84

4.7.2 Non-IID sampling 84

4.7.3 Adapting personalized methods for federated unsupervised representation learning 84

4.7.4 Figures for experiment results 85

5 Stable and Causal Inference for Discriminative Self-supervised Deep Visual Representations **90**

5.1 Introduction 90

5.2 Related Work 93

5.3 Theory 95

5.3.1 Problem formulation 95

5.3.2 Alignment in discriminative SSL 97

5.3.3 Transformation of the ground-truth factors is orthogonal to the applied augmentations 100

5.3.4 Reason and solutions for unstable change in data variables . . 102

5.4 Experiments and Discussions 104

5.4.1 Causal3DIdent 104

5.4.2 ImageNet 109

5.5 Conclusions 113

5.6 Appendix 114

5.6.1 Recent Descriminative SSL Formulations 114

5.6.2 Extended Theory and Proofs 116

5.6.3 Causal3DIdent 120

5.6.4 ImageNet 123

6 Conclusions **128**

Bibliography **131**

Biography **159**

List of Tables

2.1 WBP update rules for common transformations. 16

2.2 PPI classification performance. 19

2.3 AUC results on GA prediction. 23

2.4 LIDC-IDRI classification AUC results. 24

2.5 AUC on LIDC from different causal masking methods 27

2.6 A list of commonly used LRP rules.[MBL+19] 30

2.7 Accuracy on CUB 32

3.1 Detector performance on Visual Genome. 54

3.2 Detector performance on a subset of OpenImages at various IoU thresholds. 62

4.1 Effect of λ on representations. 82

4.2 Effect of number of layers of G on representations. 83

4.3 Dataset information 84

5.1 The effect of $\mathbf{F}$ on both stable and unstable samples.The accuracy is average over 3 random seeds. 109

5.2 Comparison of inferring different $\mathbf{F}$ on ObjectNet with a target data variable random trials. 112

5.3 Notations for important elements in SSL. 114

5.4 Hyperparameters for SSL methods in training Causal3dIdent. 121

5.5 Accuracy discrepancy between seen and unseen distributions. 121

List of Figures

2.1 Comparison between WBP and LRP. 7

2.2 Illustration of the proposed PPI learning strategy. 11

2.3 Visualization of the inferred saliency maps. Left: CUB dataset (PPI is based on WBP). Right: ImageNet dataset. 15

2.4 Quantitative evaluations of causal relevance of competing saliency maps (higher is better). 20

2.5 Saliency maps on GA dataset. 22

2.6 Saliency maps on LIDC-IDR. Saliency maps of PPI+WBP are mostly consistent with the ground truths. 23

2.7 Illustration of a 3D OCT image example. 33

2.8 OCT slice examples from 4 site. 34

2.9 Illustration of multi-view CNN based 3D OCT image classification model. 35

2.10 More comparisons of saliency maps on CUB dataset. 38

2.11 Examples of saliency maps on GA dataset based on model trained with and without PPI. 39

2.12 Examples of saliency maps on LIDC dataset based on model trained with and without PPI. 40

3.1 Datasets with missing annotations. 42

3.2 Illustration of PN and PU with missing labels. 44

3.3 Dynamic positive prior estimation. 48

3.4 Estimated and hard-coded positive priors. 51

3.5 mAP at IoU 0.5 (AP_{50}) on (a) PASCAL VOC and (b) MS COCO, for a range of label missingness ρ. 52

3.6 Sensitivities on Deep Lesion. 55

3.7 Examples of forgetting in object detection. 58

3.8 Detection rates of objects before and after training. 60

3.9 Recall of top 500 proposals from RPN after training on PASCAL VOC2007 when $\rho = 0.5$. 61

4.1 Federated unsupervised representation learning with stylized user data. 64

4.2 Intuition of style infusion to improve representations via contrastive learning. 65

4.3 FedStyle Process. 70

4.4 Generalization and Personalization at single client. 78

4.5 Generalization and Personalization at multiple clients. 79

4.6 t-SNE of extracted style feature distributions. 80

4.7 Improvement of introducing stylized features for personal downstream tasks. 81

4.8 The effect of style extraction epochs E_s on the performance. 82

4.9 Data distributions for non-IID partiotion. 85

4.10 Personalization v.s. Generalization for all methods with single client in each style dataset. 88

4.11 SimSiam at multiple clients. 89

5.1 Unstable inference. 91

5.2 A causal graph for data generation process. 96

5.3 Causal graph for V in Causal3DIdent. 104

5.4 Deterioration of unstable changing in data variables on all SSL. . . . 106

5.5 By identifying and passing top k % important dimensions, the deterioration in both accuracy and prediction score is significantly alleviated. 108

5.6 Examples of ObjectNet dataset (1st row), Stylized ImageNet dataset (2nd row), Synthetic dataset (3rd row). 110

5.7 By identifying and passing top k % important dimensions, the deterioration in both accuracy and prediction score is alleviated. 112

5.8 Examples of images in Causal3DIdent. 120

5.9 Examples of sampling intervened data samples. Before sampling and After sampling. 122

5.10 Deterioration of unstable changing in data variables on all SSL. Only 5 children nodes are intervened. 123

5.11 Illustration of ObjectNet controlling data variables. 124

5.12 Examples of Stylized ImageNet 125

5.13 Training epochs on Solution 2. 127

Chapter 1

Introduction

Visual representation learning has been a critical topic in computer vision, where the aim is to learn high-level, abstract representations of visual data that can be used for various downstream tasks, such as object recognition, image retrieval, and scene understanding[ZCS+23, LRS+19, BCV13]. The performance of such tasks heavily relies on the quality of the learned representations, and researchers have long sought to develop effective methods to learn them.

The two main approaches to visual representation learning are supervised and unsupervised learning. In supervised learning, a large dataset of labeled examples is used to train a model to map input data to output labels. In contrast, unsupervised learning aims to learn useful representations of the input data without explicit labels or supervision. Instead, the model must discover the underlying structure of the data through self-supervision or other indirect forms of supervision.

The concept of causality plays a crucial role in understanding how deep vision models learn representations in both labeled and unlabeled environments. In this Ph.D. final dissertation, I will formulate causal frameworks and discuss how to learn and interpret the representations with labeled, partially labeled, and unlabeled data in two major computer vision applications: image classifications and object detection.

1.1 Causally Interpret Supervised Vision Models

Supervised deep learning techniques have enabled vision models to achieve performance in image classification tasks that is comparable to human performance. Causally Interpreting these supervised predictions is to understand the underlying causal mechanisms of deep learning models, with the goal of improving model interpretability, robustness, and generalization. In Chapter 2, we discuss the limitations of deep vision models in interpreting their decision rules and our proposed strategy that leverages proactive interventions to guard against non-causal visual cues and a pre-informed salience mapping module to identify key image pixels to intervene. We demonstrate the utility of our proposal on natural and medical image datasets to produce more meaningful saliency maps.

1.2 Supervised Learning with Underlabeled Data

After causally relating the output and the input of deep learning models, we understand that the quality of labels is the critical factor in the performance of deep learning models. In many cases, it is challenging to obtain complete annotations for all object instances, resulting in underlabeled datasets. These underlabeled datasets impose considerable performance deficiency, especially in object detection datasets[XBG+19]. In Chapter 3, we discuss the challenges of constituting complete annotations for object detection in complex image scenes and propose a Positive-Unlabeled method to improve the learned visual representation to alleviate the drop in detection performance when there are missing labels.

1.3 Unsupervised Learning with Unlabeled Data

In situations where there are no labels associated with data, learning generalized representations from unlabeled data has become a popular research topic[LZH+21, LKHS20]. While this has been well-supported in centralized learning settings, there is a better need to sustain both high generalization and high specificity in decentralized learning settings. In Chapter 4, a novel federated learning method is presented that can learn a more generalized global model and induce personalized local models in a distributed system with private and heterogeneous client data, and it is demonstrated that this method outperforms both generalization and personalization baseline methods in a stylized decentralized setting.

1.4 Causally Understand Unsupervised Visual Representation Learning

Without a strong guidance from labels, the underlying causal factors cannot be easily identified in a unsupervised learning model. Causal inference is an emerging area of research in self-supervised learning, where the goal is to learn representations that capture causal relationships between features of the data. In Chapter 5, we analyze descriminative self-supervised learning from a causal perspective to explain unstable behaviors and propose solutions to overcome them by applying their solutions during the inference process to improve time efficiency.

Chapter 2

Proactive Pseudo-Intervention: Pre-informed Contrastive Learning For Interpretable Vision Models

2.1 Introduction

Deep neural networks hold great promise in applications requiring the analysis and comprehension of complex imagery. Recent advances in hardware, network architectures, and model optimization, along with the increasing availability of large-scale annotated datasets [KH+09, Den12, DDS+09a], have enabled these models to match and sometimes outperform human experts on a number of tasks, including natural image classification [KSH17], objection recognition [GDDM14a], disease diagnosis [Saj06], and autonomous driving [CSKX15], among others.

While deep learning solutions have been positively recognized for their ability to learn *black-box* models in a purely data driven manner, their very nature makes them less credible for their inability to communicate the reasoning for making predictions in a way that is comprehensible to humans [HEKK19, RFJV20]. This denies consequential applications where the reliability and trustworthiness of a prediction are of primary concern and require expert audit, *e.g.*, in healthcare [Saj06]. To stimulate widespread use of deep learning models, a means of interpreting predictions is necessary. However, model interpretation techniques often reveal a concerning fact, that deep learning models tend to assimilate spurious correlations that do not necessarily capture the causal relationship between the input (image) and output (label) [WWD+20]. This issue is particularly notable in small-sample-size (weak supervision) scenarios or

when the sources of non-informative variation are overwhelming, thus likely to cause severe overfitting. These can lead to catastrophic failures on deployment [FHYF19, WWK+19].

A growing recognition of the issues associated with the lack of interpretable predictions is well documented in recent years [AGM+18, HEKK19, RFJV20]. Such phenomenon has energized researchers to actively seek creative solutions. Among these, two streams of work, namely *saliency mapping* [ZZZ+18, SVZ13, DG17] and *causal representation learning* (CRL) [JSS16, WHZS20, ABGLP19], stand out as some of the most promising directions. Specifically, saliency mapping encompasses techniques for *post hoc* visualizations on the input (image) space to facilitate the interpretation of model predictions. This is done by projecting the key features used in prediction back to the input space, resulting in the commonly known *saliency maps*. Importantly, these maps do not directly contribute to model learning. Alternatively, CRL solutions are built on the principles of establishing invariance from the data, and it entails teasing out sources of variation that are spuriously associated with the model output (labels). CRL models, while emphasizing the differences between causation and correlation, are not subject to the rigor of causal inference approaches, because their goal is not to obtain accurate causal effect estimates but rather to produce robust models with better generalization ability relative to their naively learned counterparts [ABGLP19].

In this work, we present *Proactive Pseudo-Intervention* (PPI), a solution that accounts for the needs of causal representation identification and visual verification. Our key insight is the derivation of pre-informed saliency maps which facilitate visual verification of model predictions *and* enable learning that is robust to (non-causal) associations. While true causation can only be established through experimental interventions, we leverage tools from contrastive representation learning to synthesize

pseudo-interventions from observational data. Our procedure is motivated by the causal argument: perturbing the non-causal features will not change the target label.

To motivate, in Figure 2.1 we present an example to illustrate the benefits of producing causally-informed saliency maps. In this scenario, the task is to classify two bird species (A and B) in the wild. Due to the differences in their natural habitats, A-birds are mostly seen resting on rocks, while B-birds are more commonly found among bulrushes. A deep model, trained naively, will tend to associate the background characteristics with the labels, knowing these strongly correlate with the bird species (labels) in the training set. This is confirmed by the saliency maps derived from the layer-wise relevance propagation (LRP) techniques [BBM+15]: the model also attends heavily on the background features, while the difference in bird anatomy is what causally determines the label. If we were provided with an image of a bird in an environment foreign to the images in the training set, the model will be unable to make a reliable prediction, thus causing robustness concerns. This generalization issue worsens with a smaller training sample size. On the other hand, saliency maps from our PPI-enhanced model successfully focus on the bird anatomy, and thus will be robust to environmental changes captured in the input images.

PPI addresses causally-informed reasoning, robust learning, and model interpretation in a unified framework. A new saliency mapping method, named *Weight Back Propagation* (WBP), is also proposed to generate more concentrated intervention mask for PPI training. The key contributions of this paper include:

- An end-to-end contrastive representation learning strategy PPI that employs proactive interventions to identify causally relevant features.
- A fast and architecture-agnostic saliency mapping module WBP that delivers better visualization and localization performance.

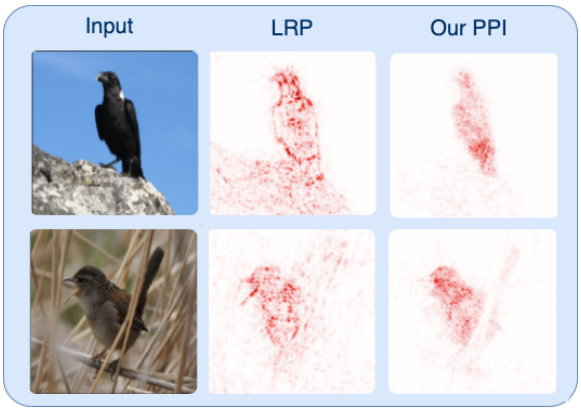


Figure 2.1: Interpretation for the bird-classification models using saliency maps generated by LRP (*layer-wise relevance propagation*) and our model PPI. LRP shows that naively trained deep model makes decisions based on the background cues (habitat, *e.g.*, rocks, bulrushes) that are spuriously correlated with the bird species, while our pre-informed PPI mostly focuses on the bird anatomy, that generalizes beyond the natural habitat.

- Experiments demonstrating significant performance boosts from integrating PPI and WBP relative to competing solutions, especially on out-of-domain predictions, data integration with heterogeneous sources and model interpretation.

2.2 Background

Visual Explanations. Saliency mapping collectively refers to a family of techniques to understand and interpret black-box image classification models, such as deep neural networks [AGM+18, HEKK19, RFJV20]. These methods project the model understanding of the targets, *i.e.*, labels, and their predictions back to the input space, which allows for the visual inspection of automated reasoning and for the communication of predictive visual cues to the user or human expert, aiming to shed model insights or to build trust for deep-learning-based systems.

In this study, we focus on *post hoc* saliency mapping strategies, where saliency maps are constructed given an arbitrary prediction model, as opposed to relying on

customized model architectures for interpretable predictions [FHYF19, WWK+19], or to train a separate module to explicitly produce model explanations [FHYF19, GWE+19, CCGD18, FV17, SGK17]. Popular solutions under this category include: activation mapping [ZKL+16, SCD+17], input sensitivity analysis [SGK17], and relevance propagation [BBM+15]. Activation mapping based methods fail at visualizing fine-grained evidence, which is particularly important in explaining medical classification models [DLSH18, SCD+17, WKG+19]. Input sensitivity analysis based methods produce fine-grained saliency maps. However, these maps are generally less concentrated [DG17, FV17] and less interpretable. Relevance propagation based methods, like LRP and its variants, use complex rules to prioritize positive or large relevance, making the saliency maps visually appealing to human. However, our experiments demonstrate that LRP and its variants highlight spuriously correlated features (boarderlines and backgrounds). By contrast, our WBP backpropagates the weights through layers to compute the contributions of each input pixel, which is truly faithful to the model, and WBP tends to highlight the target objects themselves rather than the background. At the same time, the simplicity and efficiency makes WBP easily work with other advanced learning strategies for both model diagnosis and improvements during training.

Our work is in a similar spirit to [FV17, DG17, CCGD18, WKG+19], where meaningful perturbations have been applied to the image during model training, to improve prediction and facilitate interpretation. Poineering works have relied on user supplied "ground-truth" explainable masks to perturb [RHDV17, LWP+18, RSMY20], however such manual annotations are costly and hence rarely available in practice. Alternatively, perturbations can be computed by solving an optimization for each image. Such strategies are costly in practice and also do not effectively block spurious features. Very recently, exploratory effort has been made to leverage tools from

counterfactual reasoning [GWE+19] and causal analysis [OCC+20] to derive visual explanations, but do not lend insights back to model training. Our work represents a fast, principled solution that overcomes the above limitations. It automatically derives explainable masks faithful to the model and data, without explicit supervision from user-generated explanations.

Contrastive Learning. There has been growing interest in exploiting contrastive learning (CL) techniques for representations learning [OLV18a, CKNH20, HFW+20, KTW+20, TKI19]. Originally devised for density estimation [GH10], CL exploits the idea of *learning by comparison* to capture the subtle features of data, *i.e.*, positive examples, by contrasting them with negative examples drawn from a carefully crafted noise distribution. These techniques aim to avoid representation collapse, or to promote representation consistency, for downstream tasks. Recent developments, both empirical and theoretical, have connected CL to information-theoretic foundations [TKI19, GSA+20], thus establishing them as a suite of *de facto* solutions for unsupervised representation learning [CKNH20, HFW+20].

The basic form of CL is essentially a binary classification task specified to discriminate positive and negative examples. In such a scenario, the binary classifier is known as the critic function. Maximizing the discriminative power wrt the critic and the representation sharpens the feature encoder. Critical to the success of CL is the choice of appropriate noise distribution, where the challenging negatives, *i.e.*, those negatives that are more similar to positive examples, are often considered more effective contrasts. In its more generalized form, CL can naturally repurpose the predictor and loss functions without introducing a new critic [TKI19]. Notably, current CL methods are not immune to spurious associations, a point we wish to improve in this work.

Causality and Interventions. From a causality perspective, humans learn via

actively interacting with the environment. We intervene and observe changes in the outcome to infer causal dependencies. Machines instead learn from static observations that are unable to inform the structural dependencies for causal decisions. As such, perturbations to the external factors, *e.g.*, surroundings, lighting, viewing angles, may drastically alter machine predictions, while human recognition is less susceptible to such nuisance variations. Such difference is best explained with the *do*-notation [Pea09a].

Unfortunately, carrying out real interventional studies, *i.e.*, randomized control trials, to intentionally block non-causal associations, is oftentimes not a feasible option for practical considerations, *e.g.*, due to cost and ethics. This work instead advocates the application of synthetic interventions to uncover the underlying causal features from observational data. Specifically, we proactively edit x and its corresponding label y in a data-driven fashion to encourage the model to learn potential causal associations. Our proposal is in line with the growing appreciation for the significance of establishing causality in machine learning models [Sch19]. Via promoting invariance [ABGLP19], such causally inspired solutions demonstrate superior robustness to superficial features that do not generalize [WHLX19]. In particular, [SMSB19, ZZL20] showed the importance and effectiveness of accounting for interventional perspectives. Our work brings these causal views to construct a simple solution that explicitly optimizes visual interpretation and model robustness.

2.3 Proactive Pseudo-Intervention

Below, we describe the construction of *Proactive Pseudo-Intervention* (PPI), a causally-informed contrastive learning scheme that seeks to simultaneously improve the accuracy, robustness, generalization and interpretability of deep-learning-based computer vision models.

The PPI learning strategy, schematically summarized in Figure 2.2, consists of three main components: (*i*) a saliency mapping module that highlights causally relevant features; (*ii*) an intervention module that synthesizes contrastive samples; and (*iii*) the prediction module, which is standard in recent vision models, *e.g.*, VGG [SZ14], ResNet [HZRS16a], and Inception Net [SVI+16]. Motivated by the discussions from our introduction, PPI establishes a feedback loop between the saliency map module and the prediction module, which is interfaced by the synthesized contrastive examples in the intervention module. Under this configuration, the prediction module is encouraged to modify its predictions only when provided with causally-relevant synthetic interventions. Note that components (*i*) and (*ii*) do not involve any additional parameters or neural network modules, which makes our strategy readily applicable to the training of virtually any computer vision task without major customization. Details of these building blocks are given below.

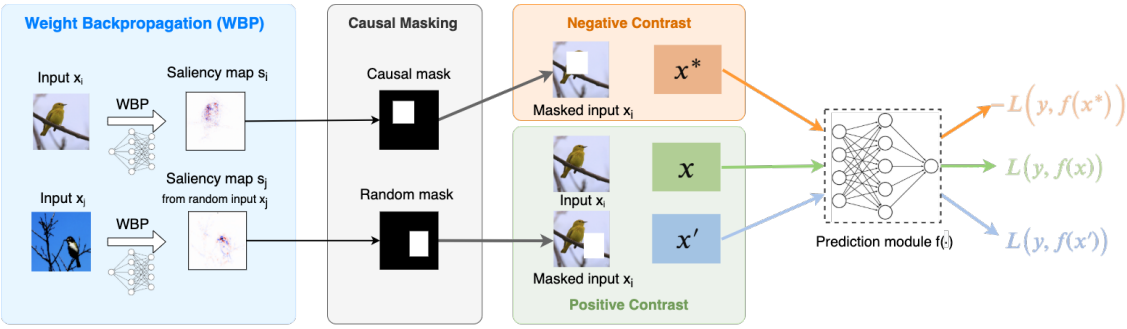


Figure 2.2: Illustration of the proposed PPI learning strategy. Input images are intervened by removing the saliency map based masks, which alters the input label (*e.g.*, negative control). For positive contrast, we use the original input as well as an input masked with a random slaiency map. We use WBP for the generation of saliency maps.

2.3.1 Synthetic causal interventions for contrasts

Key to our formulation is the design of a synthetic intervention strategy that generates contrastive examples to reinforce causal relevance during model training. Given a causal saliency map $\boldsymbol{s}_m(\boldsymbol{x})$ for an input $\boldsymbol{x}$ wrt label $y = m$, where $m = 1, \ldots, M$, and M is the number of classes, the synthetic intervention consists of removing (replacing with zero) the causal information from $\boldsymbol{x}$ contained in $\boldsymbol{s}_m(\boldsymbol{x})$, and then using it as the contrastive learning signal.

For now, let us assume the causal salience map $\boldsymbol{s}_m(\boldsymbol{x})$ is known; the procedure to obtain the saliency map will be addressed in the next section. For notational clarity, we use subscript i to denote entities associated with the i-th training sample, and omit the dependency on learnable parameters. To remove causal information from $\boldsymbol{x}_i$ and obtain a negative contrast $\boldsymbol{x}_i^*$, we apply the following *soft-masking*

$$\boldsymbol{x}_i^* = \boldsymbol{x}_i - T(\boldsymbol{s}_m(\boldsymbol{x}_i)) \odot \boldsymbol{x}_i, \tag{2.1}$$

where $T(\cdot)$ is a differentiable masking function and $\odot$ denotes element-wise (Hadamard) multiplication. Specifically, we use the thresholded sigmoid for masking:

$$T(\boldsymbol{s}_m(\boldsymbol{x}_i)) = \frac{1}{1 + \exp(-\omega(\boldsymbol{s}_m(\boldsymbol{x}_i) - \sigma))}, \tag{2.2}$$

where σ and $\omega > 0$ are the threshold and scaling parameters, respectively. We set the scaling ω so that $T(\boldsymbol{s})$ will result in a sharp transition from 0 to 1 near σ. Using equation 2.1 we define the contrastive loss as

$$L_{con}(\theta) = \sum_i \ell(\boldsymbol{x}_i^*, \neg y; f_\theta), \tag{2.3}$$

where f_θ is the prediction module, $\ell(\boldsymbol{x}, y; f_\theta)$ is the loss function we wish to optimize, *e.g.* cross entropy, and $\neg$ is used to denote that the original class label has been flipped. In the binary case, $\neg y = 1 - y$, and in the multi-class case it can be interpreted accordingly, *e.g.*, using a one *vs.* others cross-entropy loss. In practice, we set $\ell(\boldsymbol{x}, y; f_\theta) = -\ell(\boldsymbol{x}, y; f_\theta)$. We will show in the experiments that this simple and intuitive causal masking strategy works well in practice (see Tables 2.2 and 2.4, and Figure 2.5). Alternatively, we also consider a *hard-masking* approach in which a minimal bounding box covering the thresholded saliency map is removed. See the Appendix for details.

Note that we are making the implicit assumption that the saliency map is uniquely determined by the prediction module f_θ. While optimizing equation 2.3 explicitly attempts to improve the fit of the prediction module f_θ, it also implicitly informs the causal saliency mapping. This is sensible because if a prediction is made using non-causal features, which implies the associated saliency map $\boldsymbol{s}_m(\boldsymbol{x})$ is also non-causal, then we should expect that after applying $\boldsymbol{s}_m(\boldsymbol{x})$ to $\boldsymbol{x}$ using equation 2.1, we can still expect to make the correct prediction, *i.e.*, the true label, for both positive (the original) and negative (the intervened) samples.

Saliency map regularization. Note that naively optimizing equation 2.3 can lead to degenerate solutions for which any saliency map that satisfies the causal sufficiency, *i.e.*, encompassing all causal features, is a valid causal saliency map. For example, a trivial solution where the saliency map covers the entire image may be considered causal. To protect against such degeneracy, we propose to regularize the L_1-norm of the saliency map to encourage succinct (sparse) representations, *i.e.*, $L_{reg} = \|\boldsymbol{s}_m\|_1$, for $m = 1, \ldots, M$.

Adversarial positive contrasts. Another concern with solely optimizing equation 2.3 is that models can easily overfit to the intervention, *i.e.*, instead of learning

to capture causal relevance, the model learns to predict interventional operations. For example, the model can learn to change its prediction when it detects that the input has been intervened, regardless of whether the image is missing causal features. So motivated, we introduce adversarial positive contrasts:

$$\boldsymbol{x}_i' = \boldsymbol{x}_i - T(\boldsymbol{s}_m(\boldsymbol{x}_j)) \odot \boldsymbol{x}_i, \quad i \neq j, \tag{2.4}$$

where we intervene with a *false* saliency map, *i.e.*, $\boldsymbol{s}_m(\boldsymbol{x}_j)$ is the saliency map from a different input $\boldsymbol{x}_j$, while still encouraging the model to make the correct prediction via

$$L_{ad}(\theta) = \sum_i \ell(\boldsymbol{x}_i', y; f_\theta)\,, \tag{2.5}$$

where $\boldsymbol{x}_i'$ is the adversarial positive contrast. The complete loss for the proposed model, $L = L_{cls} + L_{con} + L_{reg} + L_{ad}$, consists of the contrastive loss in equation 2.3, the regularization loss, L_{reg}, and the adversarial loss in equation 2.5.

2.3.2 Saliency Weight Backpropagation

PPI requires a module to generate saliency maps that inform decision-driving features in the (raw) pixel space. This module is expected to: i) generate high-quality saliency maps that faithfully reflect the model's focus, ii) efficient, as it will be used repeatedly in the PPI training framework. There is a tension between these two goals: advanced saliency maps methods are usually time-consuming [STK+17, STY17]. PPI finds a sweet point that better balances the trade-offs: the *Weight Back Propagation* (WBP). WBP is a novel computationally efficient scheme for saliency mapping, and it is applicable to arbitrary neural architectures. Heuristically, WBP evaluates individual contributions from each pixel to the final class-specific prediction. Empirical

examinations reveals that WBP results to be more causally-relevant relative to competing solutions based on human judgement (refer to Figure 2.3 visualization).

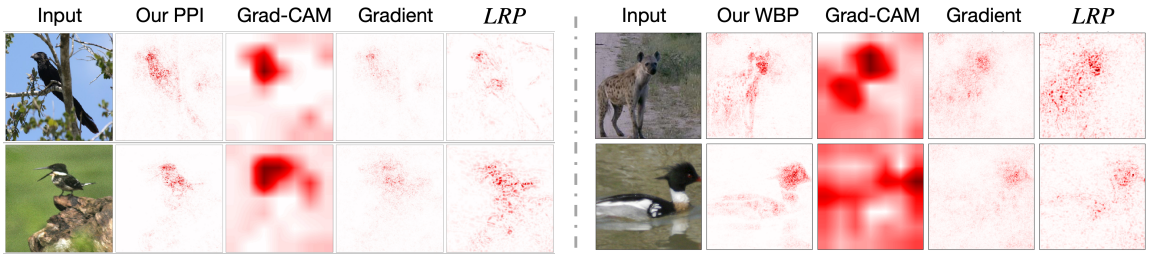


Figure 2.3: Visualization of the inferred saliency maps. Left: CUB dataset (PPI is based on WBP). Right: ImageNet dataset.

To simplify our presentation, we first consider a vector input and a linear mapping. Let $\boldsymbol{v}^l$ be the internal representation of the data at the l-th layer, with $l = 0$ being the input layer, *i.e.*, $\boldsymbol{v}^0 = \boldsymbol{v}$, and $l = L$ being the penultimate *logit* layer prior to the softmax transformation, *i.e.*, $\mathbb{P}(y|\boldsymbol{v}) = \text{softmax}(\boldsymbol{v}^L)$. To assign the relative importance to each hidden unit in the l-th layer, we notationally collapse all transformations after l into an operator denoted by $\tilde{\boldsymbol{W}}^l$, which we call the *saliency matrix*, satisfying,

$$\boldsymbol{v}^L = \tilde{\boldsymbol{W}}^l \boldsymbol{v}^l, \quad \forall l \in [0, \ldots, L], \tag{2.6}$$

where $\boldsymbol{v}^L$ is an M-dimensional vector corresponding to the M distinct classes in y. Though presented in a matrix form in a slight abuse of notation, *i.e.*, the instantiation of the operator $\tilde{\boldsymbol{W}}^l$ effectively depends on the input $\boldsymbol{v}$, thus all nonlinearities have been effectively absorbed into it. We posit that for an object associated with a given label $y = m$, its causal features are subsumed in the interactions between the m-th row of $\tilde{\boldsymbol{W}}^0$ and input $\boldsymbol{v}$, *i.e.*,

$$[\boldsymbol{s}_m(\boldsymbol{v})]_k = [\tilde{\boldsymbol{W}}^0]_{mk} [\boldsymbol{v}]_k, \tag{2.7}$$

where $[\boldsymbol{s}_m(\boldsymbol{v})]_k$ denotes the k-th element of the saliency map $\boldsymbol{s}_m(\boldsymbol{v})$ and $[\tilde{\boldsymbol{W}}^0]_{mk}$ is a single element of $\tilde{\boldsymbol{W}}^0$. A key observation for computation of $\tilde{\boldsymbol{W}}^l$ is that it can be done recursively. Specifically, let $g_l(\boldsymbol{v}^l)$ be the transformation at the l-th layer, *e.g.*, an affine transformation, convolution, activation, normalization, *etc.*, then it holds that

$$\tilde{\boldsymbol{W}}^{l+1}\boldsymbol{v}^{l+1} = \tilde{\boldsymbol{W}}^{l+1}g_l(\boldsymbol{v}^l) = \tilde{\boldsymbol{W}}^l\boldsymbol{v}^l. \tag{2.8}$$

This allows for recursive computation of $\tilde{\boldsymbol{W}}^l$ via

$$\tilde{\boldsymbol{W}}^l = G(\tilde{\boldsymbol{W}}^{l+1}, g_l), \quad \tilde{\boldsymbol{W}}^L = 1, \tag{2.9}$$

where $G(\cdot)$ is the update rule. We list the update rules for common transformations in deep networks in Table 2.1, with corresponding derivations detailed below.

Table 2.1: WBP update rules for common transformations.

Transformation	$G(\cdot)$
Activation Layer	$\tilde{\boldsymbol{W}}^l = h \circ \tilde{\boldsymbol{W}}^{l+1}$
FC Layer	$\tilde{\boldsymbol{W}}^l = \tilde{\boldsymbol{W}}^{l+1}\boldsymbol{W}^l$
Convolutional Layer	$\tilde{\boldsymbol{W}}^l = \tilde{\boldsymbol{W}}^{l+1} \otimes [\boldsymbol{W}^l]^{T_{0,1}}_{flip_{2,3}}$
BN Layer	$\tilde{\boldsymbol{W}}^l = \frac{\tilde{\boldsymbol{W}}^{l+1}}{\sigma}\gamma$
Pooling Layer	Relocate/Distribute $\tilde{\boldsymbol{W}}^{l+1}$

Fully-connected (FC) layer. The FC transformation is the most basic operation in deep neural networks. Below we omit the bias term as it does not directly interact with the input. Assuming $g_l(\boldsymbol{v}^l) = \boldsymbol{W}^l\boldsymbol{v}^l$, it is readily seen that

$$\tilde{\boldsymbol{W}}^{l+1}\boldsymbol{v}^{l+1} = \tilde{\boldsymbol{W}}^{l+1}g_l(\boldsymbol{v}^l) = (\tilde{\boldsymbol{W}}^{l+1}\boldsymbol{W}^l)\boldsymbol{v}^l, \tag{2.10}$$

so $\tilde{\boldsymbol{W}}^l = \tilde{\boldsymbol{W}}^{l+1}W^l$. Graphical illustration with standard affine mapping and ReLU activation can be found in the Appendix.

Nonlinear activation layer. Considering that an activation layer simply *rescales* the saliency weight matrices, *i.e.*, $\boldsymbol{v}^{l+1} = g_l(\boldsymbol{v}^l) = h^l \circ \boldsymbol{v}^l$, where $\circ$ is the composition operator, we obtain $\tilde{\boldsymbol{W}}^l = h \circ \tilde{\boldsymbol{W}}^{l+1}$. Using the ReLU activation as a concrete example, we have $h(\boldsymbol{v}^l) = \mathbb{1}\{\boldsymbol{v}^l \geq 0\}$.

Convolutional layer. The convolution is a generalized form of linear mapping. In practice, convolutions can be expressed as tensor products of the form $\tilde{\boldsymbol{W}}^l = \tilde{\boldsymbol{W}}^{l+1} \otimes [\boldsymbol{W}^l]^{T_{0,1}}_{flip_{2,3}}$, where $\boldsymbol{W}^l \in \mathbb{R}^{D_2 \times D_1 \times (2S+1) \times (2S+1)}$ is the convolution kernel, $T_{0,1}$ is the transpose in dimensions 0 and 1 and $flip_{2,3}$ is an exchange in dimensions 2 and 3. See the Appendix for details.

Pooling and normalization layer. Summarization and standardization are two other essential operations for the success of deep neural networks, achieved by pooling and batch normalization (BN) techniques, respectively. They too can be considered as special instantiations of linear operations. Here we summarize the two most popular operations in Table 2.1.

2.4 Experiments

To validate the utility of our approach, we consider both natural and medical image datasets, and compare it to existing state-of-the-art solutions. All the experiments are implemented in PyTorch. Due to space limitation, details of the experimental setup and additional analyses are deferred to the Appendix.

Datasets. We present our findings on five representative datasets: (*i*) `CIFAR-10` [KH+09]; (*ii*) `ImageNet (ILSVRC2012)` [RDS+15]; (*iii*) `CUB` [WBW+11], a natural image dataset with over $12k$ photos for classification of 200 bird species in the wild, heavily confounded by the background characteristics; (*iv*) `GA` [LSW+13], a new medical image dataset for the prediction of *geographic atrophy* (GA) using 3D *optical coher-*

ence tomography (OCT) image volumes, characterized by small sample size (275 subjects) and highly heterogeneous (collected from 4 different facilities); and (v) `LIDC-IDRI` [LAE+19], a public medical dataset of $1,085$ lung lesion CT images annotated by 4 radiologists. Detailed specifications are described in the Appendix.

Baselines. We compare model trained with and without PPI framework to show gains on classification, model interpretability, and cross-domain generalization. Meanwhile, we compare our proposed WBP to the following set of popular saliency mapping schemes: (i) Gradient: standard gradient-based salience mapping; (ii) GradCAM [SCD+17]: gradient-weighted class activation mapping; (iii) LRP [BBM+15]: layer-wise relevance propagation and its variants. We do no consider more advanced saliency mapping schemes, like perturbation based methods, because it is too time consuming to be used for training purposes.

Hyperparameters. The final loss of the proposed model is a weighted summation of four losses: $L = L_{cls} + w_1 L_{con} + w_2 L_{reg} + w_3 L_{ad}$. The weights are simply balanced to match the magnitude of L_{cls}, *i.e.*, $w_3 = 1$, $w_2 = 0.1$, and $w_1 = 1$ (CUB, Cifar-10, and GA) and $= 10$ (LIDC). See Appendix Sec B for details about the masking parameters σ and ω.

2.4.1 Natural Image Datasets

Classification Gains. In this experiment, we investigate how the different pairings of PPI and saliency mapping schemes (*i.e.*, GradCAM, LRP, WBP) affect performance. In Table 2.2, the first row represents VGG11 model trained with only classification loss, and the following rows represent VGG11 trained with PPI with different saliency mapping schemes. We see consistent performance gains in accuracy via incorporating PPI training on both CUB and CIFAR-10 datasets. The gains are mostly significant

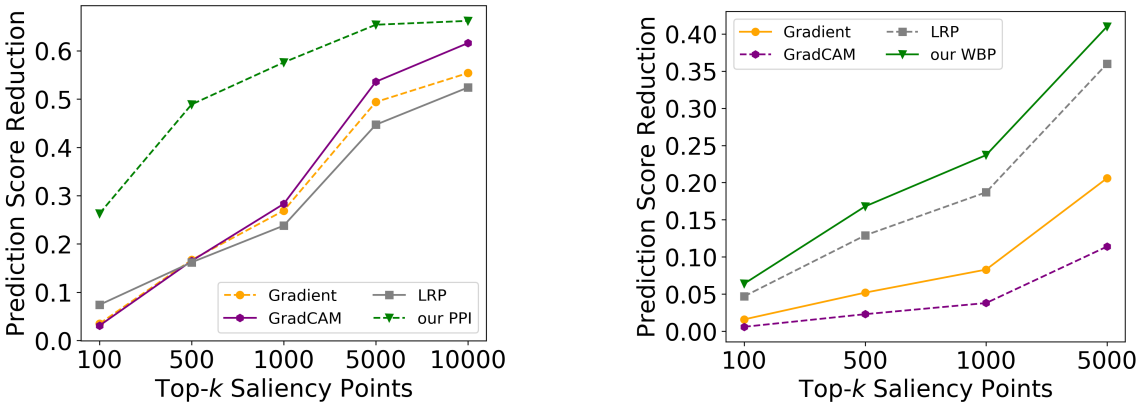
when using our WBP for saliency mapping (improving the accuracy from 0.662 to 0.696 on CUB, and from 0.881 to 0.901 on CIFAR-10).

Table 2.2: Performance improvements achieved by training with PPI on CUB, CIFAR-10, and GA dataset. We report means and standard deviations (SDs) from 5-fold cross-validation for GA prediction.

Models	**CUB (Acc)**	**Cifar-10 (Acc)**	**GA (AUC ± SD)**
Classification	0.662	0.881	0.877 ± 0.040
+PPI$_{Gradient}$	0.673	0.885	0.890 ± 0.035
+PPI$_{LRP}$	0.680	0.891	0.895 ± 0.037
+PPI$_{GradCAM}$	0.683	0.895	0.908 ± 0.036
+PPI$_{WBP}$	**0.696**	**0.901**	$\mathbf{0.925 \pm 0.023}$

Model Interpretability. In this task, we want to qualitatively and quantitatively compare the causal relevance of saliency maps generated by our proposed model and its competitors. In Figure 2.3(right), we show the saliency maps produced by different approaches for a VGG11 model trained on CUB. Visually, gradient-based solutions (Grad and GradCAM) tend to yield overly dispersed maps, indicating a lack of specificity. LRP gives more appealing saliency maps. However, these maps also heavily attend to the spurious background cues that presumably help with predictions. When trained with PPI (based on WBP), the saliency maps focus on the causal related pixels, i.e., special parts of birds body.

To quantitatively evaluate the causal relevance of competing saliency maps, we adopt the evaluation scheme proposed in [HEKK19], consisting of masking out the contributing saliency pixels and then calculating the reduction in prediction score. A larger reduction is considered better for accurately capturing the pixels that 'cause' the prediction. Results are summarized in Figure 2.4a, where we progressively remove the top-k saliency points, with $k = 100, 500, 1000, 5000, 10000$ ($10000 \approx 6.6\%$ of all pixels), from the CUB test input images. Our PPI consistently outperforms its



(a) CUB

(b) ImageNet

Figure 2.4: Quantitative evaluations of causal relevance of competing saliency maps (higher is better).

counterparts, with its lead being most substantial in the low-k regime. Notably, for large k, PPI removes nearly all predictive signal. This implies PPI specifically targets the causal features. Quantitative evaluation with additional metrics are provided in the Appendix.

To test the performance of WBP itself (without being trained with PPI), we compare WBP with different approaches for a VGG11 model trained on ImageNet from PyTorch model zoo. Figure 2.3(left) shows that saliency maps generated by WBP more concentrate on objects themselves. Also, thanks to the fine resolution of WBP, the model pays more attention to the patterns on the fur to identify the leopard (row 1). This is more visually consistent with human judgement. Figure 2.4b demonstrates WBP identifies more causal pixels on ImageNet validation images.

2.4.2 OCT-GA: Geographic Atrophy Classification

Next we show how the proposed PPI handles the challenges of small training data and heterogeneity in medical image datasets. In this experiment (with our new dataset, that we will make public), each OCT volume image consists of 100 scans of a 512×1000

sized image [BSEvLC+17]. We use a multi-view CNN model [SMKLM15] to process such 3D OCT inputs, and use it as our baseline solution (see the Appendix). We investigate how the different saliency mapping schemes (*i.e.*, Grad, GradCAM, LRP, WBP) work with PPI. For WBP, we also tested the bounding box variant, denoted as WBP (box) (see the Appendix). In Table 2.2, we see consistent performance gains in AUC score via incorporating PPI training (from 0.877 to 0.925, can be improve to 0.937 by PPI with WBP(box)), accompanied by the reductions in model variation evaluated by the standard deviations of AUC from the five-fold cross-validation. The gains are most significant when using our WBP for saliency mapping. The model trained by PPI with WBP is significantly different with other baseline models based on the Delong test for receiving operating characteristic comparisons [DDCP88]. We further compare the saliency maps generated by these different combinations. We see that without the additional supervision from PPI, competing solutions like Grad, GradCAM and LRP sometimes yield non-sensible saliency maps (attending to image corners). Overall, PPI encourages more concentrated and less noisy saliency maps. Also, different PPI-based saliency maps agree with each other to a larger extent. Our findings are also verified by experts (co-authors, who are ophthalmologists specializing in GA) confirming that the PPI-based saliency maps are clinically relevant by focusing on retinal layers likely to contain abnormalities or lesions. These results underscore the practical value of the proposed proactive interventions.

Cross-domain generalization. Common to medical image applications is that training samples are usually integrated from a number of healthcare facilities (*i.e.*, domains), and that predictions are sometimes to be made on subjects at other facilities. Despite big efforts to standardize the image collection protocols, with different imaging systems operated by technicians with varying skills, apparent domain shifts are likely to compromise the cross-domain performance of these models. We show this phenomenon

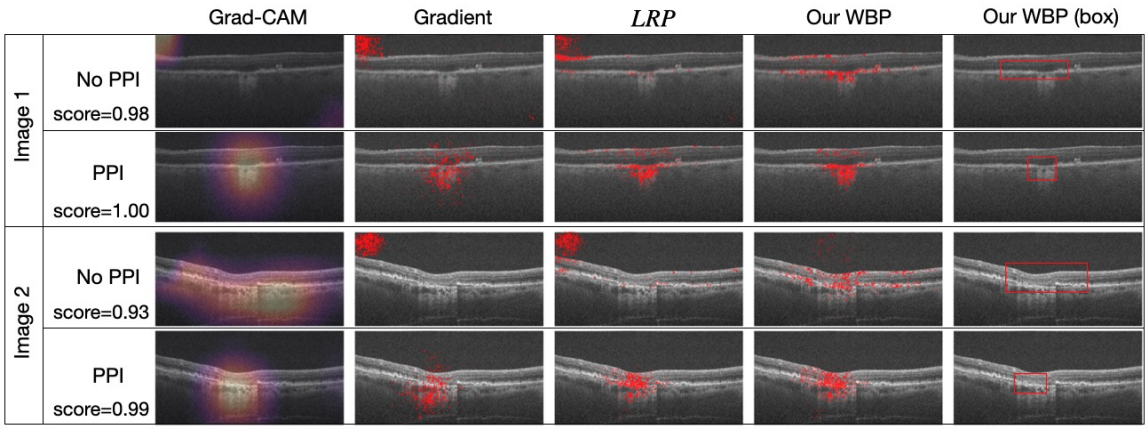


Figure 2.5: Saliency maps on GA dataset based on models trained with PPI and without PPI. Maps of models trained with PPI are more clinically relevant by focusing on retinal layers likely to contain abnormalities or lesions, and more concentrated.

on the `GA` dataset in Table 2.3, where source samples are collected from four different hospitals in different health systems (A, B, C and D, see the Appendix for details). Each cell contains the AUC of the model trained on site X (row) and tested on site Y (column), with same-site predictions made on hold-out samples. A significant performance drop is observed for cross-domain predictions (off-diagonals) compared to in-domain predictions (diagonals). With the application of PPI, the performance gaps between in-domain and cross-domain predictions are considerably reduced. The overall accuracy gains of PPI further justify the utility of causally-inspired modeling. Notably, site D manifests strong spurious correlation that help in-domain prediction but degrades out-of-site generalization, which is partly resolved by the proposed PPI.

2.4.3 LIDC-IDRI: Lung Lesions Classification

To further examine the practical advantages of the proposed PPI in real-world applications, we benchmark its utility on LIDC-IDRI; a public lung CT scan dataset [AIMB+11]. We followed the preprocessing steps outlined in [KRPM+18] to prepare the data, and adopted the experimental setup from [SD20] to predict lesions. We

Table 2.3: AUC results for GA prediction with or without PPI. Models are trained on one site and cross-validated on the other sites. Darker color indicates better performance.

With PPI	A	B	C	D	Mean	STD
A	1.000	0.906	0.877	0.865	**0.912**	**0.061**
B	0.851	0.975	0.863	0.910	**0.900**	0.056
C	0.954	0.875	0.904	0.931	**0.916**	**0.034**
D	0.824	0.846	0.853	0.904	**0.857**	**0.034**
No PPI	A	B	C	D	Mean	STD
A	1.000	0.854	0.832	0.827	0.878	0.082
B	0.810	0.874	0.850	0.906	0.860	**0.040**
C	0.860	0.779	0.873	0.862	0.843	0.043
D	0.748	0.792	0.836	0.961	0.834	0.092

use Inception_v3 [SVI+16] as our base model for both standard classification and PPI-enhanced training with various saliency mapping schemes. See the Appendix for details.

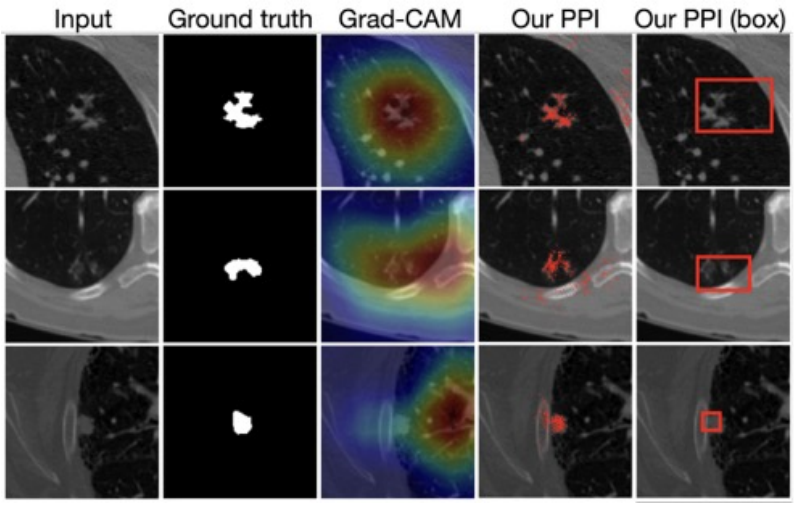


Figure 2.6: Saliency maps on LIDC-IDR. Saliency maps of PPI+WBP are mostly consistent with the ground truths.

Lesion classification. We first compare PPI to other specialized SOTA network architectures. Table 2.4 summarizes AUC scores of Tensor Net-X [EHL19], DenseNet [HLVDMW17], LoTeNet [SD20], Inception_v3 [SVI+16], as well as our Inception_v3 trained with and without PPI_{WBP}. The proposed $\text{PPI}_{WBP(box)}$ leads the

Table 2.4: LIDC-IDRI classification AUC results.

Models	**AUC**
Tensor Net-X [EHL19]	0.823
DenseNet [HLVDMW17]	0.829
LoTeNet [SD20]	0.874
Inception_v3 [SVI+16]	0.921
+PPI$_{GradCAM}$	0.933
+PPI$_{Gradient}$	0.930
+PPI$_{LRP}$	0.931
+PPI$_{WBP}$	0.935
+PPI$_{WBP(box)}$	**0.941**

performance chart by a considerable margin, improving Inception_v3 from 0.92 to 0.94.

Weakly-supervised image segmentation. In Figure 2.6, we compare saliency maps generated by GradCAM, WBP, WBP (box) to the ground truth lesion masks from expert annotations. Note that we have only supplied patch-label labels during training, not the pixel-level expert segmentation masks, which constitute a challenging task of weakly-supervised image segmentation. In line with the observations from the GA experiment, our PPI-training enhanced WBP saliency maps are mostly consistent with the expert segmentations. Together with Table 2.4, Figure 2.6 confirms that the proposed PPI+WBP improves both the classification performance and model interpretability.

2.5 Conclusions

We have presented *Proactive Pseudo-Intervention* (PPI), a novel interpretable computer vision framework that organically integrates saliency mapping, causal reasoning,

synthetic intervention and contrastive learning. PPI couples saliency mapping with contrastive training by creating artificially intervened negative samples absent of causal features. To communicate model insights and facilitate pre-informed reasoning, we derived an architecture-agnostic saliency mapping scheme called *Weight Back Propagation* (WBP), which faithfully captures the causally-relevant pixels/features for model prediction. Visual inspections of the saliency maps show that WBP, is more robust to spurious features compared to competing approaches. Empirical results on natural and medical datasets verify the combination of PPI and WBP consistently delivers performance boosts across a wide range of tasks relative to competing solutions, and the gains are most significant where the application is complicated by small sample size, data heterogeneity, or confounded with spurious correlations.

2.6 Appendix

2.6.1 Derivation of Convolutional Weight Backpropagation

Let's denote the input variable as $\boldsymbol{I} \in \mathbb{R}^{H \times W}$, the convolutional filter weight as $\boldsymbol{W} \in \mathbb{R}^{(2S+1)\times(2S+1)}$, the output variable as $\boldsymbol{O} \in \mathbb{R}^{H \times W}$, and the weight backpropagate to O as $\hat{\boldsymbol{W}} \in \mathbb{R}^{H \times W}$. We omit the bias here because it does not directly interact with the input variables. We denote $\otimes$ as the convolutional operator. We have

$$
\begin{aligned}
\boldsymbol{O} &= \boldsymbol{I} \otimes \boldsymbol{W} \\
\boldsymbol{O}_{i,j} &= \sum_{i'=-S}^{S} \sum_{j'=-S}^{S} \boldsymbol{I}_{i+i',j+j'} \boldsymbol{W}_{i'+S,j'+S} \\
\sum_i \sum_j \boldsymbol{O}_{i,j} \tilde{\boldsymbol{W}}_{i,j} &= \sum_i \sum_j \sum_{i'=-S}^{S} \sum_{j'=-S}^{S} \boldsymbol{I}_{i+i',j+j'} \boldsymbol{W}_{i'+S,j'+S} \tilde{\boldsymbol{W}}_{i,j} \\
\sum_i \sum_j \boldsymbol{O}_{i,j} \tilde{\boldsymbol{W}}_{i,j} &= \sum_i \sum_j \boldsymbol{I}_{i,j} \sum_{i'=-S}^{S} \sum_{j'=-S}^{S} \tilde{\boldsymbol{W}}_{i+i',j+j'} \boldsymbol{W}_{-i'+S,-j'+S} \\
\sum_i \sum_j \boldsymbol{O}_{i,j} \tilde{\boldsymbol{W}}_{i,j} &= \sum_i \sum_j \boldsymbol{I}_{i,j} (\tilde{\boldsymbol{W}} \otimes [\boldsymbol{W}]_{flip_{i,j}})_{i,j}
\end{aligned}
$$

Hence the weight backpropagate through a convolutional layer is $\tilde{\boldsymbol{W}}^l = \tilde{\boldsymbol{W}}^{l+1} \otimes [\boldsymbol{W}^l]_{flip}$. For the 3D cases, $\boldsymbol{I}^l \in \mathbb{R}^{D_1 \times H \times W}$,the weight back propagates to $\boldsymbol{O}^l$ is $\tilde{\boldsymbol{W}}^{l+1} \in \mathbb{R}^{D_2 \times H \times W}$ and the convolutional weight is $\boldsymbol{W}^l \in \mathbb{R}^{D_2 \times D_1 \times (2S+1) \times (2S+1)}$. To match the depth of $\tilde{\boldsymbol{W}}^{l+1}$, the W^l is transposed in the first two dimensions. So $\tilde{\boldsymbol{W}}^l = \tilde{\boldsymbol{W}}^{l+1} \otimes [\boldsymbol{W}^l]^{T_{0,1}}_{flip_{2,3}}$. If the convolutional layer is downsizing the input variable (*i.e.*, strides), the $\tilde{\boldsymbol{W}}^{l+1}_{ijk}$ is padded with zeros around the weights (left,right,up, and down) to for the input elements that the convolutional filter strides over. The number of padding zeros is equal to the number of strides minus 1.

2.6.2 Details on Causal Masking

In this work, we consider three types of causal masking: (*i*) the point-wise soft causal masking defined by Equation (2) in the main text, (*ii*) hard masking, and (*iii*) box masking. For the hard masking, for each image, we keep points with WBP weight larger than k times of the standard deviation of WBP weights of the whole image. We test k from 1 to 7 and achieve similar results. As $k = 7$ performs slightly better, we set k as 7 for all experiments. For the box masking, we use the center of mass for these kept points as the center to draw a box. The height and width of this box is defined as $center_{h/w} \pm 1.2std_{h/w}$. In this way at least 90% of filtered points are contained in the box. For the soft masking, we set ω to 100 and σ to 0.25. We have also experimented with image-adaptive thresholds instead of a fixed σ for all inputs, *i.e.*, set the threshold as mean value plus k times of the standard deviation of WBP weights of the whole image. We repeat the experiments a few times and the results are consistent. The experiment comparison of these masking methods mention above is conducted on LIDC dataset.

Table 2.5: AUC on LIDC from different causal masking methods

Models	**AUC**
WBP-soft (fixed σ)	0.931
WBP-soft (adaptive σ)	0.941
WBP-hard (point)	0.935
WBP-hard (box)	**0.941**

2.6.3 Related Work

In this work, we propose a contrastive causal representation learning strategy, i.e., Proactive Pseudo-Intervention (PPI), that leverages proactive interventions to identify

causally-relevant image features. This approach is complemented with a novel causal salience map visualization module, i.e., Weight Back Propagation (WBP), that identifies important pixels in the raw input image, which greatly facilitates interpretability of predictions.

Prior related works will be discussed in this section. Compared with alternative post-hot saliency mapping methods, WBP outperforms these methods as both a standalone causal saliency map and a trainable model for model interpretation. Compared with other trainable interpretation models, the proposed PPI+WBP improves both model performance and model interpretations.

Post-hoc Saliency Maps

We compare WBP with other post-hoc saliency mapping methods to show why WBP is able to target the causal features, and generate more succinct and reliable saliency maps.

Perturbation Based Methods These methods make perturbations to individual inputs or neurons and monitor the impact on output neurons in the network. [ZF14] occludes different segments of an input image and visualized the change in the activations of subsequent layers. Several methods follow a similar idea, but use other importance measures or occlusion strategies [PDS18, RSG16, SOO19]. More complicated works aim to generate an explanation by optimizing for a perturbed version of the image [FV17, FPV19, DG17, DLSH18]. [WKG+19] proposes a new adversarial defense technique which filters gradients during optimization to achieve fine-grained explanation. However, such perturbation based methods are computationally intensive and involve sophisticated model designs, which make it extremely hard to be integrated with other advance learning strategies.

Backpropagation Based Methods Backpropagation based methods (BBM)

propagate an importance signal from an output neuron backwards through the layers to the input. These methods are usually fast to compute and produce fine-grained importance/relevancy maps. WBP is one of such method.

The pioneer methods in this category backpropagate a gradient to the image, and branches of studies extend this work by manipulating the gradient. These methods are discussed and compared in [MV16, EBCV09]. However, these maps are generally less concentrated [DG17, FV17] and less interpretable. Other BBMs such as Layer-wise Relevance Propagation [BBM+15], DeepLift [SGK17] employ top-down relevancy propagation rules. DeepLift is sensitive to the reference inputs, which needs more human efforts and background knowledge to produce appealing saliency maps. The nature of depending on reference inputs limits its ability on model diagnosis and couple with learning strategies to continuously improving models' performance. LRP decomposes the relevance, R, from a neuron, k, in the upper layer to every connected neurons, j, in the lower layer. The decomposition is distributed through gradients under the suggested implementation [MBL+19]. Our experiments on GA and CUB datasets show that vanilla LRP performs similar to gradient based methods, which is also demonstrated in [Mon19]. The variants of LRP use complex rules to prioritize positive or large relevance, making the saliency map visually appealing to human. However, our experiments demonstrate the unfaithfulness of LRP and its variants as they highlight spuriously correlated features (boarderlines and backgrounds). By contrast, our WBP backpropagates the the weights of through layers to compute the contributions of each input pixel, which is truly faithful to the model, and WBP tends to highlight the target objects themselves rather than the background. At the same time, the simplicity and efficiency makes WBP easily work with other advanced learning strategies for both model diagnosis and improvements during training.

Activation Based Methods Methods under this category (such as CAM, Grad-

Table 2.6: A list of commonly used LRP rules.[MBL+19]

Rules	Formula
LRP	$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$
LRP_ϵ	$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k$
LRP_γ	$R_j = \sum_k \frac{a_j (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j (w_{jk} + \gamma w_{jk}^+)} R_k$
$LRP_{\alpha\beta}$	$R_j = \sum_k (\alpha \frac{(a_j w_{jk})^+}{\sum_{0,j} (a_j w_{jk})^+} - \beta \frac{(a_j w_{jk})^-}{\sum_{0,j} (a_j w_{jk})^-}) R_k$
LRP_{flat}	$R_j = \sum_k \frac{1}{\sum_j 1} R_k$
LRP_{w^2}	$R_j = \sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_j$
LRP_{Z^β}	$R_j = \sum_j \frac{x_i w_{ij} - l_i w_{ij}^+ - h_i w^- ij}{\sum_i x_i w_{ij} - l_i w_{ij}^+ - h_i w^- ij} R_j$

CAM, guided Grad-CAM, Grad-CAM++) use a linear combination of class activation maps from convolutional layers to derive a saliency map. The main difference between them is how to the linear combination weights are computed. The generation of saliency maps is easy and these methods can be coupled with advanced training strategies to improve training [LWP+18]. However, they fail at visualizing fine-grained evidence, which is particularly important in explaining medical classification models. Additionally, it is not guaranteed that the resulting explanations are faithful and reflect the decision making process of the model [DLSH18, SCD+17, WKG+19]. Grad-CAM++ [CSHB18] proposes to introduce higher-order derivatives to capture pixel-level importance, while its high computational cost in calculating the second- and third-order derivatives makes it impractical for training purposes.

Interpretable Models

Unlike the *post hoc* saliency map generation described above, an alternative approach is to train a separate module to explicitly produce model explanations [FHYF19, GWE+19, CCGD18, FV17, SGK17]. Such *post hoc* causal explanations can be generated with black-box classifiers based on a learned low-dimensional rep-

resentation of the data [OCC+20] . Related to our work is adversarial-based visual explanation method is developed in [WKG+19], highlighting the key features in the input image for a specific prediction. Contrastive explanations are produced in [DCL+18] to justify the predictions from a deep neural network. Also in [GWE+19] the authors generate counterfactual visual explanations that highlight what and how regions of an image would need to change in order for the model to predict a *distractor* class c' instead of the predicted class c. The main differences to our construction are two fold: (i) they rely on a separate module to be trained, and (ii) these approaches only produce explanations, but such explanations are not exploited to provide feedback for model improvement.

Striking the goal of both good explanation and good performance is more challenging. One promising direction is to inject model-dependent perturbations to the input images as strategic augmentations [FV17, DG17, CCGD18]. In such examples, parts of the image have been masked and replaced with various references such as mean pixel values, blurred image regions, random noise, outputs of generative models, *etc.* However, these pixel-level perturbations are very costly and difficult to craft. [WWK+19] propose new learning objectives for attention separability and cross-layer consistency, which result in improved attention discriminability and reduced visual confusion. However, it generates heat-map style attention maps, which fail in fine-resolution model explanations which is important in medical related tasks. In [FHYF19] an additional attention branch is learned to generate attention map, and then applies the attention map to the original image or feature map; they achieve compelling attention maps on natural images. However, as the attention maps are not derived directly from the classification model, there is no guarantee for their faithfulness. Further, having an additional attention network results in increased network size, which raises concerns for the risk of over-fitting, particularly on datasets with a limited sample

size.

2.6.4 CUB Experiment Details

CUB dataset descriptions and experiemnt settings

CUB has 11,788 images of 200 bird spices. To train a VGG11 network, we use 8,190 training images and validate the model on 2,311 validation images, with the accuracy are reported on 1227 testing images. The network is trained for 100 epochs with a learning rate decay of 0.1 every 30 epochs. The batch size is 32. The optimizer is a SGD with initial learning rate at 0.01.

Classification performance improvement with PPI

We compare classification performances among model trained with different objections. The baseline is VGG11 classification without PPI. Three different saliency mapping methods are tested within our PPI framework: LRP, Grad-CAM, and WBP. Top 1000 points in all saliency maps are used to generate the soft mask so the comparison is fair. During training, since only a small portion of points are used to generate the mask, the contribution from L_{con} is about 100 times smaller than other losses. To fix this imbalance, the L_{con} is weighted 100× more after the first 20 epochs. The results are shown in Table 2.7.

Table 2.7: Accuracy on CUB

Models	**Accuracy**
VGG 11	0.662
+PPI$_{LRP}$	0.680
+PPI$_{Grad-CAM}$	0.683
+PPI$_{WBP}$	**0.696**

2.6.5 Geographic Atrophy (GA) Experiment Details

GA dataset descriptions

Our GA dataset is derived from the A2A SD-OCT Study (http://ClinicalTrials.gov identifier NCT00734487), which was an ancillary observational prospective study of a subset of eyes from the AREDS2 conducted at four sites (National Eye Institute, Duke Eye Center, Emory Eye Center, and Devers Eye Institute) [LSW+13]. In this experiment (with our new dataset, that we will make public), each OCT volume image consists of 100 scans, each of which being a 512×1000 pixel image [BSEvLC+17]. $1,085$ OCT images are collected from 275 subjects during 5 years. An example of 3D OCT images is shown in Figure. 2.7.

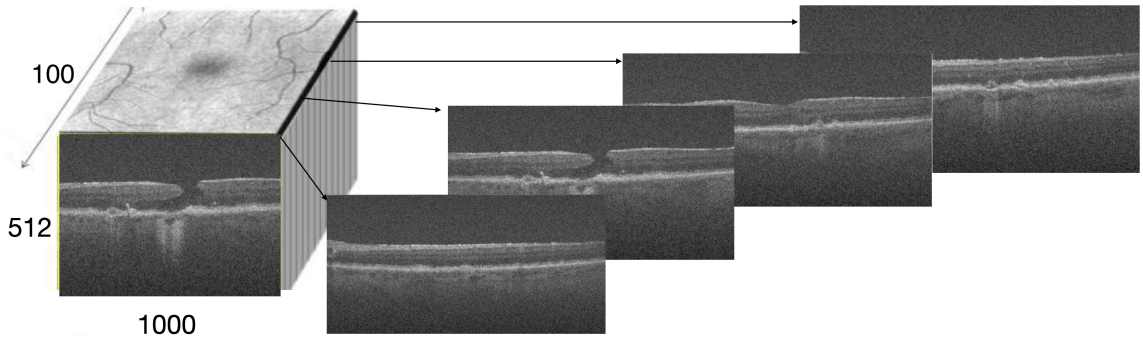


Figure 2.7: Illustration of a 3D OCT image example.

Image differences between 4 sites

Images in GA dataset are collected from 4 different sites, hereafter denoted as A, B, C, and D respectively. There are 315 images (101 positive samples) from site A, 334 images (73 positive samples) from site B, 260 images (131 positive samples) from site A, and 176 images (59 positive samples) from site D. We show typical example images from 4 sites separately in Figure. 2.8. As the dataset is collected during 7 years, some

images in site D are of smaller image size as they are sampled with different type of machine. We paddle these images by repeating the left and right areas, as show in the right bottom example.

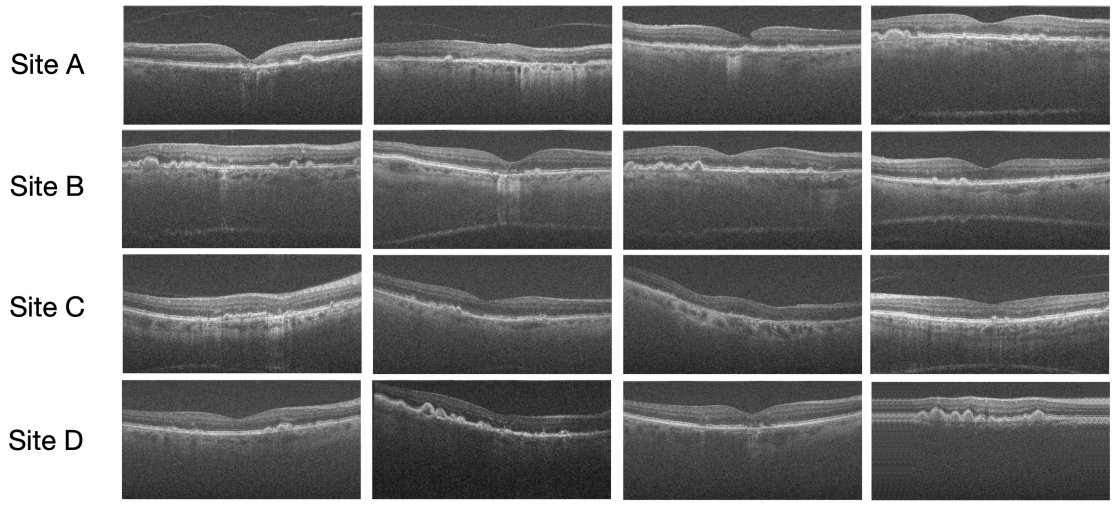


Figure 2.8: OCT slice examples from 4 site.

Multi-view CNN Variation

We use a variant of the multi-view CNN model [SMKLM15] to process the 3D OCT inputs, and use it as our baseline solution. The architecture of this model is outlined in Figure 2.9. For each slice, the model feed it into a CNN network, and get the feature f_i of slice i ($f_i = CNN(x_i)$), followed by a fully connected layer and a Sigmoid activation to get a probability score $p_i = sigmoid(FC_1(f_i))$. We observe that slices in different slices contributes differently to the identification of GA, which motivates us to implement a location-aware view pooling, illustrated in the right part of Fig. 2.9. Each slice is assigned to a position id, ranging from 1 to 100. The model first uses an embedding layer to embed the position id to a six dimension position feature vector e_i. Then, we combine the feature vector f_i extracted from the slice image with the corresponding e_i together. The combined feature vector is fed into a fully connected

layer to get the logit score a_i.

$$a_i = FC_2([f_i, e_i]) \tag{2.11}$$

To reduce computational burden during training, we randomly sample 10 out of the 100 slices (with an abuse of notation, denoted by $a_1, a_2, \ldots, a_{10}$) and send them into a Softmax function to get the attention weights for the 10 sampled slices, using the following equation

$$w_i = \frac{exp((ReLU(a_i) + \delta)/\tau)}{\sum_{k=1}^{10} exp((ReLU(a_i) + \delta)/\tau)} \tag{2.12}$$

Here δ is a trainable bias term parameter, initialized to a high value to stabilize the training, and gradually attenuated to a small number during training. τ is the temperature parameter, which is set to a small value to sharpen the attention weight, which helps us to find out the most important slices for GA diagnosis. The get final predicted probability of GA (GA score) for an image x at inference time, we compute the weighted summation of the probabilities w_i of all 100 slices $GA = \sum_i w_i p_i$.

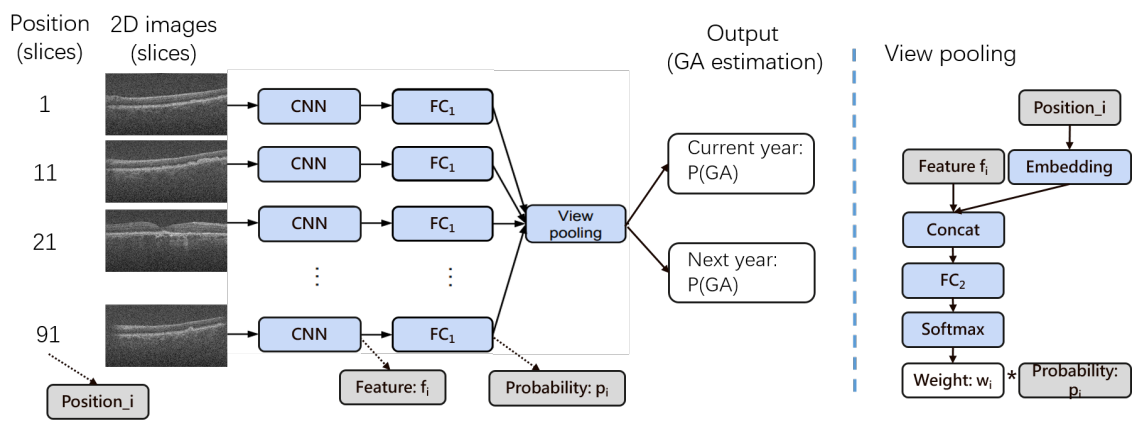


Figure 2.9: Illustration of multi-view CNN based 3D OCT image classification model.

Experiment settings

The CNN network is an Inception_v3, which is pre-trianed on ImageNet. For training all models, we use the Adam optimizer with a learning rate of 5×10^{-5} with a learning rate decay of 0.5 every 10 epochs for the pre-trained CNN network, and the Adam optimizer with a learning rate of 5×10^{-3} with a learning rate decay of 0.2 every 10 epochs for the other layers in the model. We use a batch size of 2 because of the large size of 3D OCT images and our multi-view CNN model (will be illustrated in 2.6.5). Random horizontal flips, and Gaussian noise are used for data augmentations during training.

2.6.6 LIDC Experiment Details

LIDC dataset description and experiment settings

We also test the proposed method on a public medical CT scan dataset LIDC-IDRI [AIMB+11]. We follow the settings in [KRPM+18, SD20] that crops the original images into 128×128 patches centered on a lesion for which at least one radiologist has annotated. In our experiment, we focus on the classification task of predicting the presence of a lesions, which is consistent with the setup of [SD20]. There are four radiologists annotates each patch with both lesion label and lesion mask. A patch in the dataset is labeled as positive if more than two (*i.e.* ≥ 3) radiologists have annotated presence of a lesion, otherwise negative. The ground-truth mask is the pixel-level union set of the four masks. We use Inception-v3 [SVI+16] as our base model for both standard classification and PPI-enhanced training with various saliency mapping schemes. To match the receptive field of an Inception-v3 model, we resize the input patches to 299×299. For training all models, we use the Adam

optimizer with a learning rate of 10^{-4} with a learning rate decay of 0.3 every 10 epochs after epoch 50, and a batch size of 64. Random horizontal flips, vertical flips and rotations within 20 degrees are used for data augmentations during training.

2.6.7 Additional Saliency Map Comparisons

Extra saliency map comparisons on CUB, GA, and LIDC are shown in Figure. 2.10, 2.11, and 2.12.

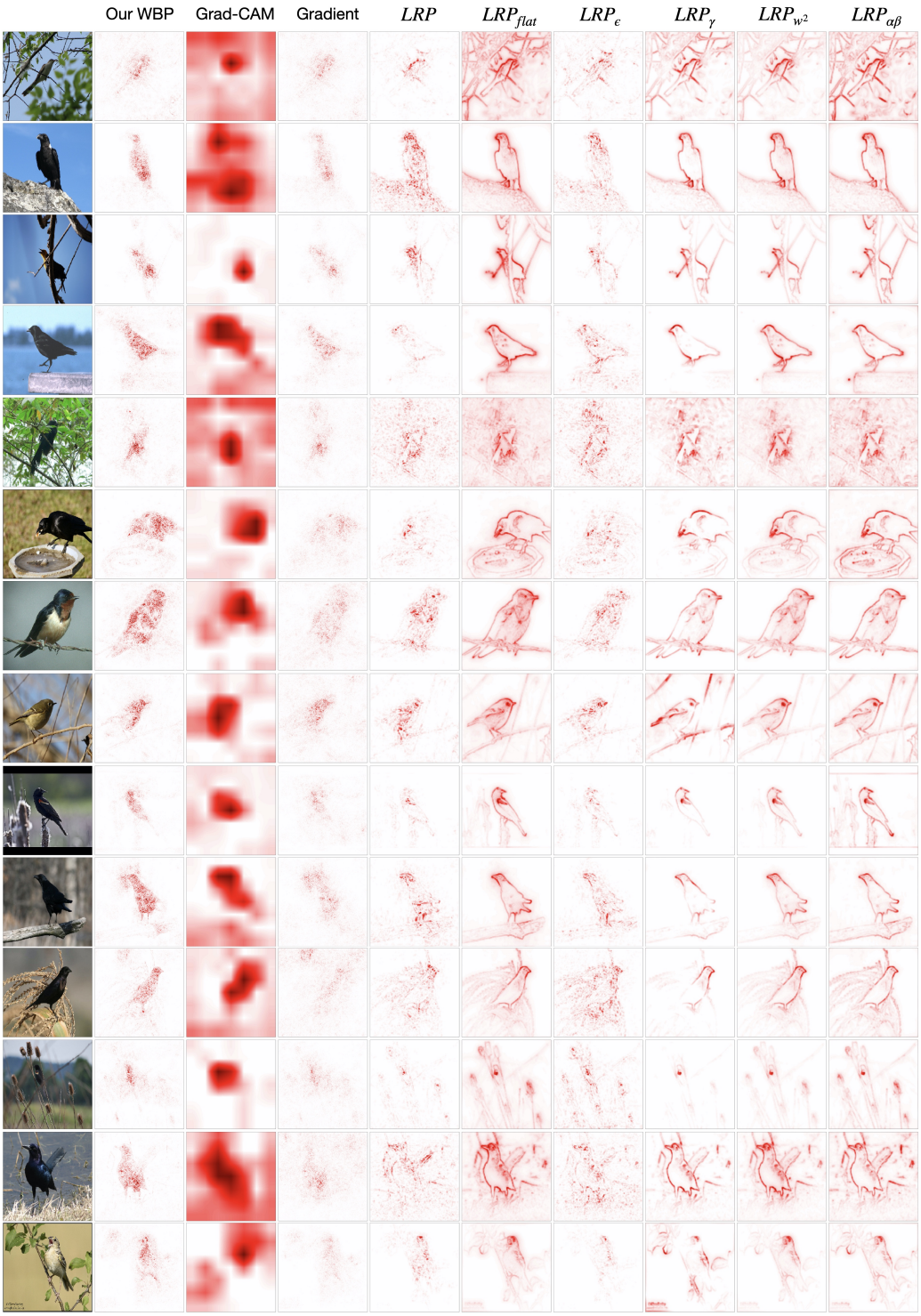


Figure 2.10: More comparisons of saliency maps on CUB dataset.

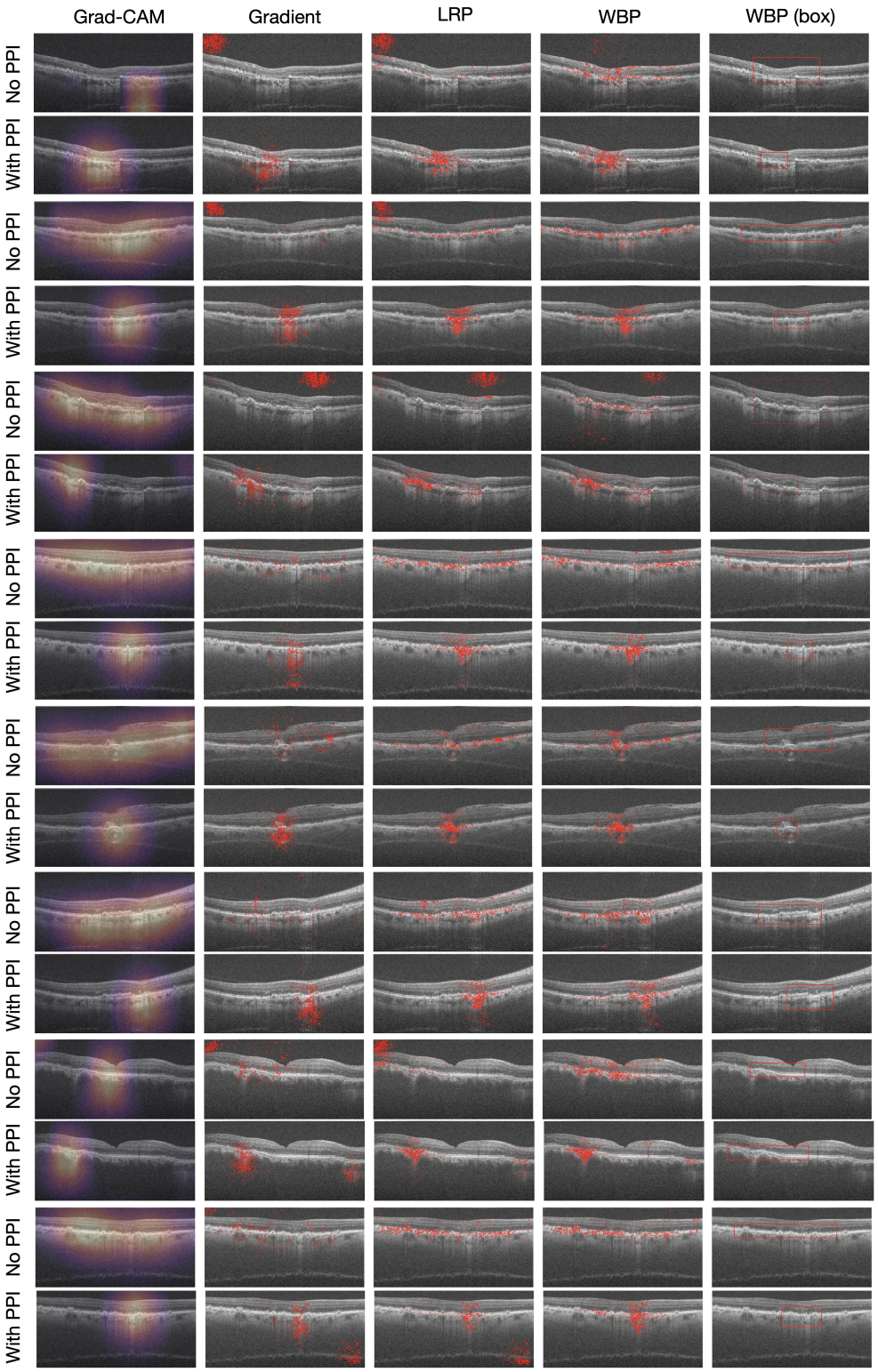


Figure 2.11: Examples of saliency maps on GA dataset based on model trained with and without PPI.

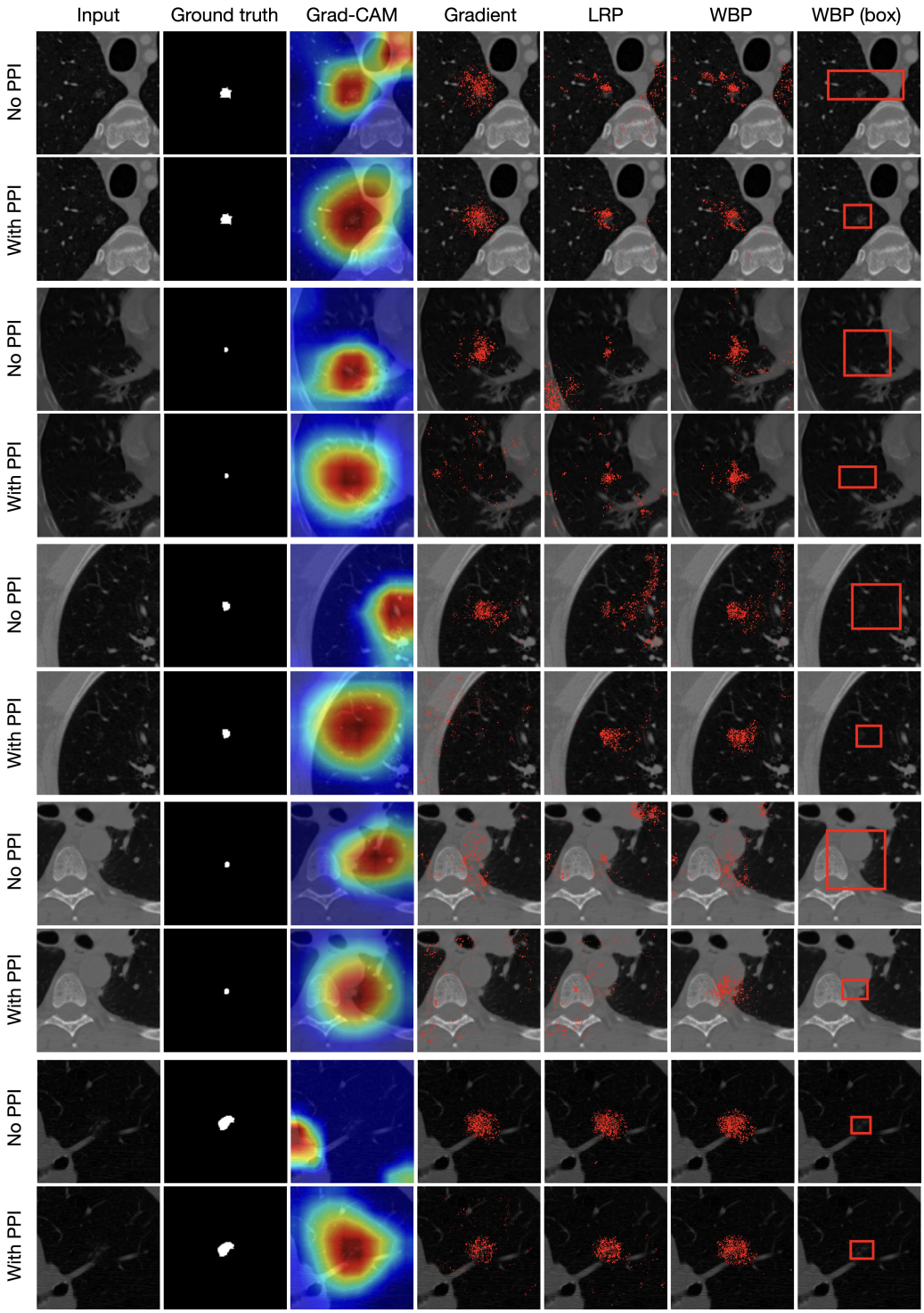


Figure 2.12: Examples of saliency maps on LIDC dataset based on model trained with and without PPI.

Chapter 3

Object Detection as a Positive-Unlabeled Problem

3.1 Introduction

The performance of supervised deep learning models is often highly dependent on the quality of the labels they are trained on [ZBB+17, VAC+17, JZL+18]. Recent work by [TSdC+19] has implied the existence of "support vectors" in datasets: hard to classify examples that have an especially significant influence on a classifier's decision boundary. As such, ensuring that these difficult examples have the correct label would appear to be important to the final classifier. However, collecting complete labels for object detection [GDDM14b, Gir15, LAE+16, RHGS15, RDGF16, DLHS16, RF17] can be challenging, much more so than for classification tasks. Unlike the latter, with a single label per image, the number of objects in an image is often variable, and objects can come in a large variety of shapes, sizes, poses, and settings, even within the same class. Furthermore, object detection scenes are often crowded, resulting in object instances that may be occluded. Given the requirement for tight bounding boxes and the sheer number of instances to label, constituting annotations can be very time-consuming. For example, just labeling instances, without localization, required ∼30K worker hours for the 328K images of MS COCO [LMB+14], and the airport checkpoint X-ray dataset used in [LHG+18, SSLC20], which required assembling bags, scanning, and hand labeling, took over 250 person hours for 4000 scans over the span of several months. For medical datasets [MAD+12, YWLS17, LGH+17, WPL+17], this becomes even more problematic, as highly trained (and expensive) radiologist

Figure 3.1: Because of the inherent difficulty of instance labeling, the ground truth of object detection datasets can be incomplete. Example images and their ground truth labels shown for (left to right) PASCAL VOC [EVW+10] (missing people, bottles), MS COCO [LMB+14] (missing people), Visual Genome [KZG+17] (missing people, tree, clothing, etc.), and DeepLesion [YWLS17] (ground truth is the dotted line; two boxes on the left indicate two unlabeled nodules).

experts or invasive biopsies are needed to determine ground truth.

As a result of its time-consuming nature, dataset annotations are often crowd-sourced when possible, either with specialized domain experts or Amazon Mechanical Turk. In order to ensure consistency, dataset designers establish labeling guidelines or have multiple workers label the same image [EVW+10, LMB+14]. Regardless, tough judgment calls, inter- and even intra-worker variability, and human error can still result in overall inconsistency in labeling, or missing instances entirely. This becomes especially exacerbated when establishing a larger dataset, like OpenImages [KRA+18], which while extremely large, is incompletely labeled.

Despite this, object detection algorithms often use the standard cross-entropy loss for object classification. Implicit to this loss function is the assumption that any region without a bounding box does not contain an object; in other words, classification is posed as a positive-negative (PN) learning problem. While reasonable for a completely labeled dataset, despite best efforts, this is often not the case in practice due to the previously outlined difficulties of instance annotations. As shown in Figure 3.1 for a wide array of common datasets, the lack of instance labels does not always mean the absence of true objects.

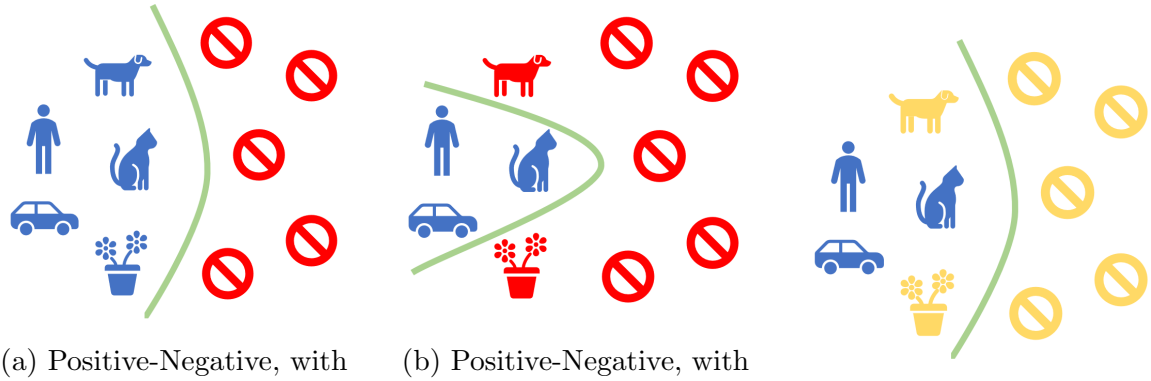
While the result of this characterization constitutes a noisy label setting, it is not noisy in the same respect as is commonly considered for classification problems [ZBB+17, VAC+17, JZL+18]. The presence of a positive label in object detection datasets are generally correct with high probability; it is the *lack* of a label that should not be interpreted with confidence as a negative (or background) region. Thus, given these characteristics common to object detection data, we propose recasting object detection as a positive-unlabeled (PU) learning problem [Den98, DCDGL99a, LDG00, EN08, KNDS17]. With such a perspective, existing labels still implies a positive sample, but the lack of one no longer enforces that the region must be negative. This can mitigate the confusing learning signal that often occurs when training on object detection datasets.

In this work, we explore how the characteristics of object detection annotation lend themselves to a PU learning problem and demonstrate the efficacy of adapting detection model training objectives accordingly. We perform a series of experiments to demonstrate the effectiveness of the PU objective on two popular, well-labeled object detection datasets (PASCAL VOC [EVW+10] and MS COCO [LMB+14]) across a range of label missingness, as well as two datasets with real incomplete labels (Visual Genome [KZG+17], DeepLesion [YWLS17]).

3.2 Methods

3.2.1 Faster R-CNN

In principle, the observed problem is characteristic of the data and is thus general to any object detection framework. However, in this work, we primarily focus on Faster R-CNN [RHGS15], a popular 2-stage method for which we provide a quick overview



(a) Positive-Negative, with full labels

(b) Positive-Negative, with missing labels

(c) Positive-Unlabeled

Figure 3.2: A classifier (green) learns to separate proposals by "objectness". Models trained with a standard cross-entropy loss implicitly assume positive-negative (PN) learning: regions with bounding boxes are considered positive (blue), and any other proposed boxes are treated as negative (red). This is reasonable when labels are complete (a), but in reality, object detection datasets are inherently missing annotations; this forces the classifier to exclude unlabeled objects from the positive class (b). We propose a positive-unlabeled (PU) approach (c), which considers non-labeled regions as unlabeled (yellow) rather than negative, allowing non-positive regions to be classified as positive. Best viewed in color.

here.

As with other object detection models, given an input image X, the desired output of Faster R-CNN is a bounding box $B^{(i)} \in \mathbb{R}^4$ and class probabilities $c^{(i)} \in \mathbb{R}^k$ for each object (indexed by i) present, where k is the number of classes and the final classification decision is commonly $\arg\max c^{(i)}$. Faster R-CNN does this in a 2-stage process. First, a convolutional neural network (CNN) [LBD+89] is used to produce image features h. A Region Proposal Network (RPN) then generates bounding box proposals $\hat{B}^{(i)}$ relative to a set of reference boxes spatially tiled over h. At the same time, the RPN predicts an "objectness" probability $\hat{c}^{(i)}$ for each proposal, learned as an object-or-not binary classifier. The second stage then takes the proposals with the highest scores, and predicts bounding box refinements to produce $B^{(i)}$ and the final classification probabilities $c^{(i)}$.

Of particular interest is how the classifier producing $\hat{c}^{(i)}$ is trained. Specifically, the cross-entropy loss $H(t, y)$ is employed, where $H(t, y)$ signifies the loss incurred when the model outputs t when the ground truth is y. In the RPN, this results in the following classification risk minimization:

$$R_{pn}^{RPN} = \pi_p \mathbb{E}[H(\hat{c}_p, +1)] + \pi_n \mathbb{E}[H(\hat{c}_n, -1)] \tag{3.1}$$

where π_p and π_n are the class probability priors for the positive and negative classes, respectively, and $\hat{c}_p^{(i)}$ and $\hat{c}_n^{(i)}$ are the predicted "objectness" probabilities for ground truth positive and negative regions. This risk is estimated with samples as:

$$\mathcal{L}_{pn}^{RPN} = \frac{\hat{\pi}_p}{N_p} \sum_{i=1}^{N_p} H(\hat{c}_p^{(i)}, +1) + \frac{\hat{\pi}_n}{N_n} \sum_{i=1}^{N_n} H(\hat{c}_n^{(i)}, -1) \tag{3.2}$$

where N_p and N_n are the number of ground truth positive and negative regions being considered, respectively, and the class priors are typically estimated as $\hat{\pi}_p = \frac{N_p}{N_p+N_n}$ and $\hat{\pi}_n = \frac{N_n}{N_p+N_n}$. Notably, this training loss treats all non-positive regions in an image as negative.

3.2.2 PU learning

In a typical binary classification problem, input data $X \in \mathbb{R}^d$ are labeled as $Y \in \{\pm 1\}$, resulting in what is commonly termed a positive-negative (PN) problem. This implicitly assumes having samples from both the positive (P) and negative (N) distributions, and that these samples are labeled correctly (Figure 3.2a). However, in some scenarios, we only have labels for some positive samples. The remainder of our data are *unlabeled* (U): samples that could be positive or negative. Such a situation is called a positive-unlabeled (PU) setting, where the N distribution is replaced by

an unlabeled (U) distribution (Figure 3.2c). Such a representation admits a classifier that can appropriately include unlabeled positive regions on the correct side of the decision boundary. We briefly review PN and PU risk estimation here.

Let $p(x, y)$ be the underlying joint distribution of (X, Y), $p_p(x) = p(x|Y = +1)$ and $p_n(x) = p(x|Y = -1)$ be the distributions of P and N data, $p(x)$ be the distribution of U data, $\pi_p = p(Y = +1)$ be the positive class-prior probability, and $\pi_n = p(Y = -1) = 1 - \pi_p$ be the negative class-prior probability. In a PN setting, data are sampled from $p_p(x)$ and $p_n(x)$ such that $\mathcal{X}_p = \{x_i^p\}_{i=1}^{N_p} \sim p_p(x)$ and $\mathcal{X}_n = \{x_i^n\}_{i=1}^{N_n} \sim p_n(x)$. Let g be an arbitrary decision function that represents a model. The risk of g can be estimated from $\mathcal{X}_p$ and $\mathcal{X}_n$ as:

$$\hat{R}_{pn}(g) = \pi_p \hat{R}_p^+(g) + \pi_n \hat{R}_n^-(g) \tag{3.3}$$

$\hat{R}_p^+(g) = 1/N_p \sum_{i=1}^{N_p} \ell(g(x_i^p), +1)$ and $\hat{R}_n^-(g) = 1/N_n \sum_{i=1}^{N_n} \ell(g(x_i^n), -1)$, where ℓ is the loss function. In classification, ℓ is commonly the cross-entropy loss $H(t, y)$.

In PU learning, $\mathcal{X}_n$ is unavailable; instead we have unlabeled data $\mathcal{X}_u = \{x_i^u\}_{i=1}^{N_u} \sim p(x)$, where N_u is the number of unlabeled samples. However, the negative class empirical risk $\hat{R}_n^-(g)$ in Equation 3.3 can be approximated indirectly [DPNS15, DPNS14]. Denoting $R_p^-(g) = \mathbb{E}_p[\ell(g(X), -1)]$ and $R_u^-(g) = \mathbb{E}_{X \sim p(x)}[\ell(g(X), -1)]$, and observing $\pi_n p_n(x) = p(x) - \pi_p p_p(x)$, we can replace the missing term $\pi_n R_n^-(g) = R_u^-(g) - \pi_p R_p^-(g)$. Hence, we express the overall risk without explicit negative data as

$$\hat{R}_{pu}(g) = \pi_p \hat{R}_p^+(g) + \hat{R}_u^-(g) - \pi_p \hat{R}_p^-(g) \tag{3.4}$$

where $\hat{R}_p^-(g) = 1/N_p \sum_{i=1}^{N_p} \ell(g(x_i^p), -1)$ and $\hat{R}_u^-(g) = 1/N_u \sum_{i=1}^{N_u} \ell(g(x_i^u), -1)$.

However, a flexible enough model can overfit the data, leading to the empirical risk in Equation 3.4 becoming negative. Given that most modern object detectors

utilize neural networks, this type of overfitting can pose a significant problem. In [KNDS17], the authors propose a non-negative PU risk estimator to combat this:

$$\hat{R}_{pu}(g) = \pi_p \hat{R}_p^+(g) + \max\{0, \hat{R}_u^-(g) - \pi_p \hat{R}_p^-(g)\} \tag{3.5}$$

We choose to employ this non-negative PU risk estimator for the rest of this work.

3.2.3 PU learning for object detection

PU object proposals

In object detection datasets, the ground truth labels represent positive samples. Any regions that do not share sufficient overlap with a ground truth bounding box are typically considered as negative background, but the accuracy of this assumption depends on every object within a training image being labeled, which may not be the case (Figure 3.1). As shown in Figure 3.2b, this results in the possibility of positive regions being proposed that are labeled negative during training, due to a missing ground truth label; effects of this phenomenon during training are investigated empirically in Supplementary Materials Section 3.6.1. We posit that object detection more naturally resembles a PU learning problem than PN.

We recognize that two-stage detection naturally contains a binary classification problem in the first stage. In Faster R-CNN specifically, the RPN assigns an "objectness" score, which is learned with a binary cross-entropy loss (Equation 3.2). As previously noted, the PN nature of this loss can be problematic, so we propose replacing it with a PU formulation. Combining Equations 3.2 and 3.5, we produce

the following loss function:

$$\mathcal{L}_{pu}^{RPN} = \frac{\pi_p}{N_p}\sum_{i=1}^{N_p} H(\hat{c}_p^{(i)}, +1) + \max\left\{0, \frac{1}{N_u}\sum_{i=1}^{N_u} H(\hat{c}_u^{(i)}, -1) - \frac{\pi_p}{N_p}\sum_{i=1}^{N_p} H(\hat{c}_p^{(i)}, -1)\right\} \quad (3.6)$$

Such a loss function relaxes the penalty of positive predictions for unlabeled objects.

Estimating π_p

The PU cross-entropy loss in Equation 3.6 assumes the class-prior probability of the positive class π_p is known. In practice, this is not usually the case, so π_p must be estimated, denoted as $\hat{\pi}_p$. For object detection, estimating π_p is especially problematic because π_p is not static: as the RPN is trained, an increasing proportion of region proposals will (hopefully) be positive. While [KNDS17] showed some robustness to π_p misspecification, this was only on a fairly narrow range of $\pi_p \in [0.8\pi_p, 1.2\pi_p]$. During object detection performance, π_p starts from virtually 0 and grows steadily as the RPN improves. As such, any single estimate $\hat{\pi}_p$ poses the risk of being significantly off the mark during a large portion of training.

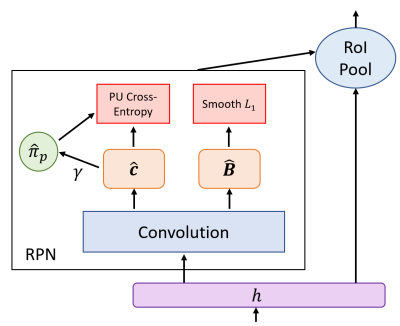


Figure 3.3: Faster R-CNN [RHGS15] RPN with the proposed PU cross-entropy loss. Positive class prior estimate $\hat{\pi}_p$ is updated with objectness predictions $\hat{c}$, with momentum γ.

To address this, we recognize that the RPN of Faster R-CNN is already designed to infer the positive regions of an image, so we count the number of positive regions produced by the RPN and use it as an estimator for π_p:

$$\hat{\pi}_p = \frac{N_p^{RPN}}{N^{RPN}} \quad (3.7)$$

where N^{RPN} is the total number of RPN proposals that are sampled for training, and N_p^{RPN} being those with classifier confidence of at least 0.5. Note that this estimation of π_p comes essentially for free. Given that Faster R-CNN is trained one image at a time and the prevalence of objects varies between images, we maintain an exponential moving average with momentum γ in order to stabilize $\hat{\pi}_p$ (see Figure 3.3). This estimate $\hat{\pi}_p$ is then used in the calculation of the loss $\mathcal{L}_{pu}^{RPN}$ and its gradients.

3.3 Related work

Most object detection frameworks are designed fully supervised [GDDM14b, Gir15, LAE+16, RHGS15, RDGF16, DLHS16, RF17]: it is assumed that there exists a dataset of images where every object is labeled and such a dataset is available to train the model. However, as discussed above, collecting such a dataset can be expensive. Because of this, methods that can learn from partially labeled datasets have been a topic of interest for object detection. What "partially labeled" constitutes can vary, and many types of label missingness have been considered.

Weakly supervised object detection models [BV16, OBLS15, RW15] assume a dataset with image level labels, but without instance labels. These models are somewhat surprisingly competent at identifying approximate locations of objects in an image without any object-specific cues, but have a harder time with precise localization. This is especially the case when there are many of the same class of object in close proximity to each other, as individual activations can blur together, and the lack of bounding boxes makes it difficult to learn precise boundaries. Other approaches consider settings where bounding boxes are available for some classes (e.g., PASCAL VOC's 20 classes) but not others (e.g., ImageNet [DDS+09b] classes). LSDA [HGT+14] does this by modifying the final layer of a CNN [KSH12] to recognize

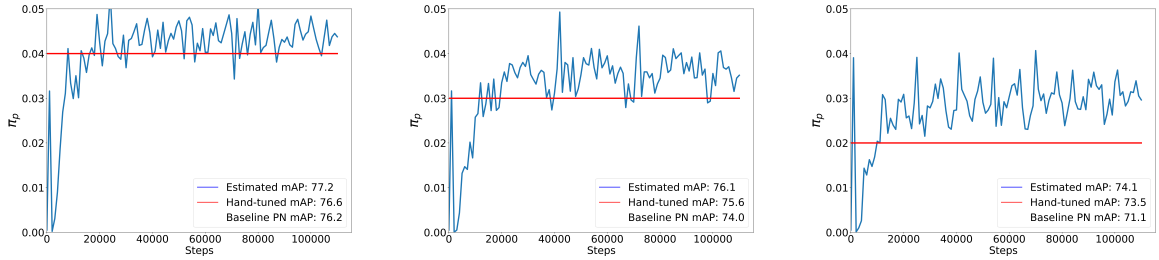
classes from both categories, and [TWG+16] improves upon LSDA by taking advantage of visual and semantic similarities between classes. OMNIA [RGBYO18] proposes a method merging datasets that are each fully annotated for their own set of classes, but not each other's.

There are also approaches that consider a single dataset, but the labels are undercomplete across all classes. This setting most resembles what we consider in our paper. In [DZM+18], only 3-4 annotated examples per class are assumed given to start; additional pseudo-labels are generated from the model on progressively more difficult examples as the model improves. Soft-sampling has also been proposed to re-weight gradients of background regions that either have overlap with positive regions or produce high detection scores in a separately trained detector [WBS+19]; experiments were done on PASCAL VOC with a percentage of annotations discarded and on a subset of OpenImages [KRA+18].

Positive-unlabeled learning has been proposed to improve statistical models with limited labeled data [LL03, LDG00]. [KNDS17, XXXT17] analyzed the theoretical effectiveness of positive-unlabeled risk estimation on binary classification tasks, though with the proportion of positive class examples preset and known to the model. [JWR16, DPNS14] infer the positive class prior with additional classifiers or matching algorithms. Our work extends the application of positive-unlabeled learning to alleviate the incomplete labels that can be found in most object detection datasets.

3.4 Experiments

For our experiments, we use the original Faster-RCNN [RHGS15], with a ResNet101 [HZRS16b] feature extractor (without feature pyramid network [LDG+17]) pre-trained on ImageNet [DDS+09b]. To control the level of missingness of each dataset, each



(a) $\rho = 0.4$ (b) $\rho = 0.5$ (c) $\rho = 0.6$

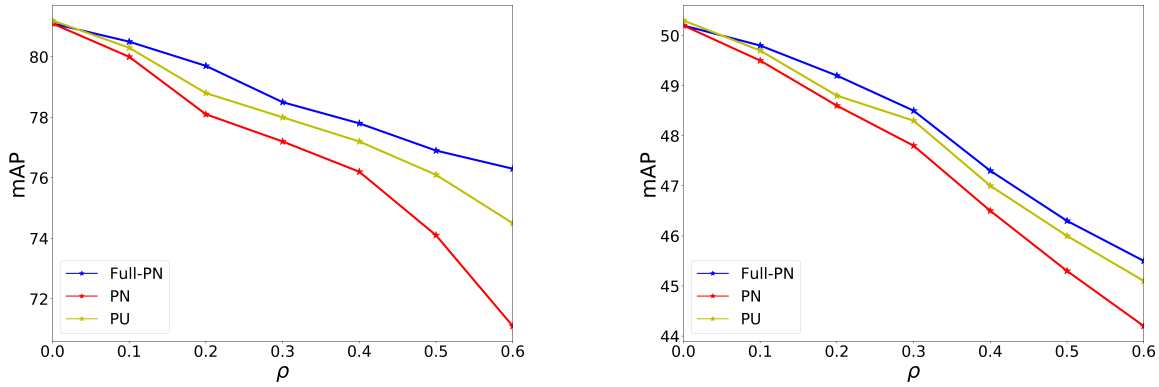
Figure 3.4: Positive class prior $\hat{\pi}_p$ estimated during training of Faster R-CNN on PASCAL VOC versus from hand-tuning π_p as a hyperparameter, for instance label missingness proportion $\rho = \{0.4, 0.5, 0.6\}$.

annotation from a object class provided by original dataset is randomly discarded with a probability ρ. This effectively makes the available annotations to be $1 - \rho$ of total annotations on average.

3.4.1 Hand-tuning versus estimation of π_p

As discussed in Section 3.2.3, PU risk estimation requires the prior π_p. We experiment with two ways of determining π_p. In the first method (*Hand-Tuned*), we treat π_p as a constant hyperparameter and tune it by hand. In the second (*Estimated*), we infer π_p as described in Equation 3.7, setting momentum γ to 0.9. We compare the estimate $\hat{\pi}_p$ inferred automatically with the hand-tuned π_p that yielded the highest mAP on PASCAL VOC. To see how our estimate changes in response to label missingness, when assembling our training set, we remove each annotation from an image with probability ρ, giving us a dataset with $1 - \rho$ proportion of the total labels, and then do our comparison for $\rho = \{0.4, 0.5, 0.6\}$ in Figure 3.4.

In all tested settings of ρ, the estimation $\hat{\pi}_p$ increases over time before stabilizing. Such a result matches expectations, as when an object detection model is first initialized, its parameters have yet to learn good values, and thus the true proportion of positive regions π_p is likely to be quite low. As the model trains, its ability to



(a) PASCAL VOC (b) MS COCO

Figure 3.5: mAP at IoU 0.5 (AP_{50}) on (a) PASCAL VOC and (b) MS COCO, for a range of label missingness ρ.

generate accurate regions improves, resulting in a higher proportion of regions being positive. This in turn results in a higher true value of π_p, which our estimate $\hat{\pi}_p$ follows. As the model converges, π_p (and $\hat{\pi}_p$) stabilizes towards the true prevalence of objects in the dataset relative to background regions. Interestingly, the final value of $\hat{\pi}_p$ settles close to the value of π_p found by treating the positive class prior as a static hyperparameter, but consistently above it. We hypothesize that this is due to a single static value having to hedge against the early stages of training, when π_p is lower. We use our proposed method of auto-inferring π_p for the rest of our experiments, with $\gamma = 0.9$, rather than hand-tuning it as a hyperparameter.

3.4.2 PU versus PN on PASCAL VOC and MS COCO

We investigate the effect that incomplete labels have on object detection training for the popular datasets PASCAL VOC [EVW+10] and MS COCO [LMB+14], using Faster R-CNN [RHGS15] with a ResNet101 [HZRS16b] convolutional feature extractor. In order to quantify the effect of missing labels, we artificially discard a proportion ρ of the annotations. We compare three settings, each for a range of values of ρ. Given

that the annotations are the source of the learning signal, we keep the number of total instances constant between settings for each ρ as follows:

- *PN*: We remove a proportion of labels from every image in the dataset, such that the total proportion of removed labels is equal to ρ, and all images are included in the training set. We then train the detection model with a PN objective, as is normal.
- *Full-PN*: We discard a proportion ρ of entire images and their labels, resulting in a dataset of fewer images, but each of which retains its complete annotations.
- *PU*: We use the same images and labels as in *PN*, but instead train with our proposed PU objective.

A comparison of mean average precision (mAP) performance at IoU 0.5 for these 3 settings on PASCAL VOC and MS COCO is shown in Figure 3.5. As expected, as ρ is increased, the detector's performance degrades. Focusing on the results for *PN* and *Full-PN*, it is clear that for an equal number of annotated objects, having fewer images that are more thoroughly annotated is preferable to a larger number of images with less thorough labels. On the other hand, considering object detection as a PU (*PU*) problem as we have proposed allows us to improve detector quality across a wide range of label missingness. While having a more carefully annotated set (*Full-PN*) is still superior, the PU objective helps close the gap. Interestingly, there is a small gain (PASCAL VOC: +0.2, MS COCO: +0.3) in mAP at full labels ($\rho = 0$), possibly due to better learning of objects missing labels in the full dataset. Additional analysis on the improvements to RPN recall are in Supplemental Materials Section 3.6.2.

3.4.3 Visual Genome

Visual Genome [KZG+17] is a scene understanding dataset of objects, attributes, and relationships. While not as commonly used as an object detection benchmark as PASCAL VOC or MS COCO, Visual Genome is popular when relationships or attributes of objects are desired, as when Faster R-CNN is used as a pre-trained feature extractor for Visual Question Answering [ALA+15, AHB+18]. Given the large number of classes (33,877) and the focus on scene understanding during the annotation process, the label coverage of all object instances present in each image is correspondingly lower. In order to achieve its scale, the labeling effort was crowd-sourced to a large number of human annotators. As pointed out in [EVW+10], even increasing from 10 classes of objects in PASCAL VOC2006 to the 20 in VOC2007 resulted in a substantially larger number of labeling errors, as it became more difficult for human annotators to remember all of the object classes. This problem is worse by several orders of magnitude for Visual Genome. While the dataset creators implemented certain measures to ensure quality, there still are many examples of missing labels. In such a setting, the proposed PU risk estimation is especially appropriate, even with all included labels.

We train ResNet101 Faster R-CNN using both PN and the proposed PU risk estimation on 1600 of the top object classes of Visual Genome, as in [AHB+18]. We

Table 3.1: Detector performance on Visual Genome, with full labels, at various IoU thresholds. "Y" indicates weighting by class frequency, while "N" denotes without weighting.

	AP_{25}		AP_{50}		AP_{75}	
	Y	N	Y	N	Y	N
PN	12.09	22.79	9.11	17.35	2.46	9.98
PU	**13.83**	**25.56**	**10.44**	**19.89**	**4.52**	**11.79**

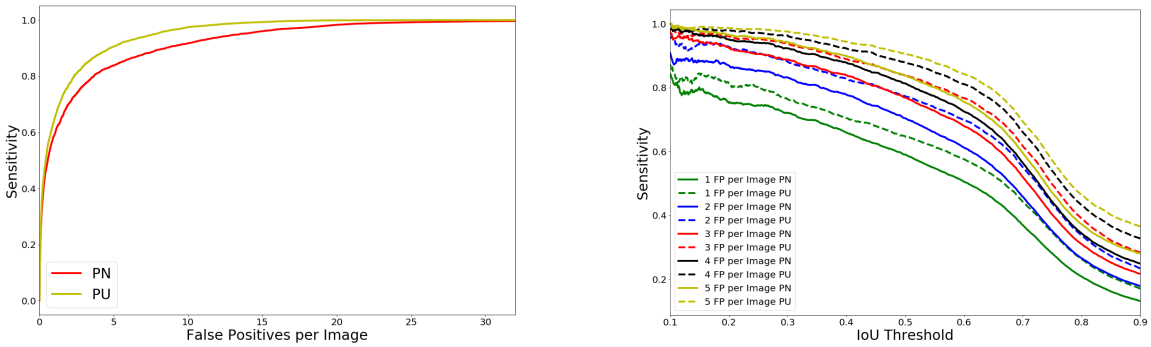


Figure 3.6: Lesion sensitivity versus (a) false positive rate and (b) IoU threshold for different false positive (FP) allowances per image. We compare the baseline Faster R-CNN variant in [YWLS17] trained with a PN objective versus the proposed PU objective.

evaluate performance on the classes present in the test set and report mAP at various IoU thresholds $\{0.25, 0.50, 0.75\}$ in Table 3.1. We also show mAP results when each class's average precision is weighted according to class frequency, as done in [AHB+18]. The PASCAL VOC and MS COCO results in Figure 3.5 indicate that we might expect increasing benefit from utilizing a PU loss as missing labels become especially prevalent, and for Visual Genome, where this is indeed the case, we observe that PU risk estimation outperforms PN by a significant margin, across all settings. Similar improvements are also observed on OpenImages [KDA+16, KRA+18], a dataset with a similar degree of object missingness (Supplemental Materials Section 3.6.3).

3.4.4 DeepLesion

The recent progress in computer vision has attracted increasing attention towards potential health applications. To encourage deep learning research in this direction, the National Institutes of Health (NIH) Clinical Center released DeepLesion [YWLS17], a dataset consisting of 32K CT scans with annotated lesions. Unlike PASCAL VOC, MS COCO, or Visual Genome, labeling cannot be crowd-sourced for most medical

datasets, as accurate labeling requires medical expertise. Even with medical experts, labeling can be inconsistent; lesion detection is a challenging task, with biopsy often necessary to get an accurate result. Like other datasets labeled by an ensemble of annotators, the ground truth of medical datasets may contain inconsistencies, with some doctors being more conservative or aggressive in their diagnoses. Due to these considerations, a PU approach more accurately characterizes the nature of the data.

We re-implemented the modified version of Faster R-CNN described in [YWLS17] as the baseline model and compare against our proposed model using the PU objective, making no other changes. We split the dataset into 70%-15%-15% parts for training, validation, and test. Following [YWLS17], we report results in terms of free receiver operating characteristic (FROC) and sensitivity of lesion detection versus intersection-over-union (IoU) threshold for a range of allowed false positives (FP) per image (Figure 3.6). In both cases, we show that switching from a PN objective to a PU one results in gains in performance.

3.5 Conclusion and future work

Having observed that object detection data more closely resembles a positive-unlabeled (PU) problem, we propose training object detection models with a PU objective. Such an objective requires estimation of the class probability of the positive class, but we demonstrate how this can be estimated dynamically with little modification to the existing architecture. Making these changes allows us to achieve improved detection performance across a diverse set of datasets, some of which are real datasets with significant labeling difficulties. While we primarily focused our attention on object detection, a number of other popular tasks share similar characteristics and could also benefit from being recast as PU learning problems (e.g., segmentation [RFB15,

LSD15a, HGDG17], action detection [SZS12, IZJ+17, GSR+18]).

In our current implementation, we primarily focus on applying the PU objective to the binary object-or-not classifier in Faster R-CNN's Region Proposal Network. A natural extension of this work would be to apply the same objective to the second stage classifier, which must also separate objects from background. However, as the second stage classifier outputs one of several classes (or background), the classification is no longer binary, and requires estimating multiple class priors $\{\pi_c\}_{c=1}^{k}$ [XXXT17], which we leave to future work. Such a multi-class PU loss would also allow extension to single-stage detectors like SSD [LAE+16] and YOLO [RDGF16, RF17]. Given the performance gains already observed, we believe this to be an effective and natural improvement to the object detection classification loss.

3.6 Appendix

3.6.1 Example forgetting in object detection

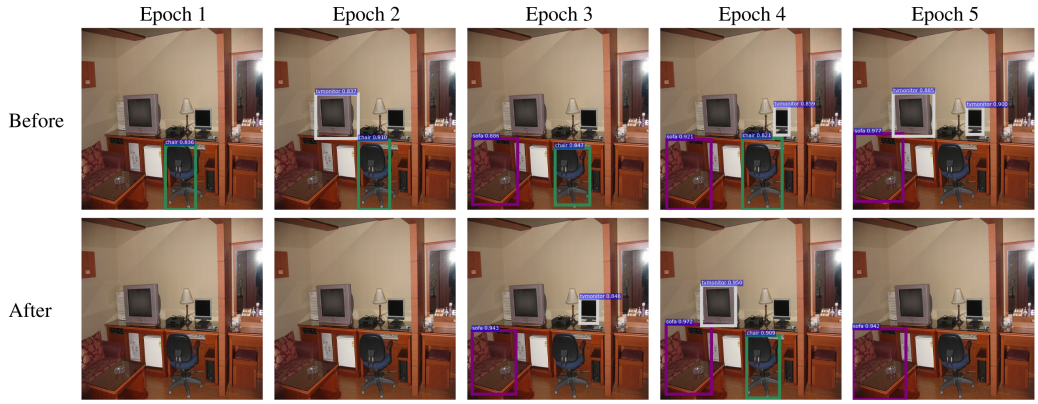


Figure 3.7: Detections on a PASCAL VOC train set image missing annotations throughout training: only the sofa in the lower left has a label. Each column shows detections directly before (top) and after (bottom) the model is trained on the image shown, for each epoch. While the sofa is consistently detected (purple box) after being learned, the unlabeled objects (2 monitors, a chair) are repeatedly found and then suppressed after being trained upon.

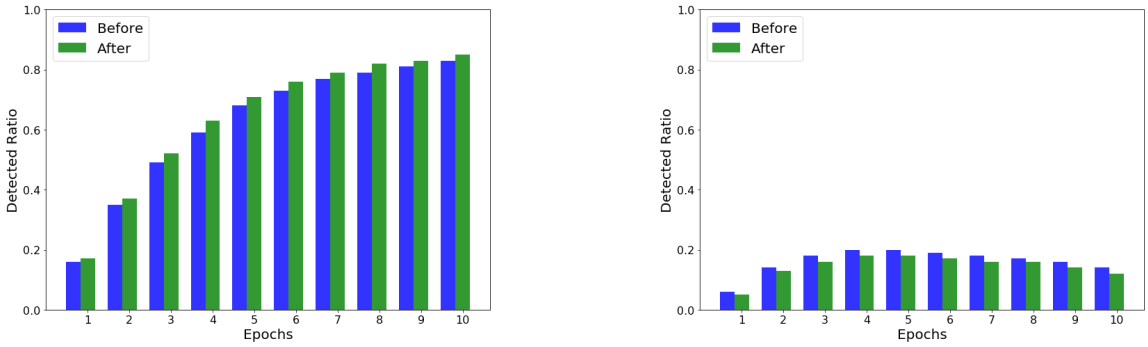
In a recent study of training dynamics of neural network classifiers, [TSdC+19] defined a "forgetting event" as a training example switching from being classified correctly by the model to being classified incorrectly during training. It was found that certain examples were forgotten more frequently than others while others were never forgotten (termed "unforgettable"), with the degree of forgetting for individual examples being consistent across neural network architectures and random seeds. When visualized, the forgotten examples tend to have atypical or uncommon characteristics (e.g., pose, lighting, angle), relative to "unforgettable" examples. Interestingly, a significant number of "unforgettable" examples could be removed from the training set with only a marginal reduction in test accuracy, if the "hard" examples were kept. This implies that the "hard" examples play a role akin to support vectors in

max-margin learning, while easier "unforgettable" examples have little effect on the final decision boundary.

Within the context of object detection datasets, we hypothesize that unlabeled object instances form a similar group of hard examples that are also learned and then forgotten throughout training. Unlike the inter-batch catastrophic forgetting in [TSdC+19], however, where hard examples are learned while part of the current minibatch and then forgotten while learning other examples, unlabeled samples in object detection are learned from other examples and then *suppressed* after incurring misclassification losses during training (see Figure 3.7).

Unlabeled instances strongly resemble positive examples throughout the rest of the dataset, but their lack of labels mean that the typical PN classification objective incentivizes learning them as negatives. Given that hard examples have a strong influence on classifier boundaries, having unlabeled examples trained as negatives may prove especially detrimental to training.

We perform a similar study as [TSdC+19] and investigate forgetting events on PASCAL VOC [EVW+10] by tracking detection rates of labeled and unlabeled instances in the training set throughout learning. In particular, an object is considered detected if the detector produces a bounding box with intersection over union (IoU) of at least 0.5 and the classifier is at least 80% confident in the correct class. We track whether or not an object was detected directly before the image it belongs to is trained upon, and then again after the gradients have been applied. These indicator variables are then combined across objects for each epoch and reported as a percentage. While PASCAL VOC does naturally have unlabeled instances, we do not have access to these without a re-labeling effort. As such, we remove 10% of object annotations randomly across all object classes during training, and use them to calculate detection rates for this experiment.



(a) Labeled objects

(b) Unlabeled objects

Figure 3.8: Detection rates of objects before and after training on their corresponding images for (a) labeled instances and (b) instances with labels withheld during training.

Detection rates for labeled and unlabeled objects over time are shown in Figure 3.8. As expected, the model learns to detect a higher percentage of labeled instances over time, and objects are overall more likely to be detected immediately after the detector trains on them. Despite not having an explicit learning signal, unlabeled objects are still learned throughout training, but at a lower rate than labeled ones. In contrast with labeled objects, unlabeled object detections are discouraged with each PN gradient, leading to a dip in overall detection rates immediately after training. Despite this, overall detection rates of unlabeled objects grows through the first 5 epochs of training, implying a repeated cycle of learning unlabeled objects from other intra-class examples, forgetting them when explicitly trained against them, and then learning them again. Given the undesirability of this forced suppression of detected objects, we seek a method to remedy this behavior.

3.6.2 Recall of Region Proposals

To investigate the effect of the PU risk estimator on the quality of the proposals from RPN stage, we examine the recall of the top 500 proposals, compared with the complete annotations. Higher recall means there are more proposals that match with

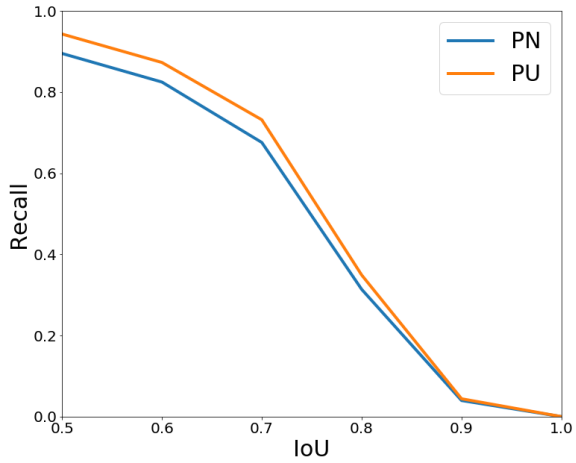


Figure 3.9: Recall of top 500 proposals from RPN after training on PASCAL VOC2007 when $\rho = 0.5$.

the full-labeled annotations. In Figure 3.9, the recalls using the PU risk estimator are higher than those using the PN risk estimator. This illustrates the inclusion of more object proposals that are not included by the PN risk estimator because the corresponding ground truth annotations are missing.

3.6.3 OpenImages

OpenImages [KDA+16, KRA+18] is a large object dataset consisting of 15.4 million bounding boxes from 600 classes across 1.9 million images. In order to achieve its scale, the labeling effort was crowd-sourced to a large number of human annotators. As pointed out in [EVW+10], even increasing from 10 classes of objects in PASCAL VOC2006 to the 20 in VOC2007 resulted in a substantially larger number of labeling errors, as it became more difficult for human annotators to remember all of the object classes. With 500 classes, this problem is worse by an order of magnitude for OpenImages. While the creators of OpenImages designed an annotator training process to insure quality, there still are many examples of missing labels. As such, PU learning as proposed is especially appropriate, even when considering full labels.

As in Section 3.4.2, we train a ResNet101 Faster R-CNN object detector with both PN and PU classification losses. Given the large size of the dataset, we restrict our analysis to 50 of the most prevalent classes, and subsample 140K images from the dataset containing at least one of the selected classes. Of these 140K images, we train on 100K with *full* annotations, and hold out 10K for validation and 30K as our test split. We observe that, all other things equal, switching to our proposed PU approach results in an increase of +3.0, +3.0, and +5.0 for mAPs with IoU thresholds $\{0.25, 0.50, 0.75\}$ (see Table 3.2).

Table 3.2: Detector performance on a subset of OpenImages at various IoU thresholds.

Method	AP_{25}	AP_{50}	AP_{75}
PN	37.7	33.6	20.7
PU	**40.7**	**36.6**	**25.7**

Chapter 4

Federated Unsupervised Visual Representation Learning via Exploiting General Content and Personal Style

4.1 Introduction

Deep learning models are growing exponentially over the demand on endless data. In a centralized setting, all data is accessible to train a deep learning model at a massive scale. However, with the emerging of edge devices, it is more desirable to train a small-sized deep learning model on local training data and aggregate local models to form a global model for a new deployment. [MMR+17] proposes federated learning that enables multiple local clients to collectively train a global model in a iterative updating process without sharing local training data between clients. By labelling each example generated on a local device, federated learning has demonstrated competitive performances on various computer vision tasks [LMX+19, LHL+20, LSD15b]. However, manually labeling all data from numerous local devices is increasingly expensive and concerns users with privacy issue. This substantial requirement leaves most of data on local devices unlabeled. Additionally, data collected and accessible locally is very user-centered, which induces high data heterogeneity across devices due to distinctive user style and preference. Figure 4.1 depicts the learning process on user-centered unlabeled data in a distributed environment.

In the absence of labeled training data, Self-Supervised Learning (SSL) has been developed to pretrain a model with unlabeled data and learn visual representations that are well-generalized to downstream tasks. Discriminative and generative self-

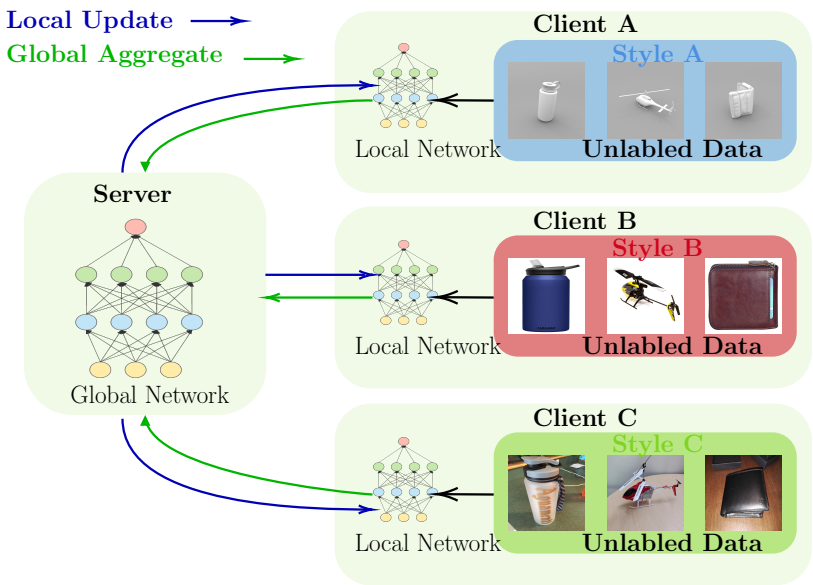


Figure 4.1: Federated unsupervised representation learning with stylized user data. Distinctive user style and preference introduce data heterogeneity and lead to deteriorated global generalization and local personalizations.

supervised methods received growing attentions in recent years [CKNH20, CFGH20, DS19, GSK18, JBZ+20, ZIE16, HCX+22]. Contrastive learning has become the most popular discriminative SSL method that either computes InfoNCE [OLV18a] on positive and negative examples [CKNH20, CFGH20] or predicts projected latent of only positive examples [GSA+20, CH21a, ZJM+21]. The success of the contrastive learning on visual datasets engenders research and study about applying such a method to other formats of data, such as audio data [SGZ21] and medical data [MIM20]. However, the effectiveness of the method in a distributed environment seeks further study.

Applying contrastive learning in a distributed system with stylized local data poses two core challenges. First, without any supervision, learning a generalized global model is more difficult when all local data is diverse in distribution and limited in volume. Second, while securing a generalized global model, customizing personalized local models is also harder when local data is dispersed not only in class distribution

but also in style. These two obstacles require completely different treatments to resolve. Generalization involves a strong regularization to align different local models whilst personalization weakens such regularization to promote customized fit on local dispersed data.

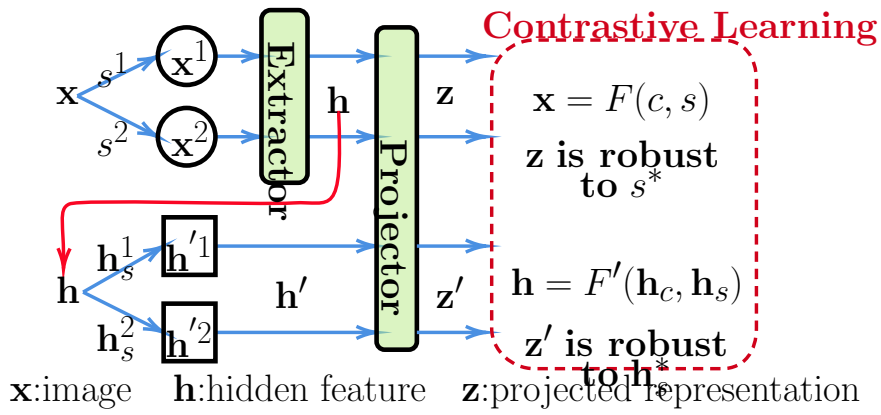


Figure 4.2: Intuition of style infusion to improve representations via contrastive learning.

In this work, we address both generalization and personalization of the federated unsupervised representation learning with the utilization of local style information. In [VKSG+21], authors assume that an image is generated by a content factor (c) and a style factor (s). The authors further prove that, under a self-supervised learning framework, the feature extractor learns content-identifiable representations that are robust to augmentations applied (upper part of Figure 4.2). Based on this argument, our intuition is that a locally extracted image feature is a product of a generalized content feature (h_c) and a personalized style feature (h_s). During self-supervised learning, the content feature is infused with locally extracted style features so that the learned representation is also robust to this latent distortion (lower part of Figure 4.2). Since every client has its own style and preference, infusing extracted local style features into global content features and facilitating personalized features in a decentralized setting become feasible to achieve good results in both generalization

and personalization. We propose Federated with Style (FedStyle) via training a local style feature extractor and combining a content feature extracted via a local content model with the style feature to generate a style-infused representation for contrastive learning. This incorporation of style and content features empowers the local content models to be robust to local distortion in addition to other image-level distortions applied for contrastive learning [CKNH20]. With local models robust to local distortions, a generalized global model robust to various distortions can be aggregated. When deploying on downstream tasks, apart from the generalized content feature, a stylized content feature is also used to promote personalization.

We improve both **generalization** and **personalization** via following contributions:

- We extract style representations locally by contrasting local style with Sobel filtered style.
- By infusing style and content hidden features in the latent space, we learn more robust local models that aggregate a more generlized global model.
- By generating stylized content representation that is better customized to local data, we improve the personalization of local models in the downstream task.

With extensive experiments, we evaluate the effectiveness of FedStyle in terms of generalization and personalization under both IID and non-IID data distributions. To simulate user-centered data style, we evaluate FedStyle on three different data types: 1. Digit consisting MNIST [Den12], USPS [Hul94], and SVHN [NWC+11]; 2. Office Home [VECP17] consisting Art, Clipart, Product, and Real World styles; 3. Adaptiope [RS21] consisting Synthetic, Product, Real Life styles. FedStyle delivers a more generalized global model than the FedAvg baseline and achieves higher

personalized accruacy than personalized federated algorithms: FedRep [CHMS21], FedPer [AASC19], FedProx [LSZ+20], APFL [DKM20], and Ditto [LHBS21].

4.2 Related work

Unsupervised Representation Learning, URL, aims to learn generalized data representations from unlabeled data. Self-supervised Learning (SSL) designs pretext tasks based on either easily constructed labels (discriminative) or reconstructions of original data (generative). Before the surge of contrastive learning, discriminative SSL methods train on manually constructed labels such as colors [ZIE16], rotations [GSK18], positions of split patches [NF16], etc. **Contrastive Learning**, CL, relies on a simple idea that different views of the same data example should be as similar as possible. In other words, the representation of an image should be robust to any distortions applied to the image's appearance. Many CL approaches compute the InfoNCE loss [OLV18a]. Minimizing the loss pulls representations of similar/positive examples closer and pushes representations of dissimilar/negative examples further in the latent space. [CKNH20] formulates augmented views of the same image as positive examples and all other representations in a mini training batch as negative examples. [CFGH20] recognizes the same positive examples but adopts a memory bank to store many momentum-updated representations as negative examples. Other CL methods exclude the notion of negative examples and only predict positive representations from a projected view (BYOL [GSA+20] and SimSiam [CH21a]) or reduce redundancy in the feature dimension space (Barlow Twin [ZJM+21]).

Federated Unsupervised Representation Learning, FURL, adapts URL in a distributed system that learns a common representation model without supervision while preserving data privacy. **Federated Learning** allows selected local models to

be trained on local data and aggregates these local models to form a global model in a iterative communication process. In the absence of labeled data, unsupervised representation learning such as contrastive learning is applied to the distributed system. However, simply aggregating local models causes deterioration in performances due to local data heterogeneity and small local training size. Methods have been developed to mitigate the deterioration. FedCA [ZKY+20] utilizes a dictionary storing representations from clients to address the heterogeneity and an alignment model trained on a public dataset on the server to address the misalignment. FedU [ZGW+21] modifies BYOL that only updates the predictor of the local model if the divergence from the previous communication round is below a certain threshold. [HYM+21] explores the flexibility of centralized CL methods in the distributed system and emphasizes the personalization of local models by updating global parameters locally and updating local parameters separately. [WWZ+21] proposes the fusion of features from other clients to reduce the false negative ratio and feature space matching of neighbouring clients to alleviate misalignment. Nonetheless, these methods may violate the user privacy protocol as they share information between local clients, or they require extraneous architecture for each client, which increases the memory burden at clients' ends. FedX [HPW+22] resolves privacy concerns via distilling knowledge locally and globally with additional loss terms to maxmize the embedding similarity between different views. Nonetheless, the aforementioned methods do not address both global generalization and local personalizations and do not apply to more than one contrastive learning algorithm.

Personalized Federated Learning, PFL, focuses on better local performances with auxiliary local architecture or information. Aggregating diverse local models trained on heterogeneous data causes a slow convergence of the global model and results in poor local performances. Multitask-learning-based PFL [DVT+21, MNB+21]

trains a personalized model for each client by considering each client as a different task and computing similarity between pairwise client relationships. Clustering-based PFL [SMS20, BFA20] clusters local models based on group-level client relationships and adapts new client models based on the clusters. Meta-learning-based PFL [FMO20, KBT19] adapts the global model to a personalized model by finetuning locally. However, the aforementioned types of PFL require extra computational power to compute similarity or finetuning, and this requirement slows down the learning process. Without labeled data, unsupervised learning expects an even longer training time to yield good performance. Model-based PFL learns a global model for the common knowledge while learning local models for local knowledge. To learn the local model only requires regularization to the global model or partial local model update on the fly. FedPer [AASC19] only updates the base of the global model at local ends while finetunes the MLP classifier locally. FedRep [CHMS21] trains the encoder and the classifier separately for different local epochs and only aggregates the local base encoders on the server. APFL [DKM20] learns a convex mixture of a local and a global model as the personalized model. FedProx [LSZ+20] regularizes the local model to the current global model to avoid fast divergence while allowing the local model to converge better on local data. Ditto [LHBS21] extends FedProx by optimizing the global model locally and regularizing the local model to the optimized global model. We compare with these model-based PFL methods as the proposed method includes a style model as the local model and the training involves two models.

4.3 Federated with Style (FedStyle)

Under a decentralized system with unlabeled local data, we adapt unsupervised representation learning to train a generalized global model. Figure 4.3 demonstrates three

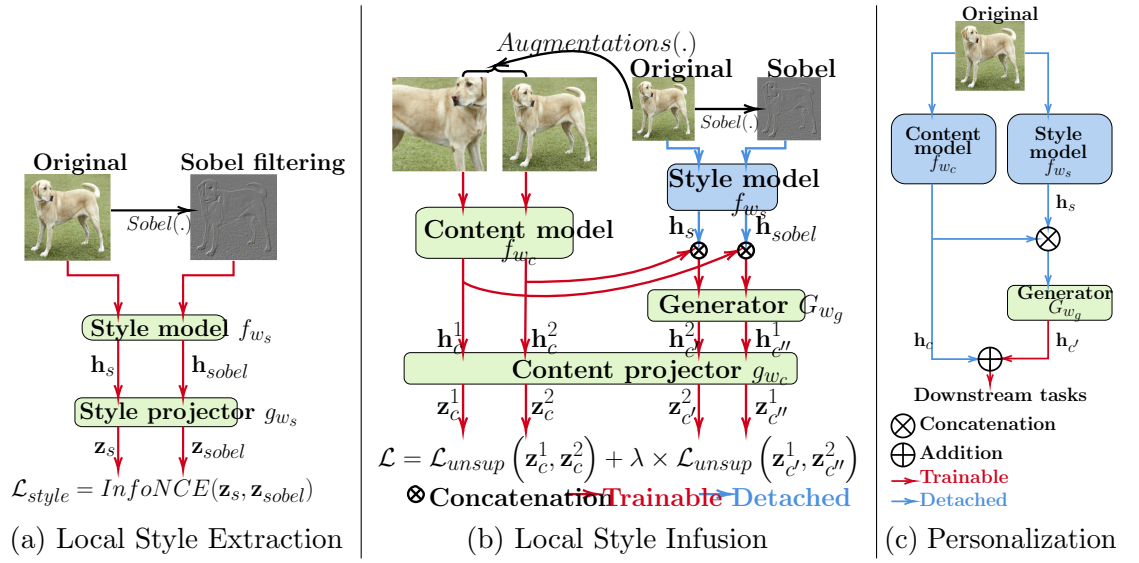


Figure 4.3: Proposed FedStyle process: (a) Extract local style by contrasting original with the Sobel filtered images. (b) Infuse local style feature with the content feature for contrastive learning. (c) Impart both content feature and stylized content feature for personalized tasks.

components of the proposed method. We propose to extract local style information by contrasting local original images with sobel filtered images (4.3a). We propose to improve the generalization of the global model by importing the style information for a style infusion (4.3b) that is beneficial to unsupervised representation learning. And we propose improve personalization of the local models via stylized content features (4.3c) that are tailored to local heterogeneous data. Additionally, we provide an analysis of the effectiveness of proposed style-infused loss on learning better representations.

Problem Setting In a distributed system, there are N clients. The current global model weight is updated to a set $\{C\}$ containing α portion of total clients at each communication round. Each selected client, $k \in \{C\}$, trains a model with weights w_k on local data $\mathcal{D}_k$ for E_l local epochs. After local training, the selected clients send w_k to the server, and the server aggregates it as a global model with weights w. This

process is repeated for E_g communication rounds. The overall goal is to minimize:

$$\min_{w} \mathcal{L}(w) = \frac{1}{N} \sum_{k=1}^{N} \mathcal{L}_k(w, \mathcal{D}_k), \tag{4.1}$$

where $\mathcal{L}_k(w, \mathcal{D}_k)$ is the local loss of client k on data distribution $\mathcal{D}_k$. In this problem, we assume each $\mathcal{D}_k$ has an user style s_k. The federated unsupervised representation learning approach trains a local content model to compute a local loss and averages local content models on the server. Algorithm 1 outlines the overall process of federated unsupervised representation learning.

Algorithm 1 Federated unsupervised representation learning algorithm. $*$ denotes a federated learning method (FedAvg, FedRep, FedProx, FedPer, APFL, Ditto, FedStyle)

Input: Server global model, w. Number of clients, N. Client local models, w_k. Local data distribution, $\mathcal{D}_k$. Communication rounds, E_g. Local training epochs, E_l. Client sample ratio, α. Method arguments, $args$.

```
Initialize w, w_{k,k∈[1:N]}.
for round = 1 : E_g do
    Sample K = ⌊α × N⌋ clients set {C}.
    for i ∈ {C} do
        LocalUpdate_*(w,w_i)
        for epoch = 1 : E_l do
            minimize L_i ←LocalTraining_*(w_i,D_i,args_*)
        end for
    end for
    w = Σ_{k∈{C}} (1/|D_k|) w_k {global aggregate}
end for
```

4.3.1 Local style extraction

We propose to extract local style information for each client by contrasting original local data with a Sobel filtered [KVB88] data which is a strong augmentation with a consistent and distinctive style. (Figure 4.3a as a visual description). This consistent style serves as a reference across clients to isolate user styles without sharing local

information. Each client is equipped with a style model, f_{w_s}, and a style projector, g_{w_s}, to extract a style representation $\mathbf{z}_s$ from each data $\mathbf{x}$. $\mathbf{z}_s = g_{w_s}(f_{w_s}(\mathbf{x}))$. To contrast local style s_k with the Sobel filtered style, we compute a supervised InfoNCE loss to separate local style representations from the Sobel style representations. Furthermore, InfoNCE maximizes the mutual information between data with the same local style [TDR+19] hence the extracted style representations contain as much local information as possible. The style loss of a mini-batch $\mathbf{x}_\mathcal{B} \sim \mathcal{D}_k$ is:

$$\begin{aligned}\mathcal{L}_{style}(\mathbf{x}_\mathcal{B}) = \sum_{\mathbf{x}_i \in \mathbf{x}_\mathcal{B}, i=1}^{2B} \frac{1}{2B-1} \sum_{\mathbf{x}_j \in \mathbf{x}_\mathcal{B}, j=1}^{2B} \mathbf{1}_{i \neq j} \cdot \mathbf{1}_{s_i = s_j} \\ \cdot -log \frac{exp(\mathbf{z}_i \cdot \mathbf{z}_j / \tau)}{\sum_{n=1}^{2B} \mathbf{1}_{i \neq n} \cdot exp(\mathbf{z}_i \cdot \mathbf{z}_n / \tau)},\end{aligned} \tag{4.2}$$

where $B = |\mathbf{x}_\mathcal{B}|$, $\mathbf{z}_i = g_{w_s}(f_{w_s}(\mathbf{x}_i))$ and τ is a temperature term. Each image x_i in the mini-batch is processed via a Sobel filter and this doubles the total number of images in the mini-batch. In the end, $2B$ images have two styles with an equal number. Minimizing (4.2) will maximize the similarity between representations of the same style. A detailed style extraction procedure is listed in Algorithm 2.

Algorithm 2 Local style extraction algorithm.

Input: Local style model, $f_{w_s,k}$. Local style projector, $g_{w_s,k}$. Local data distribution, $\mathcal{D}_k$. Local style extraction epochs, E_s.

```
for epoch=1:E_s do
    for mini batch x_B ~ D_k do
        for x_i ~ x_B do
            h_{s,i} ← f_{w_s}(x_i), h_{sobel,i} ← f_{w_s}(Sobel(x_i))
            z_{s,i} ← g_{w_s}(h_{s,i}), z_{sobel,i} ← g_{w_s}(h_{sobel,i}))
        end for
        g_{w_s}, f_{w_s} ← backward(L_style(x_B))
    end for
end for
```

4.3.2 Local style infusion for contrastive learning

With a style model, each client can extract local style for each local data point. By infusing the extracted local style feature with the generalized content feature, we allow the content model to be also robust to local style distortion. Therefore, we can obtain a more generalized global model when aggregating improved local content models. The process is described in Figure 4.3b. To combine the style feature with the content feature and generate a style-infused content feature, we apply a non-linear MLP as a generator G_{w_g} that takes the concatenation of the content feature and the style feature as inputs. $\mathbf{h}^*_{c'} = G_{w_g}(concat[\mathbf{h}^*_c, \mathbf{h}_{s'}])$. The style-infused feature is passed to the content projector, g_{w_c}, and we compute a style-infused unsupervised loss based on the output of g_{w_c}. The total loss of local training is:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{unsup}(g_{w_c}(\mathbf{h}^1_c), g_{w_c}(\mathbf{h}^2_c)) + \\ \lambda \times \mathcal{L}_{unsup}(g_{w_c}(\mathbf{h}^1_{c'}), g_{w_c}(\mathbf{h}^2_{c''})),\end{aligned} \tag{4.3}$$

where $\mathcal{L}_{unsup}$ can be any general contrastive learning algorithm (SimCLR, MoCo, SimSiam, BYOL, Barlow Twins, etc.), and λ is a scale factor modulating the effect on the style-infused unsupervised learning loss. A detailed algorithm is outlined in Algorithm 3. The clients in FedStyle only transmit local content models to the server. The overall FedStyle algorithm is listed in Algorithm 4.

4.3.3 Personalizing with stylized content feature

For personalized downstream tasks, apart from the learned generalized content feature, we also impart the stylized content feature for the tasks (4.4). By adding the stylized content feature, the linear classifier can learn more customized variations of the image

Algorithm 3 Local style infusion algorithm.

Input: Local content model, $f_{w_c,k}$. Local content projector, $g_{w_c,k}$. Local style model, $f_{w_s,k}$. Local generator, $G_{w_g,k}$. Local data distribution, $\mathcal{D}_k$. Local unsupervised loss, $\mathcal{L}_{unsup}$. Scale on style-infused loss, λ.

freeze $f_{w_s,k}$
for mini batch $\mathbf{x}_\mathcal{B} \sim \mathcal{D}_k$ **do**
  for $\mathbf{x}_i \sim \mathbf{x}_\mathcal{B}$ **do**
    $\mathbf{h}_{s,i} \leftarrow f_{w_s}(\mathbf{x}_i)$, $\mathbf{h}_{sobel,i} \leftarrow f_{w_s}(Sobel(\mathbf{x}_i))$
    $\mathbf{x}_i^1, \mathbf{x}_i^2 \leftarrow Augmentations(\mathbf{x}_i)$
    $\mathbf{h}_{c,i}^1 \leftarrow f_{w_c}(\mathbf{x}_i^1), \mathbf{h}_{c,i}^2 \leftarrow f_{w_c}(\mathbf{x}_i^2)$
    $\mathbf{h}_{c',i}^1 \leftarrow G_{w_g}(concat[\mathbf{h}_{c,i}^1, \mathbf{h}_{s,i}])$
    $\mathbf{h}_{c'',i}^2 \leftarrow G_{w_g}(concat[\mathbf{h}_{c,i}^2, \mathbf{h}_{sobel,i}])$
  end for
  $g_{w_c}, f_{w_c} \leftarrow backward(\mathcal{L}_{unsup}(g_{w_c}(\mathbf{h}_{c,\mathcal{B}}^1), g_{w_c}(\mathbf{h}_{c,\mathcal{B}}^2))$
  $+\lambda \times \mathcal{L}_{unsup}(g_{w_c}(\mathbf{h}_{c',\mathcal{B}}^1), g_{w_c}(\mathbf{h}_{c'',\mathcal{B}}^2)))$
end for

Algorithm 4 FedStyle algorithm

Initialize w, $w_{c,k}$, $w_{s,k}$,$w_{g,k}$, $k \in [1:N]$.
Algorithm 2$(w_{s,k}, \mathcal{D}_k, E_s), k \in [1:N]$
for $round = 1 : E_g$ **do**
  Sample $K = \lfloor \alpha \times N \rfloor$ clients set $\{C\}$.
  for $i \in \{C\}$ **do**
    $w_{c,i} \leftarrow w$
    for $epoch = 1 : E_l$ **do**
      Algorithm 3$(w_{c,i}$,$w_{s,i}$,$w_{g,i}$,$\mathcal{D}_i$,$\lambda)$
    end for
  end for
  $w = \sum_{k \in \{C\}} \frac{1}{|\mathcal{D}_k|} w_{c,k}$
end for

content feature and hence identify the object more correctly. The stylized content feature serves as an additional augmentation that is tailored to local data. For the personalization linear evaluation, we optimize the generator along with the linear classifier.

$$\mathbf{h} = \mathbf{h}_c + G_{w_g}(concat[\mathbf{h_c}, \mathbf{h_s}]) \tag{4.4}$$

4.3.4 Analysis of style-infused representation

During the training of FedStyle, we introduce style-infused representations to compute an additional unsupervised loss in (4.3). Like the first part of the total loss, the second unsupervised loss trains the content model to learn invariant features between two representations infused with different local styles. In [VKSG+21], authors prove that self-supervised learning with data augmentations trains the model to approximate an inverse of a function that simulates the generation process of an image from a content factor and a style factor. The image generation process is assumed to be a smooth and invertible function of the content, c, and the style, s. $\mathbf{x} = F(c, s)$. The global minimum of the unsupervised loss $\mathcal{L}_{unsup}(g_{w_c} \circ f_{w_c}(\mathbf{x}^1), g_{w_c} \circ f_{w_c}(\mathbf{x}^2))$ is reached when $g \circ f$ is approaching the smooth inverse function of F. And the content factor is identifiable from the extracted representation through $g \circ f$. In other words, the $g \circ f$ is learning invariant features that are robust at identifying the content factor. However, representation collapse can happen when $g \circ f$ is not invertible. With different constraints like architecture regularizations (batch norm layer or deeper network [FA20]) to maximize the output entropy of the $g \circ f$, the collapse can be prevented. Similar to the argument, we assume that the image representation in the latent space, $\mathbf{z}$, is a product of the content feature, $\mathbf{h}_c$, and the style feature, $\mathbf{h}_s$. $\mathbf{z} = F'(\mathbf{h}_c, \mathbf{h}_s)$. We deploy the generator, G, so that $g \circ G$ is approximating the F' and the content feature is identifiable from the style-infused representation by minimizing the second part of (4.3). Since this generation process takes place in a low-dimensional latent space, F' can be approximated by two simple networks (g and G are both two-layer MLPs). However, since $g \circ G$ is not invertible, a strong emphasis (large λ) on the second unsupervised loss can cause G to collapse due to the weak architecture regularization. With both unsupervised losses, the local content model

learns robust representations that are invariant to local distortions in the latent space as well as various augmentation distortions on the original data.

4.4 Experiments

Datasets We run experiments on three different data types that each has different number of styles of same objects: Digit (MNIST, USPS, SVHN), Office Home (Art, Clipart, Product, Real World), and Adaptiope (Synthetic, Product, Real Life). Detailed information on each data type is listed in the Appendix 4.7.1. Since Office Home and Adaptiope do not have designed training and testing splits, we run 4 random trials[1] with different training and testing splits (0.7/0.3 training/testing split) and report the average performance with a 95% confidence interval. In IID settings, each client has the same amount of samples across classes, and the total number of images for each client is the same so that local models are optimized with the same training steps. In non-IID settings, the amount of samples in each class on each client is sampled from a Dirichlet distribution parameterized by the concentration β (see Appendix 4.7.2), and all clients have the same amount of total images.

Federated learning settings Both local style model and local content model are ResNet18 [HZRS16a], and the projector and the generator are two-layer MLPs [Hay94] with the ReLU activation. The hidden dimension of the MLPs is set to 256 by default, and the output dimension is set to 128. We run $E_g = 100$ communication rounds, and $E_l = 5$ local epochs. $\alpha = 1.0$ when there is only one client for each style, and $\alpha = 0.5$ when there are $\{5,10\}$ clients for each style. For the FedStyle setting, $E_s = 10$ and λ is listed in the result figures. The input size of Digit datasets is 32 by 32 and other data types is 224 by 224. For other personalized federated learning methods, the adaptation

[1] Random seeds 2011, 2015, 2021, 2022

to federated unsupervised representation learning and the setting are described in Appendix. We use the SGD optimizer with a learning rate of 0.1 for all approaches. The batch size for all experiments is 64. For unsupervised learning methods, we apply SimCLR [CKNH20] and SimSiam [CH21a] since these two methods only require one content model. Hence the memory requirement for each client is minimized. Note that SimSiam has an additional two-layer MLP predictor. The temperature term $\tau = 0.07$ for all InfoNCE loss functions.

4.5 Evaluations and Results

We evaluate both generalization and personalization by linear evaluation. We follow [ZGW+21] for the linear evaluation in a decentralized setting. To evaluate the generalization ability of the aggregated global model, the global model is fixed and a linear classifier is trained on all data in each style dataset and evaluates the Top-1 accuracy on the corresponding testing data. To evaluate the personalization ability of a local model, FedStyle fixes both the local content model and local style model and follow (4.4) to extract local features. Other baseline methods fix local model(s) to extract local features. The linear classifier is trained on local data only and evaluates the Top-1 accuracy on the corresponding testing data in each style dataset. In Figure 4.4 and 4.5, the 95% confidence intervals are indicated as error bars.

Single client in each style dataset In this setting, each style dataset has only one client and all clients are sampled with an equal amount of unlabeled data. We only evaluate the IID setting where there is no heterogeneity in user preferences (class distributions). In Figure 4.4, we present the *personalization* v.s. *generalization* top-1 accuracy of the proposed method and baseline methods. FedStyle is consistently performing better than baseline methods in terms of generalization and personalization.

In terms of personalization ability, FedStyle works better than all baseline methods in all datasets, and on average it achieves the highest accuracy in both SimCLR and SimSiam contrastive learning variations. In terms of generalization, introducing style-infused representations as additional distortions improves the content model more. The improvement of FedStyle shown in Figure 4.4 suggests the effectiveness of the proposed method when there is heterogeneity in user styles.

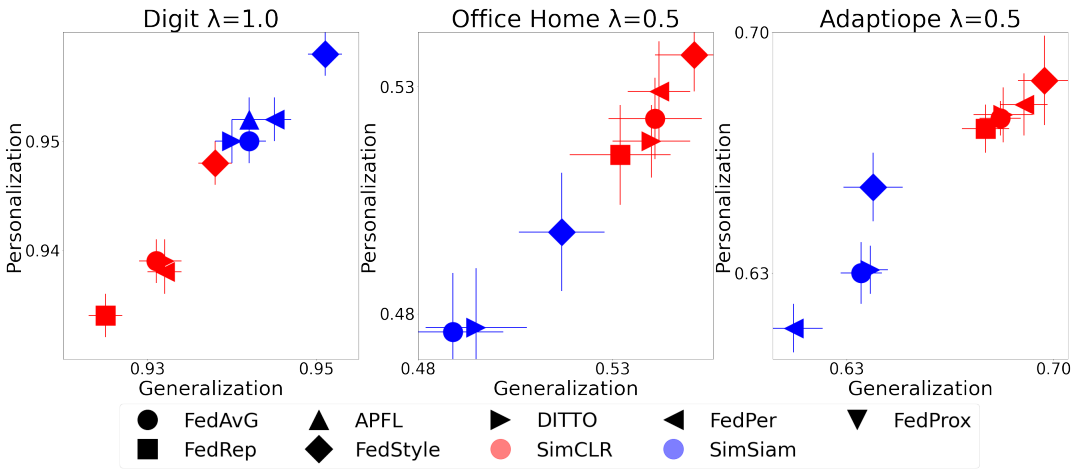


Figure 4.4: Top right section of Generlization v.s. Personalization accuracy. Points with low generalization and/or personalization accuracy are not shown. See Appendix 4.7.4 for detailed results.

Multiple clients in each style dataset Next we evaluate all methods on a slightly larger scale of federated learning. We perform the linear evaluation when there are 5 clients in each style dataset for Office Home and Adaptiope data types and 10 clients in Digit data type with considerations of the size of each data type. We also evaluate all methods in both IID (Homogeneous) and two non-IID (Heterogeneous @$\beta = 0.2$ and @$\beta = 0.8$) settings. As shown in Figure 4.5, FedStyle outperforms all the baselines in both generalization and personalization in all setting of all the datasets. This consistent improvement suggests that FedStyle works at a larger scale where the data is heterogeneous in both style and preference. The overall improvement

seems less significant w.r.t the single client setting because only half of the clients are selected at each communication round. On average, each client is only trained for 250 epochs on fewer data. Under an SSL framework, a larger data size and longer training time are essential to achieve better performance. However, there are cases where FedStyle outperforms baselines to a larger extent (Office Home Heter@0.2 and Heter@0.8).

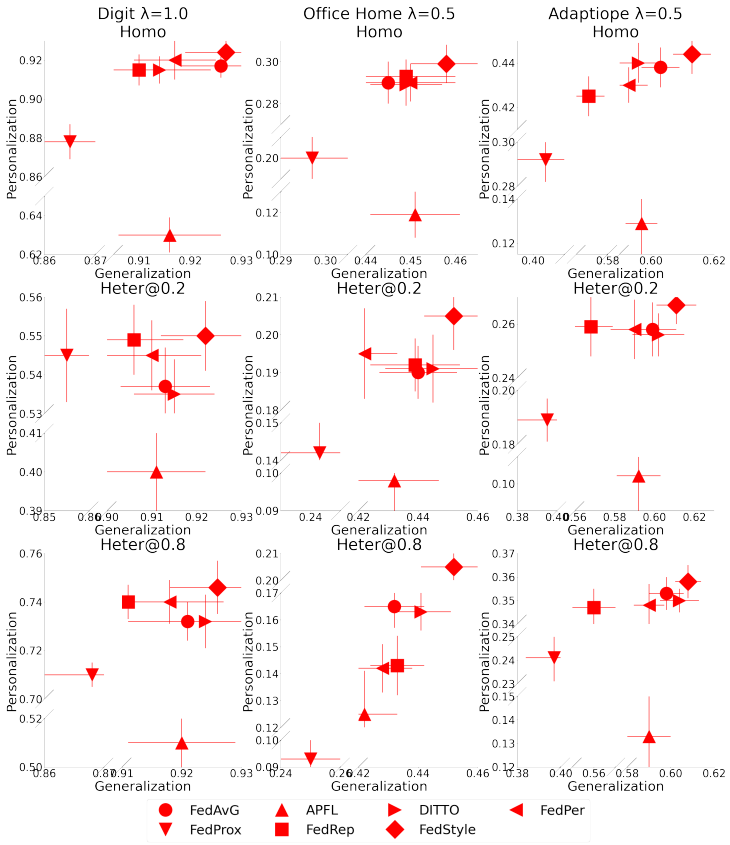
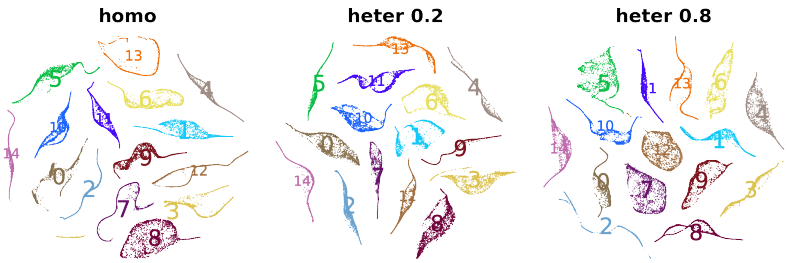
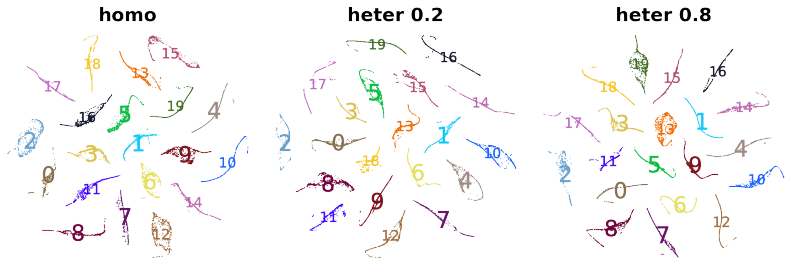


Figure 4.5: **SimCLR** *Personalization* v.s. *Generalization* accuracies for three distribution settings: (1st Row) Homogeneous; (2nd Row) Heterogeneous@$\beta = 0.2$; (3rd Row) Heterogeneous@$\beta = 0.8$. For **SimSiam** see Appendix 4.7.4.

Style feature under non-IID data distribution We visualize the distributions of extracted style features, $\mathbf{h}_s$, of all clients using t-SNE [VdMH08]. As illustrated in Figure 4.6, the relative positions of style features extracted from all clients are consistent across different settings. This suggests that the features extracted via



(a) Adaptiope @ 15 total clients



(b) Office Home @ 20 total clients

Figure 4.6: t-SNE of extracted style feature distributions.

the local style model are consistent with the local style. The shape/distribution of the same client is different across different settings due to the difference in class distributions within each client. The distinctive cluster of each client style feature suggests that the local style model extracts style information that is tailored to the local client and therefore is beneficial for personalized tasks.

4.5.1 Ablation studies

Stylized content feature The benefit of the stylized content feature while deploying the downstream tasks is to improve personalization. By imparting both generalized and stylized content features, the combined feature results in better performance. We quantify this benefit by comparing the performances with only the generalized feature. The comparison results of personalization performance in the single-client setting for each style dataset are shown in Figure 4.7. It is shown that by infusing the stylized features, personalized accuracy improves, which illustrates the effectiveness of the

stylized content features. It is notable that the improvement of the stylized features on Digit dataset is less significant compared with Adaptiope and Office Home. The reason is that the domain gap between the images of different styles in Digit is much smaller than the other two datasets (e.g., MNIST and USPS in Digit are both binary image datasets of handwritten digits).

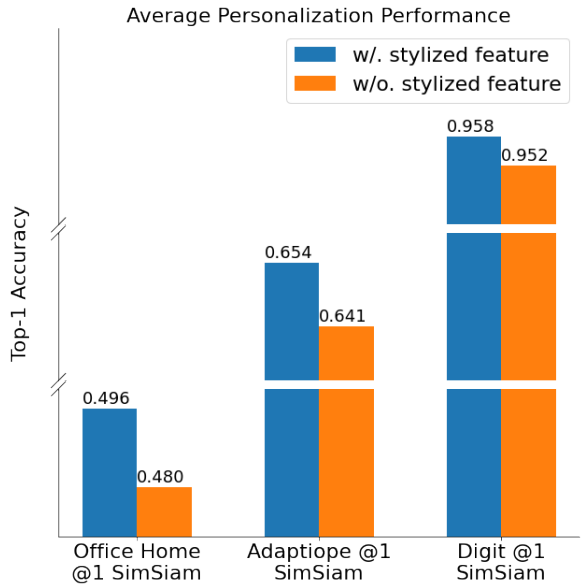


Figure 4.7: Improvement of introducing stylized features for personal downstream tasks.

Style extraction epochs (E_s) To investigate the effect of E_s, we run experiments on Adaptiope @ 1 client with $E_s = \{0, 10, 100\}$. When $E_s = 0$, there is only biased noise infused with the content feature. Though this allows the content model to be robust to biased noise in the latent space, there is an improvement when the trained style feature is infused instead of biased noise. Figure 4.8 shows such improvement when style extraction time increases to 10 and 100 epochs. But a downside of continuing style extraction for a long period is that it drastically increases the computational requirement locally.

Local style-infused loss strength (λ) To investigate the effect of λ on the

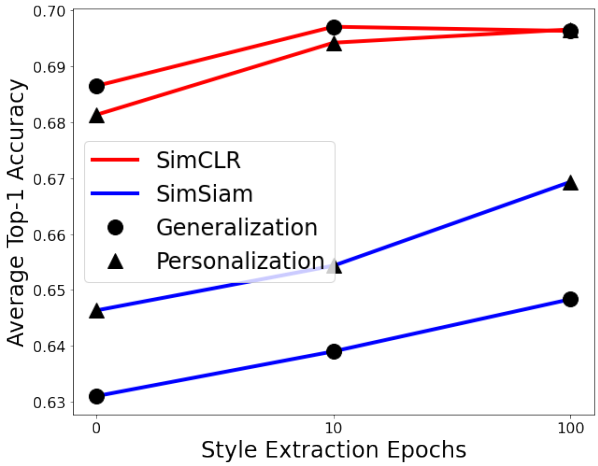


Figure 4.8: The effect of style extraction epochs E_s on the performance.

learned representations, we run experiments with $\lambda = \{0.01, 0.10, 0.50, 1.00, 5.00, 10.0\}$. We report average linear evaluations on Office Home @ 1 client and Adaptiope @ 1 client in Table 4.1. For the SimCLR method, as λ is too large, personalizing with the generator G fails and collapses due to the weak architecture regularization. In the case of Simsiam, both personalization and generalization deliver much lower accuracy when large λ is applied. Simsiam still works to some extent under large λ because it regularizes the output entropy of the model via stop-gradient operation while SimCLR only relies on architectural regularization. The results are consistent with our theoretical analysis in Section 4.3.4.

Table 4.1: Effect of λ on representations.

	Office Home SimCLR @1 Client		Adaptiope SimSiam @ 1 Client	
λ	Generalization Avg.	Personalization Avg.	Generalization Avg.	Personalization Avg.
0.01	0.542±0.009	0.518±0.013	0.634±0.007	0.647±0.011
0.10	0.540±0.010	0.512±0.007	0.635±0.009	0.649±0.015
0.50	0.549±0.011	0.535±0.012	0.639±0.007	0.651±0.011
1.00	0.546±0.010	0.354±0.227	0.624±0.007	0.638±0.010
5.00	0.539±0.010	0.022±0.002	0.542±0.025	0.566±0.024
10.0	0.534±0.012	0.021±0.003	0.552±0.040	0.566±0.036

Generator structures (G) is set to be a two-layer MLP to approximate the generation function (F') in the latent space along with the projector (g), $g \circ G \approx F'$. We experiment with different numbers of layers ($\mathbf{n}$) to investigate the robustness of

the generator to generate stylized content feature for personalization. In Table 4.2, the MLP is robust in both generalization and personalization to different contrastive variations and different dataset when strong architectural constraint is imposed. Nonetheless, consistent with the analysis in Section4.3.4, the personalization collapses due to weaker architectural constraint when a one-layer linear MLP is used.

Table 4.2: Effect of number of layers of G on representations.

	Office Home SimCLR @1 Client		Adaptiope SimSiam @ 1 Client	
n	Generalization Avg.	Personalization Avg.	Generalization Avg.	Personalization Avg.
1	0.545±0.008	0.026±0.002	0.632±0.006	0.008±0.001
2	0.549±0.011	0.535±0.012	0.639±0.007	0.651±0.011
3	0.547±0.010	0.516±0.013	0.635±0.009	0.664±0.008

4.6 Conclusion

In this paper, we propose FedStyle to address the generalization of the global model and personalization of the local models in a user-centered decentralized system with unlabeled data. We propose to extract local style information to improve the global generalization and stylize the content feature to encourage local personalization. We provide an analysis of the effectiveness of style infusion in enhancing the robustness of the learned representation. The proposed method extends to multiple contrastive learning frameworks. In various controlled decentralized settings, FedStyle addresses data heterogeneity caused by heterogeneous user preference and styles.

4.7 Appendix

4.7.1 Information on datasets

We run experiments on the three data types. Each data type consists of different styles of same objects. This simulates the distinctive user style in a distributed system. The detail of each data type is listed in the Table 4.3. To sample equal number of images for each style dataset in a data type, the average number of total training images in all style datasets is sampled (duplicate sampling if necessary).

Table 4.3: Dataset information

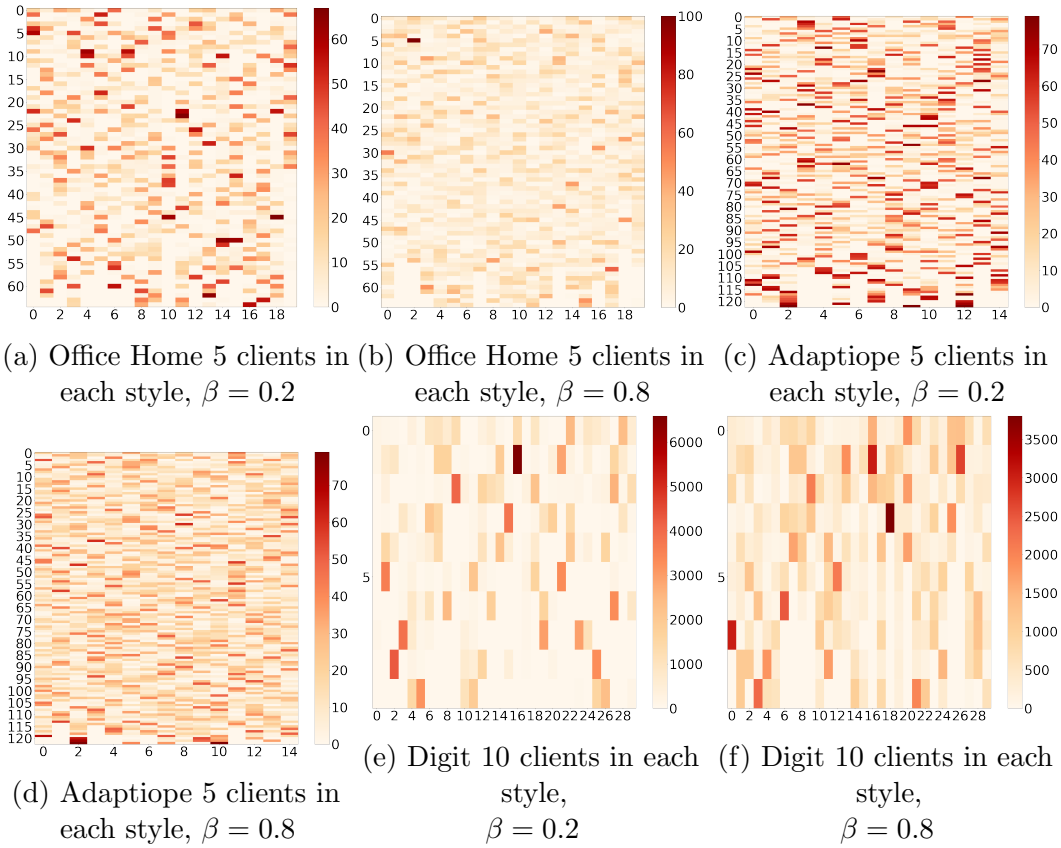
Data types	Style datasets	# object classes	# images
Digit	MNIST/SVHN/USPS	10	70k/∼100k/∼10k
Office Home	Art/Clipart/Product/Real World	65	∼24k/∼44k/∼44k/∼44k
Adaptiope	Sythetic/Product/Real Life	123	12.3k/12.3k/12.3k

4.7.2 Non-IID sampling

To sample different proportions of object classes in each client, we use a Dirichlet distribution with concentration β to determine the number of each class to be sampled. A larger β means a more even distribution among classes. Figure 4.9 shows examples of data distributions with different concentrations for different data types.

4.7.3 Adapting personalized methods for federated unsupervised representation learning

We adapt all baseline methods (FedAvg, FedProx, FedRep, Fedper, APFL, and Ditto) to the federated unsupervised learning since these methods are originally proposed to solve supervised problems. The workflow for all methods is as illustrated in Algorithm



(a) Office Home 5 clients in each style, $\beta = 0.2$

(b) Office Home 5 clients in each style, $\beta = 0.8$

(c) Adaptiope 5 clients in each style, $\beta = 0.2$

(d) Adaptiope 5 clients in each style, $\beta = 0.8$

(e) Digit 10 clients in each style, $\beta = 0.2$

(f) Digit 10 clients in each style, $\beta = 0.8$

Figure 4.9: The data distribution of each client using non-IID data partition. The color bar denotes the number of data samples. Each rectangle represents the number of data samples of a specific class in a client.

1, but details in $args$, $LocalUpdate()$, and $LocalTraining$ are different and will be described in the following sections. Note that w consists of weights in a feature extraction model f and weights in a feature projector g. The default settings for parameters used in the experiments is listed in the $[args:]$ for each method.

4.7.4 Figures for experiment results

$LocalUpdate_*(w, w_k)$

FedAvg

$w_k \leftarrow w$

FedRep, FedPer

$w_k^* \leftarrow w^*$ for $* \in f$

FedProx, APFL, Ditto

$w_{k,g} \leftarrow w$ {Store w temporally as $w_{k,g}$}

$LocalTraining_*(w_k, \mathcal{D}_k, args_*)$

FedAvg, FedPer [$args$: None]

Initialize optimizer
for $\mathbf{x}_\mathcal{B} \sim \mathcal{D}_k$ **do**
  for $\mathbf{x}_i \sim \mathbf{x}_\mathcal{B}$ **do**
    $\mathbf{x}_i^1, \mathbf{x}_i^2 \leftarrow Augmentations(\mathbf{x}_i)$
    $\mathbf{h}_i^1 \leftarrow f_{w_k}(\mathbf{x}_i^1), \mathbf{h}_i^2 \leftarrow f_{w_k}(\mathbf{x}_i^2)$
    $\mathbf{z}_i^1 \leftarrow g_{w_k}(\mathbf{h}_i^1), \mathbf{z}_i^2 \leftarrow g_{w_k}(\mathbf{h}_i^2)$
  end for
  $w_k \leftarrow backward(\mathcal{L}_{unsup}(\mathbf{z}_\mathcal{B}^1, \mathbf{z}_\mathcal{B}^2))$
end for

FedRep [$args$:None]

for $epoch = 1 : E_l$ **do**
  Initialize optimizer, freeze f_{w_k}
  for $\mathbf{x}_\mathcal{B} \sim \mathcal{D}_k$ **do**
    for $\mathbf{x}_i \sim \mathbf{x}_\mathcal{B}$ **do**
      $\mathbf{x}_i^1, \mathbf{x}_i^2 \leftarrow Augmentations(\mathbf{x}_i)$
      $\mathbf{h}_i^1 \leftarrow f_{w_k}(\mathbf{x}_i^1), \mathbf{h}_i^2 \leftarrow f_{w_k}(\mathbf{x}_i^2)$
      $\mathbf{z}_i^1 \leftarrow g_{w_k}(\mathbf{h}_i^1), \mathbf{z}_i^2 \leftarrow g_{w_k}(\mathbf{h}_i^2)$
    end for
    $g_{w_k} \leftarrow backward(\mathcal{L}_{unsup}(\mathbf{z}_\mathcal{B}^1, \mathbf{z}_\mathcal{B}^2))$
  end for
end for
for $epoch = 1 : E_l$ **do**
  Initialize optimizer, freeze g_{w_k}
  for $\mathbf{x}_\mathcal{B} \sim \mathcal{D}_k$ **do**
    Repeat 4-8.
    $f_{w_k} \leftarrow backward(\mathcal{L}_{unsup}(\mathbf{z}_\mathcal{B}^1, \mathbf{z}_\mathcal{B}^2))$
  end for
end for

<u>*APFL*</u> $[args : \alpha = 0.5, w_{k,g}]$

Initialize optimizers
for $\mathbf{x}_{\mathcal{B}} \sim \mathcal{D}_k$ **do**
  for $\mathbf{x}_i \sim \mathbf{x}_{\mathcal{B}}$ **do**
    $\mathbf{x}_i^1, \mathbf{x}_i^2 \leftarrow Augmentations(\mathbf{x}_i)$
    $\mathbf{h}_{i,g}^1 \leftarrow f_{w_{k,g}}(\mathbf{x}_i^1), \mathbf{h}_{i,g}^2 \leftarrow f_{w_{k,g}}(\mathbf{x}_i^2)$
    $\mathbf{z}_{i,g}^1 \leftarrow g_{w_{k,g}}(\mathbf{h}_{i,g}^1), \mathbf{z}_{i,g}^2 \leftarrow g_{w_{k,g}}(\mathbf{h}_{i,g}^2)$
    $\mathbf{h}_i^1 \leftarrow f_{w_k}(\mathbf{x}_i^1), \mathbf{h}_i^2 \leftarrow f_{w_k}(\mathbf{x}_i^2)$
    $\mathbf{z}_i^1 \leftarrow g_{w_k}(\mathbf{h}_i^1), \mathbf{z}_i^2 \leftarrow g_{w_k}(\mathbf{h}_i^2)$
    $\hat{\mathbf{z}}_i^1 \leftarrow \alpha \times \mathbf{z}_{i,g}^1.detach() + (1-\alpha) \times \mathbf{z}_i^2$
    $\hat{\mathbf{z}}_i^2 \leftarrow \alpha \times \mathbf{z}_{i,g}^2.detach() + (1-\alpha) \times \mathbf{z}_i^2$
  end for
  $w_{k,g} \leftarrow backward(\mathcal{L}_{unsup}(\mathbf{z}_{\mathcal{B}}^1, \mathbf{z}_{\mathcal{B}}^2))$
  $w_k \leftarrow backward(\mathcal{L}_{unsup}(\hat{\mathbf{z}}_{\mathcal{B}}^1, \hat{\mathbf{z}}_{\mathcal{B}}^2))$
end for

<u>*FedProx*</u> $[args : \mu = 0.2, w_{k,g}]$

Initialize optimizer
for $\mathbf{x}_{\mathcal{B}} \sim \mathcal{D}_k$ **do**
  for $\mathbf{x}_i \sim \mathbf{x}_{\mathcal{B}}$ **do**
    $\mathbf{x}_i^1, \mathbf{x}_i^2 \leftarrow Augmentations(\mathbf{x}_i)$
    $\mathbf{h}_i^1 \leftarrow f_{w_k}(\mathbf{x}_i^1), \mathbf{h}_i^2 \leftarrow f_{w_k}(\mathbf{x}_i^2)$
    $\mathbf{z}_i^1 \leftarrow g_{w_k}(\mathbf{h}_i^1), \mathbf{z}_i^2 \leftarrow g_{w_k}(\mathbf{h}_i^2)$
  end for
  $w_k \leftarrow backward(\mathcal{L}_{unsup}(\mathbf{z}_{\mathcal{B}}^1, \mathbf{z}_{\mathcal{B}}^2) + \frac{\mu}{2}||w_k - w_{k,g}||^2)$
end for

<u>*Ditto*</u> $[args : \mu = 2, w_{k,g}]$

for $epoch = 1 : E_l$ **do**
  Initialize optimizer
  for $\mathbf{x}_{\mathcal{B}} \sim \mathcal{D}_k$ **do**
    for $\mathbf{x}_i \sim \mathbf{x}_{\mathcal{B}}$ **do**
      $\mathbf{x}_i^1, \mathbf{x}_i^2 \leftarrow Augmentations(\mathbf{x}_i)$
      $\mathbf{h}_{i,g}^1 \leftarrow f_{w_{k,g}}(\mathbf{x}_i^1), \mathbf{h}_{i,g}^2 \leftarrow f_{w_{k,g}}(\mathbf{x}_i^2)$
      $\mathbf{z}_{i,g}^1 \leftarrow g_{w_{k,g}}(\mathbf{h}_{i,g}^1), \mathbf{z}_{i,g}^2 \leftarrow g_{w_{k,g}}(\mathbf{h}_{i,g}^2)$
    end for
    $w_{k,g} \leftarrow backward(\mathcal{L}_{unsup}(\mathbf{z}_{\mathcal{B},g}^1, \mathbf{z}_{\mathcal{B},g}^2))$
  end for
end for
for $epoch = 1 : E_l$ **do**
  Initialize optimizer
  for $\mathbf{x}_{\mathcal{B}} \sim \mathcal{D}_k$ **do**
    for $\mathbf{x}_i \sim \mathbf{x}_{\mathcal{B}}$ **do**
      $\mathbf{x}_i^1, \mathbf{x}_i^2 \leftarrow Augmentations(\mathbf{x}_i)$
      $\mathbf{h}_i^1 \leftarrow f_{w_k}(\mathbf{x}_i^1), \mathbf{h}_i^2 \leftarrow f_{w_k}(\mathbf{x}_i^2)$
      $\mathbf{z}_i^1 \leftarrow g_{w_k}(\mathbf{h}_i^1), \mathbf{z}_i^2 \leftarrow g_{w_k}(\mathbf{h}_i^2)$
    end for
    $w_k \leftarrow backward(\mathcal{L}_{unsup}(\mathbf{z}_{\mathcal{B}}^1, \mathbf{z}_{\mathcal{B}}^2)$
$$+ \frac{\mu}{2}||w_k - w_{k,g}||^2)$$
  end for
end for

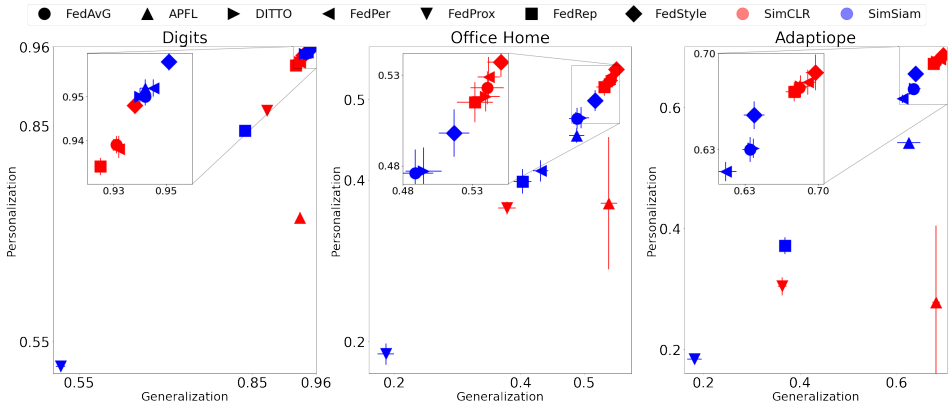


Figure 4.10: Personalization v.s. Generalization for all methods with single client in each style dataset.

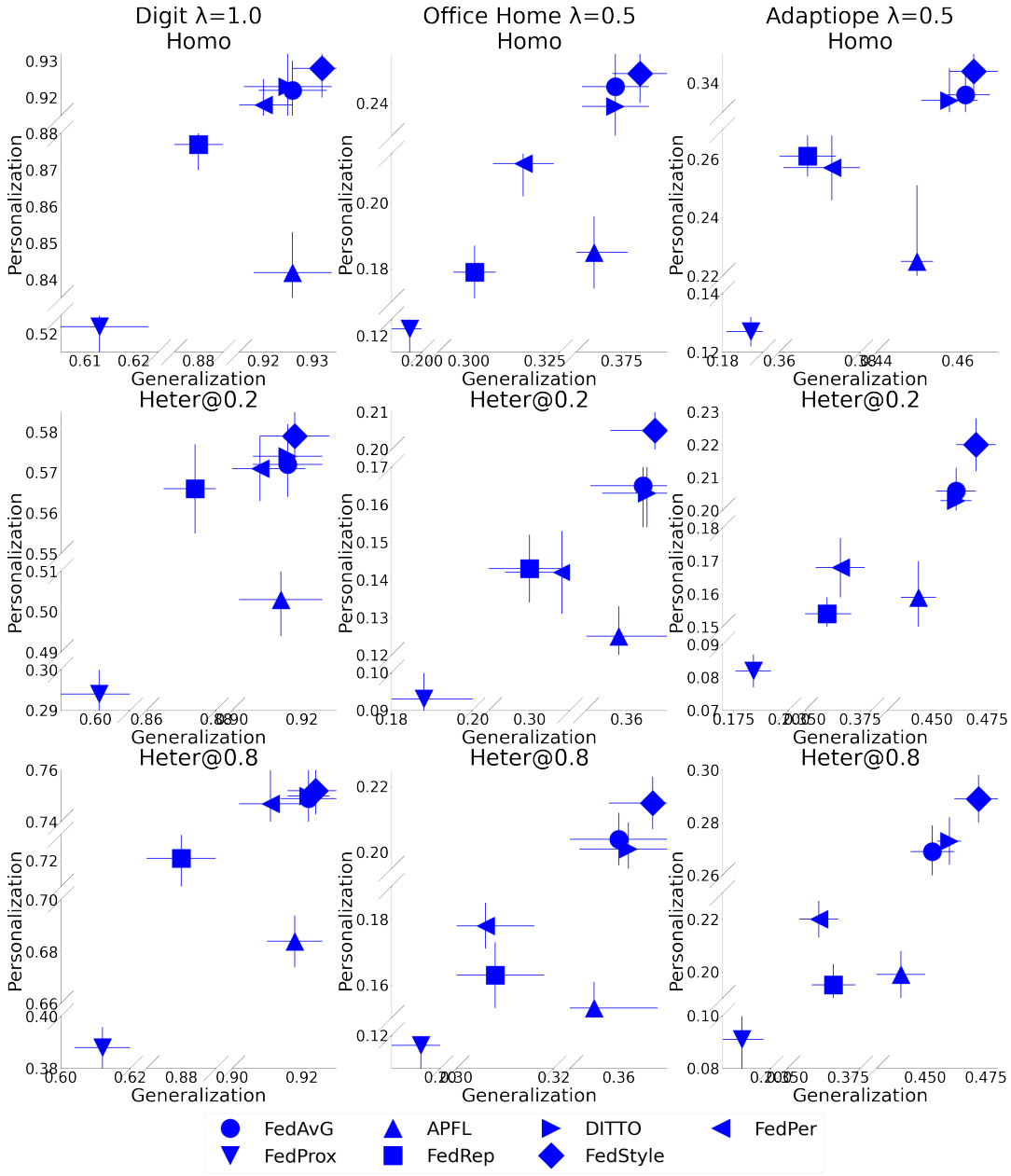


Figure 4.11: **SimSiam** *Personalization* v.s. *Generalization* accuracies for three distribution settings: (1st Row) Homogeneous; (2nd Row) Heterogeneous@$\beta = 0.2$; (3rd Row) Heterogeneous@$\beta = 0.8$.

Chapter 5

Stable and Causal Inference for Discriminative Self-supervised Deep Visual Representations

5.1 Introduction

Learning generalized representation with unlabeled data is a challenging task in various fields, but Self-Supervised Learning (**SSL**) has recently demonstrated remarkable success in learning semantic invariant representations without labels [JBZ+20, JT20, LZH+21]. There are two main types of self-supervised learning (SSL) based on the pretext task used: generative and discriminative SSL, with generative SSL reconstructing altered or distorted data to its original input [CRC+20, HCX+22, HZ93, PKD+16, VLL+10, WFX+22]and early discriminative SSL predicting easily designed labels and task-specific representations that are not very generalizable [GSK18, NF16, ZIE16]. More recent discriminative SSL trains the model to identify similarities and differences between pairs of augmented examples [CMM+20, CKNH20, CH21a, GSA+20, HFW+20, ZJM+21]. The success of SSL in deep image models has resulted in progress in other data modalities [LZH+21, LMRPC+22, LJP+22, SRS22, SD22] and attention-based models like transformers [CXH21, CTM+21a, LYZ+21, XLY+21]. Recent discriminative SSL aims to learn content and semantic invariant representations that are robust to data augmentations, but the learned representations can be *unstable* when one subtle factor of the data is changed to a value that is not accessible through all augmentations. To avoid the high cost of incorporating all possible subtle changes during training, insights are needed to uncover the root cause of instability and find a

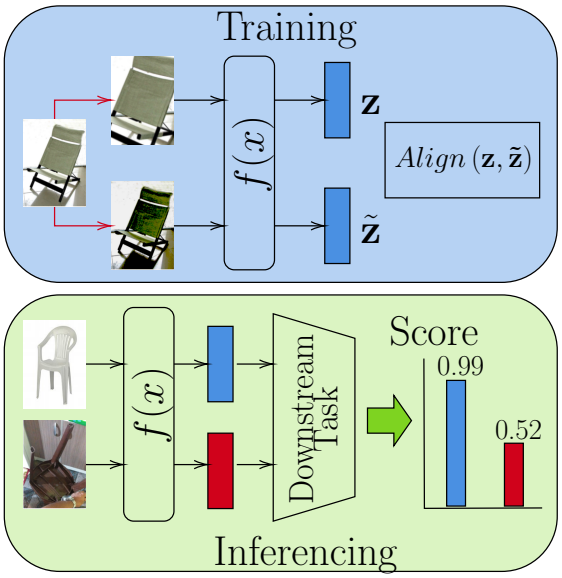


Figure 5.1: During the training of discriminative SSL, aligning positive representations will be robust to the changes applied as augmentations (red arrows). However, during the inference stage, one small change in the data variable (such as *view angle*) will result in an unexpected degradation on downstream performance.

solution to prevent performance deterioration during inference. Figure 5.1 summarizes this deterioration effect.

Causality [Pea09b] is a vital tool to investigate the causal relationships between variables in observational data, and can uncover the underlying causal factors that explain unexpected model behavior due to changes in the environment. Independent Component Analysis (**ICA**) is often used to disentangle sources in unsupervised training [HM16, HM17, HST20, KKMH20, KSS+20], and causal analysis has been applied in follow-up works [ZSS+21, VBS+21] to examine the empirical success of SSL under an ICA framework. However, while these works identify factors that contribute to SSL's success, they do not address the problem's unstable mode, which can cause a severe performance drop when underlying factors shift slightly to an unseen environment. Some works [MMW+20, HWGM21, LLSZ21] attempt to incorporate causality during the training process to identify and alleviate the impact of such shifts, but this approach is time-costly and only marginally improves performance compared

to non-causal SSL methods. A more time-efficient and accessible solution would be to simply reverse the unstable shift during inference.

We aim to address the issue of unstable behavior during the inference stage by building upon previous theories of successful InfoNCE-facilitated contrastive SSL and extending it to all recent SSL methods with additional assumptions and constraints. Drawing inspiration from the relationship between the ground truth positive pairs distribution and learned positive pairs distribution, we demonstrate that the approximated transformation between the ground truth representations and learned representations is orthogonal to the augmentations applied during training. However, a change in the data factor that violates the conditions for successful SSL can cause a corresponding shift in the inferred representation, resulting in a decline in downstream performance. To overcome this issue, we propose learning targeted transformations that regularize the violating shift and restore performance on the unseen data shift. This approach effectively avoids the undesirable behavior and improves performance on previously unseen data shifts.

To summarize, our contributions are following:

- Through the use of a comparable data generation process in prior research, we show that current SSL techniques benefit from the alignment of positive pairs.
- Through our alternative derivation of the transformation matrix between the ground truth representation and the learned representation, we have shown that the augmentations applied during training are orthogonal to the resulting matrix.
- By interpreting a change in the data variable causally, we propose two solutions to modify the negative shift in representation space caused by such a change.

- We validate the proposed solutions by conducting experiments on both controlled and realistic datasets, providing evidence for their efficacy, and suggest potential areas for future research and applications.

5.2 Related Work

Discriminative SSL learns invariant representations from positive pairs of unlabeled data samples, but previous attempts to use trivial labels like colors[ZIE16], rotations[GSK18], and patch positions[NF16] offer minimal benefits for complex downstream tasks due to their easy augmentations. Recent discriminative SSL apply random augmentations to generate two views of an image sample and train an encoder to extract representations for maximizing similarity between the paired augmentations. SimCLR and MoCo[CKNH20, HFW$^+$20] are pioneer SSL works that maximize the cosine similariy between positive pairs and minimize the cosine similarity between negative pairs via optimizing the InfoNCE loss[OLV18b]. Immense resources are used to enforce a large number of negative samples since a larger number can tighten the upper bound of the mutual information between positive pairs[OLV18b]. Later advancement of discriminative SSL excludes the notion of negative pairs by only aligning positive pairs and preventing *representation collapse* through various regularizations. BYOL[GSA$^+$20] predicts an Exponential-Moving-Averaged (EMA) representation of one view with a projected representation of another view. SimSiam[CH21a] maximizes similarity between a projected representation and a detached representation of two positive samples. Unlike previous work focusing optimization on an instance level, Barlow Twins[ZJM$^+$21] encourages high similarity in corresponding feature dimensions and discourages redundancy across different feature dimensions between two views of a data sample. Detailed formulations are exhibited in 5.6.1.

Mutual Information is a different perspective on the behaviour of discriminative SSL. Referred to InfoMax principle[Lin88], the MI between different transformations of a data sample is maximized via optimizing the InfoNCE loss[BHB19, HFLM+18, KTW+20]. Though showing theoretical relation between optimizing InfoNCE and maximizing MI between positive pairs, the underlying factors instructing the behaviour of different SSL methods are not explored. Nonlinear ICA[HP99], on the other hand, captures complex data structures of SSL methods by disentangling underlying factors via minimizing the mutual information between learned representations and the original data input[DKB14, XKTS21, EHRP04]. Other works associate the nonlinear ICA objectives with the contrastive SSL so the MI between positive pairs are maximized and negative pairs are minimized.

Other researchers have explored **causality and causal inference** as a means of understanding the success of discriminative SSL. Prior work has focused on partitioning the InfoNCE loss to *alignment* between the positive pairs and *uniformity* between aggregations of all positive clusters[WI20]. By formulating a data generation process, [ZSS+21] empirically explains that networks optimized via InfoNCE infer an orthogonal transformation of the ground truth latent representations. Furthermore [VBS+21] validates that that augmentations used in both generative and discriminative SSL isolate the *content* factor from the *style* factor. Our theory and work draw great inspirations from these two work. However, instead of solely focusing on InfoNCE-driven SSL and two factors, this work extends the framework to all recent discriminative SSL methods and identifies reasons for unstable circumstances analytically. We also propose methods to nullify the negative effects of unstable representations during inference.

Domain Adaptation is a strategy to bridge the gap between the model performance on a source domain and that on a target domain [BDBCP06, WD18]. Feature

adaptation methods try to learn a new feature representation that is more invariant to the domain shift[VMSM18, DXT12, DI09, LDXT13], while instance adaptation methods try to reweight the importance of the labeled source examples to better align with the target domain[LWD+14, CLW18, HCH+15]. Recent studies also implement a contrastive framework to learn a shared latent space between the source and target domains by maximizing the agreement between representations of corresponding samples while minimizing the agreement between representations of non-corresponding samples [WWW+22, TL21, KJYH19]. Unlike these works focusing on adapting to a target domain for better performance, we investigate the underlying causal factors for the performance gap and based on the findings we propose easy solutions to connect the gap.

5.3 Theory

In this section, we build on previous data generation process [ZSS+21, VBS+21] (5.3.1) and show that current SSL techniques benefit from the alignment of positive pairs (5.3.2). During a deeper dive into the generation process, we show the relation between the ground truth representations and the inferred representations, and this relation is an linear transformation matrix that is orthogonal to the augmentations applied during training (5.3.3). Finally, we disclose the causal reason for the unstable behaviour caused by a change in the data variable during the inference stage and propose analytical solutions to address this unstable issue (5.3.4).

5.3.1 Problem formulation

Data generation is assumed to be a generation function that takes ground truth latent representation as the input to generate an observation data/image.

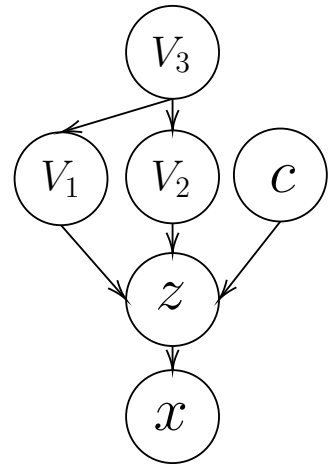


Figure 5.2: A causal graph for data generation process.

Formally, we assume that the marginal distribution of sampling ground truth representations $\mathbf{z} \subseteq \mathcal{Z} \in \mathbb{R}^{d_1}$ w.r.t. a *class* is uniform on a unit sphere $\mathbb{S}^{d_1-1}$. An injective generation function $g(.) : \mathbb{R}^{d_1} \to \mathbb{R}^d$ takes a ground truth representation and generate an observation sample $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{x} = g(\mathbf{z})$. Different variables, denoted as a set $V = \{V_i\}$ and *class/content* variable c, constitute the values of each dimension in the ground truth representation. These variables can be direct causes or confounding factors, $\mathbf{z} = (c, \{V_i\})$. Specific examples include *view angles*, *object size*, *background colors*, etc. A simple causal graph depicts this relationship is shown in Figure 5.2. So the general generation process is as following:

$$p(\mathbf{z} = (c, V_*)) \sim \frac{1}{|\mathcal{Z}|} \qquad \mathbf{x} = g(\mathbf{z}) \tag{5.1}$$

To sample a positive example w.r.t the same *class*, we assume the conditional distribution is a von Mises-Fisher (vMF) distribution [Fis53]:

$$p(\tilde{\mathbf{z}} = (c, \tilde{V}_*)|\mathbf{z} = (c, V_*)) = C_p^{-1} e^{\kappa_1 \mathbf{z}^\top \tilde{\mathbf{z}}} \tag{5.2}$$

where $C_p = \int e^{\kappa_1 \mathbf{z}^\top \tilde{\mathbf{z}}} d\tilde{\mathbf{z}}$ is a normalization constant and κ is a concentration parameter.

Representation learning is a process that a feature encoder, $f(.) : \mathbb{R}^d \to \mathbb{R}^{d_2}$,

extracts representations from two positive observations $f(\tilde{\mathbf{x}}) = f \circ g(\tilde{\mathbf{z}}), f(\mathbf{x}) = f \circ g(\mathbf{z})$ and a distribution associated with the encoder f through $h = f \circ g$ is:

$$q_h(\tilde{\mathbf{z}}|\mathbf{z}) = C_q^{-1}(\mathbf{z}) e^{\kappa_2 h(\tilde{\mathbf{z}})^\top h(\mathbf{z})} \tag{5.3}$$

with $C_q(\mathbf{z}) = \int e^{\kappa_2 h(\tilde{\mathbf{z}})^\top h(\mathbf{z})} d\tilde{\mathbf{z}}$ be the normalization term.[1] Optimizing any discriminative SSL objective will maximize the similarity between these positive pairs. An example of a well constructed objective is the InfoNCE loss:

$$\mathcal{L}_{infoNCE} = \mathbb{E}[-log \frac{e^{f(\tilde{\mathbf{x}})^\top f(\mathbf{x})/\tau}}{e^{f(\mathbf{x})^\top f(\mathbf{x})/\tau} + \sum_{i=1}^{K} e^{f(\mathbf{x}_i^-)^\top f(\mathbf{x})/\tau}}] \tag{5.4}$$

where $\tilde{\mathbf{x}}$ is a positive example w.r.t. $\mathbf{x}$ in the obsevation space and $\{\mathbf{x}_i^-\}_1^K$ are K samples from distributions of all observations. The global minimum of (5.3.1) is reached when the cosine similarity between positive pairs is maximized and the cosine similarity between all negative pairs is minimized. In the following section, we will show that discriminative SSL including InfoNCE-driven and non InfoNCE-driven (EMA and Siamese with a predictor) follows strict rules of alignment to maximize the similarity.

5.3.2 Alignment in discriminative SSL

In this section we will combine theories stated in [ZSS+21, VBS+21, WI20] so that the general factors of successful discirminative SSL can be summarized. Additionally, instead of focusing on just InfoNCE SSL variation and content block-identifiability [HH00, LZYN11], we extend the combined theory to demonstrate that all discrimina-

[1]The mapping of representations on a hypersphere may be different to Barlow Twins methodology, but as shown in [ZJM+21], normalize representations on a unit sphere also works under Barlow Twins.

tive SSL benefit from *alignment* between positive representations.

Theorem 5.3.1. *With a data generation process described in 5.3.1, all discriminative SSL objectives have an alignment loss function between positive pairs from the network:*

$$\mathcal{L}_{align} = \|f(\mathbf{x}) - f(\tilde{\mathbf{x}})\|_2^2 \tag{5.5}$$

This is a weaker statement than **Theorem 4.4** in [VBS+21] since we only focus on the alignment term. For analysis on the regularization term on the network entropy, see 5.6.2.

Proof. For InfoNCE-driven SSL (SimCLR and MoCo), as derived in [WD18, ZSS+21], the InfoNCE loss converges to an *alignment* term and a *uniformity* term as the number of negative samples approaches infinity:

$$\begin{aligned} &\lim_{K\to\infty} \mathcal{L}_{InfoNCE} - logK = \\ &\underbrace{-\frac{1}{\tau} \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [f(\mathbf{x})^\top f(\tilde{\mathbf{x}})]}_{alignment} + \underbrace{\mathop{\mathbb{E}}_{\mathbf{x}}[log \mathop{\mathbb{E}}_{\mathbf{x}^-} [e^{f(\mathbf{x}^-)^\top f(\mathbf{x})/\tau}]]}_{uniformity} \end{aligned} \tag{5.6}$$

with the *alignment* term in (5.6) equivlent to $\mathbb{E}_{(\mathbf{x},\tilde{\mathbf{x}})}(1 - \|f(\mathbf{x}) - f(\tilde{\mathbf{x}})\|_2^2/2)$ and the *uniformity* term equivalent to $-H(f(\mathbf{x})) + logC_q(\mathbf{z})$ [AL76]. Hence complete the proof by:

$$\begin{aligned} &\lim_{K\to\infty} \mathcal{L}_{InfoNCE} - logK = \\ &\frac{1}{\tau} \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [\|f(\mathbf{x}) - f(\tilde{\mathbf{x}})\|_2^2/2 - 1] - H(f(\mathbf{x})) + logC_q(\mathbf{z}) \end{aligned} \tag{5.7}$$

For EMA-based SSL methods (BYOL), a predictor p is associated with the online network so the SSL objective becomes $\mathcal{L} = \mathbb{E}_{(\mathbf{x},\tilde{\mathbf{x}})} \|p \circ f(\mathbf{x}, \theta) - f(\tilde{\mathbf{x}}, \xi)\|_2^2$, where

$\xi^t = \alpha\xi^{t-1} + (1-\alpha)\theta^t$ is target network parameter and θ is the online network parameter. Denote $p' = p \circ f$. By adding and subtracting $p'(\tilde{\mathbf{x}}, \theta)$ we derive:

$$\begin{aligned}
\mathcal{L} &= \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} \|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta) + p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi)\|_2^2 \\
&= \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} \|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta)\|_2^2 + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}} \|p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi)\|_2^2 \\
&\quad -2 \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [(p'(\tilde{\mathbf{x}},\theta) - p'(\mathbf{x},\theta))^\top (p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi))]
\end{aligned} \tag{5.8}$$

Since $f(\mathbf{x},\theta)$ and $f(\tilde{\mathbf{x}},\xi)$ maps in the same space $\mathbb{R}^{d_2}$, p can be considered as a bijective linear transformation within $\mathbb{R}^{b_2}$. In fact, the performance difference between a linear predictor and non-linear predictor is subtle. The global minimizer of (5.8) must align network output w.r.t to $(\mathbf{x}, \tilde{\mathbf{x}})$ with the first term in (5.8) and align outputs from different networks with second term in (5.8). Hence completes the proof.

For Siamese network with a predictor (SimSiam), a similar approach can refomulate the objective as it is a special case when $f(\tilde{\mathbf{x}},\xi) = f(\tilde{\mathbf{x}},\theta)$ and a `stop-gradient` is applied on the $f(\tilde{\mathbf{x}},\theta)$. Hence by substuting $\mathtt{sg}(f(\tilde{\mathbf{x}},\theta))$ with $f(\tilde{\mathbf{x}},\xi)$ in (5.8), we complete the proof by:

$$\begin{aligned}
\mathcal{L} &= \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} \|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta)\|_2^2 \\
&\quad + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}} \|p'(\tilde{\mathbf{x}},\theta) - \mathtt{sg}(f(\tilde{\mathbf{x}},\theta))\|_2^2 \\
&\quad -2 \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [(p'(\tilde{\mathbf{x}},\theta) - p'(\mathbf{x},\theta))^\top (p'(\tilde{\mathbf{x}},\theta) - \mathtt{sg}(f(\tilde{\mathbf{x}},\theta)))]
\end{aligned} \tag{5.9}$$

Note that the third term in (5.8) and (5.9) can be formulated by the differential entropy of $H(p'(.))$ hence prevent representation collapse.

For Barlow Twins, the diagonal of the cross-correlation matrix C_{ii} is the cosine

similarity between positive pairs. Hence completes the proof by:

$$\sum_{i}^{d_2}(1 - C_{ii})^2 = \sum_{i}(1 - (f(\mathbf{x})^\top f(\tilde{\mathbf{x}}))_i/(d_2 - 1))^2 \tag{5.10}$$

$$= \sum_{i}(\|f(\mathbf{x})_i - f(\tilde{\mathbf{x}})_i\|_2^2/(2 * (d_2 - 1))^2 \tag{5.11}$$

5.3.3 Transformation of the ground-truth factors is orthogonal to the applied augmentations

By demonstrating that all discriminative SSL have a alignment loss term, the transformation between the ground truth representations and the inferred representations can be derived as in [ZSS+21]. But different to [ZSS+21], our derivation of minimization of cross entropy is assumed to be a lower bound since we only include the alignment term and the uniformity term is always positive [CK07, BHS19]. However, with all SSL objectives there are additional terms to maximize the output entropy of the model (some described in 5.3.2). So optimizing SSL objectives as a complete loss function will minimize the cross entropy $\mathbb{E}[H(p(.|\mathbf{z}), q_h(.|\mathbf{z}))]$.

Theorem 5.3.2. *By considering the generation conditional distribution as* $p(\tilde{\mathbf{z}}|\mathbf{z}) = C_p^{-1} e^{\kappa_1 \mathbf{z}^\top \tilde{\mathbf{z}}}$*, the inferred conditional distribution* $q_h(\tilde{\mathbf{z}}|\mathbf{z})$ *can match* $p(\tilde{\mathbf{z}}|\mathbf{z})$ *by minimizing* $\|h(\mathbf{z}) - h(\tilde{\mathbf{z}})\|_2^2$ *and* $\forall \mathbf{z}, \tilde{\mathbf{z}} : \kappa \mathbf{z}^\top \tilde{\mathbf{z}} = h(\mathbf{z})^\top \tilde{\mathbf{z}}$ *with* $h = f \circ g$ *or* $h = p \circ f \circ g$ *maps onto a hypersphere with radius* $\sqrt{\kappa_1/\kappa_2}$.

The proof exactly follows **Proposition 1** in [ZSS+21] just with minor modification on the concentration term κ and h so that **Theorem** 5.3.2 can apply to non-InfoNCE SSL. A global minimizer of the alignment term $\|h(\mathbf{z}) - h(\tilde{\mathbf{z}})\|_2^2$ will also minimize the cross entropy of between $p(\tilde{\mathbf{z}}|\mathbf{z})$ in (5.2) and $q_h(\tilde{\mathbf{z}}|\mathbf{z})$ in (5.3). This indicates that minimizers of SSL objective alignment maintain the dot product. Then we

can use **Proposition 2** in [ZSS+21] directly to show that h is an orthogonal linear transformation.

Theorem 5.3.3. *Assume the data generation process (cf. 5.3.1), a model parameterized by $h = f \circ g$ or $h = p \circ f \circ g$ ($h : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$) that minimizes the alignment term in all discriminative SSL objectives: $\|h(\mathbf{z}) - h(\tilde{\mathbf{z}})\|_2^2$ as (5.6), h is an orthogonal transformation: $h(\tilde{\mathbf{z}}) = \mathbf{A}\tilde{\mathbf{z}}$ where $\mathbf{A}$ is an orthogonal matrix.*

Theorem 5.3.4. *Assume the data generation process (cf. 5.3.1), a model parameterized by $h = f \circ g$ or $h = p \circ f \circ g$ ($h : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$) that minimizes the alignment term in all discriminative SSL objectives: $\|h(\mathbf{z}) - h(\tilde{\mathbf{z}})\|_2^2 - H(h(\mathbf{z}))$ as (5.6), h is an orthogonal transformation: $h(\tilde{\mathbf{z}}) = \mathbf{A}\tilde{\mathbf{z}}$ where $\mathbf{A}$ is an orthogonal matrix.*

The proof follows that a function h minimizes the alignment will also minimize the cross entropy between the ground truth conditional distribution $p(\tilde{\mathbf{z}}|\mathbf{z})$ and the inferred conditional distribution $q_h(\tilde{\mathbf{z}}|\mathbf{z})$. Therefore if h is isometric w.r.t the dot product as indicated in **Theorem** 5.3.2 then $h(\tilde{\mathbf{z}}) = \mathbf{A}\tilde{\mathbf{z}}$ according to **Proposition 2** in [ZSS+21].

Theorem 5.3.5. *Assume augmentations applied during the training can be represented as a change in the ground truth representations, i.e. $\tilde{\mathbf{x}} = g(\tilde{\mathbf{z}}(c, \tilde{V*}))$ and change in data variables induces a shift in the ground truth representations $\delta\mathbf{z} = (c, \tilde{V_*}) - (c, V_*)$, then $\delta\mathbf{z}$ is in orthogonal to $\mathbf{A}$ i.e. $\mathbf{A}\delta\mathbf{z} = 0$ if a discriminative SSL objective is optimized.*

Proof. With augmentations normally applied during SSL such as color distortions, rotations, random cropping, and etc., one can view the alteration as a change of a variable in the data variable (color, view angles, sizes). This change of variable under the generation framework described in 5.3.1 will result in a change in $V = \{V_i\}$ since

c is not changed. Regardless of structure of the causal graph shown in Figure 5.2, the change in V can be reflected in the ground truth representation space as $\tilde{\mathbf{z}} = \mathbf{z} + \delta\mathbf{z}$. A global minimizer of any discriminative SSL objective will minimize the alignment term $h(\mathbf{z}) = h(\tilde{\mathbf{z}}) = h(\mathbf{z} + \delta\mathbf{z})$. According to **Theorem** 5.3.4 we derive:

$$h(\mathbf{z}) = \mathbf{A}\mathbf{z} = \mathbf{A}(\mathbf{z} + \delta\mathbf{z}) \tag{5.12}$$

$$\mathbf{A}\delta\mathbf{z} = 0 \tag{5.13}$$

with $\mathbf{A}$ being a linear orthogonal matrix. This indicates that the transformation A is learned to annul the change in data variables or the effect of augmentations applied.

5.3.4 Reason and solutions for unstable change in data variables

In section 5.3.3 we show that generalized representation is robust to augmentations since $\mathbf{A}$, the linear transformation between the ground truth representation and inferred representations, is orthogonal to augmentation applied during the training. And if a change in the data variable reflects a shift, $\delta\mathbf{z}$, in the ground truth representations and the newly inferred representation is stable, meaning $\delta\mathbf{z}$ will be absorbed by the transformation matrix $\mathbf{A}$, then the shift in the ground truth representation corresponds to an augmentation that is applied during the training. However, when $\delta\mathbf{z}$ appears in the range of $\mathbf{A}$ and $\mathbf{A}\delta\mathbf{z} \neq 0$, the resultant inferred representation can be unstable and lead to performance drop on downstream models (denote as $D(\mathbf{z})$). We quantify this deterioration on D by:

$$m(D(h(\mathbf{z}))_{stable}) - m(D(h(\tilde{\mathbf{z}})_{unstable}) \tag{5.14}$$

where $m(.)$ is a metric on an outcome of the downstream task. In example of D being a linear classifier, $m(.)$ can be the probability of predicting the target class. In order to overcome this deterioration, we propose two methods:

1. Under a stable condition, $D(f(\mathbf{x}))^2 = D(\mathbf{A}\mathbf{z})$, each dimension in the inferred representation $f(\mathbf{x})$ is a linear combination of dimensions of the ground truth representation. The dimensions contributing most to $D()$ should be robust to unstable shift $\delta\mathbf{z}_{unstable}$. In other words, most robust dimensions of $f(\mathbf{x})$ should be also robust in $f(\tilde{\mathbf{x}})$ where $\tilde{\mathbf{x}} = g(\mathbf{z} + \delta\mathbf{z}_{unstable})$ as some dimensions of $\mathbf{A}\delta\mathbf{z}_{unstable}$ will be zero. Hence identifying most important dimensions in $f(\mathbf{x})_{stable}$ and pass through the same dimensions of $f(\tilde{\mathbf{x}})_{unstable}$ should alleviate the deterioration by making $m(D(h(\mathbf{z}))_{stable})_{\{dim\}} = m(D(h(\tilde{\mathbf{z}})_{unstable})_{\{dim\}}$ where $\{dim\}$ is a set of most robust dimensions. In example of D being a linear classifier, the contribution of each dimension can be calculated by $W_c^\top f(\mathbf{x})$ where $W_c^\top$ is the Jacobian of the linear classifier w.r.t target class c.

2. Since $f(\mathbf{x})_{stable} - f(\tilde{\mathbf{x}})_{unstable} = -\mathbf{A}\delta\mathbf{z}$, we can learn another linear transformation $\mathbf{F}$ to absorb $\delta\mathbf{z}$. Especially, we want to learn $\mathbf{F}f(\tilde{\mathbf{z}})_{unstable} = \mathbf{F}\mathbf{A}\mathbf{z} + \mathbf{F}\mathbf{A}\delta\mathbf{z}$ such that the additional $\mathbf{F}$ will not only set $\mathbf{A}\delta z$ to 0, but also make stable representation more robust to the unstable shift by assuring $\mathbf{FA}$ orthogonal to $\delta\mathbf{z}_{unstable}$ in addition to the augmentations applied during training. Formally, we model a linear layer $l(f(\tilde{\mathbf{x}})_{unstable}) = f(\mathbf{x})_{stable} - f(\tilde{\mathbf{x}})_{unstable}$, hence during the inference $f(\mathbf{x}) + l(f(\mathbf{x}))$ is used for downstream task.

Relation to Causal Inference Since we only have observations in the image space, we consider the augmentations or changes of a data variable as *interven-*

[2]Note that in general f contains a projector. However we exclude the notion of the projector to simplify the problem and [CKNH20, HFW+20, GSA+20] show that a projector is not necessary in SSL.

tions on the ground truth representations: $\tilde{\mathbf{x}} = g(\tilde{\mathbf{z}}|(c, do(V_i = v_i))$. With an access to ground truth representation, we can evaluate treatment-control effect i.e. $Pr(f(g(\tilde{\mathbf{z}}|(c, do(V_i = v_{stable}))) - Pr(f(g(\tilde{\mathbf{z}}|(c, do(V_i = v_{unstable}))))$. Without any access to the ground truth representations, we can evaluate the average treatment effect [Hol86] by $\mathbb{E}[D(f(\mathbf{x})_{stable}) - D(f(\tilde{\mathbf{x}})_{unstable})]$ via synthesizing manual data samples $\tilde{\mathbf{x}}_i = g(\tilde{\mathbf{z}}|(c, do(V_i = v_i))$.

5.4 Experiments and Discussions

In this section we evaluate our solutions to unstable shift in data variable on two datasets: Causal3DIdent and ImageNet pretrained SSL and corresponding linear classifiers as the downstream task.

5.4.1 Causal3DIdent

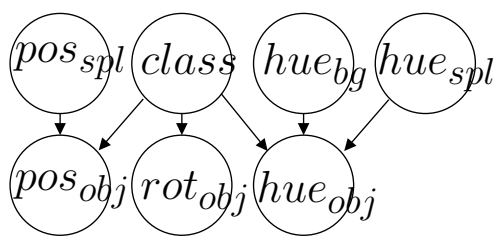


Figure 5.3: Causal graph for V in Causal3DIdent.

[ZSS+21] develops the dataset that 7 objects in the dataset that each 224×224 image is associated with an object class and a 10-dimensional latent representation. These 10 dimensions correspond to 10 data variables. A causal graph imposed on these variables is shown in Figure 5.3. As the dependency shown in Figure 5.3, an object is placed at $pos_{obj} = (x, y, z)$ with $rot_{obj} = (\phi, \theta, \psi)$ and hue_obj under a spot light at an angle pos_{spl} with color hue_{spl} on a background with color hue_{bg}. Detailed

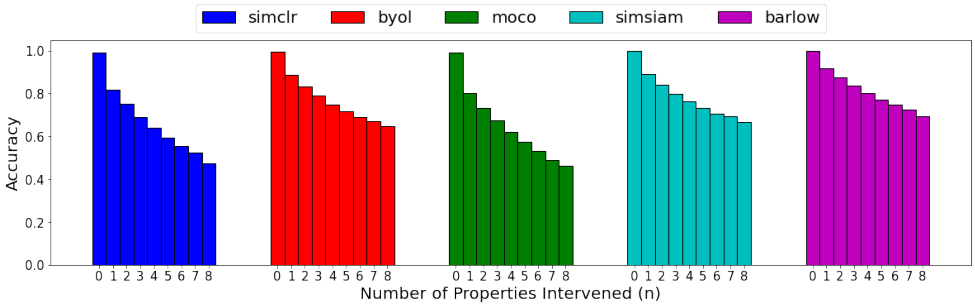
information in 5.6.3.

To simulate unstable change of variable that is not accessible during the training, a range of some data variables is hold out and hence this portion of data is treated as potential unstable samples that causes the performance drop on the linear classifier. Since all data variables are truncated in range [-1,1], we hold out the edge value(s) to further portray 'unexpected' values during the inference time. We select 8 dimensions to intervene, namely: z in object position[3], all 3 object rotation angles, spot light position, and all 3 hue variables. Both training and testing data are sampled and for detailed sampling procedures refer to 5.6.3.

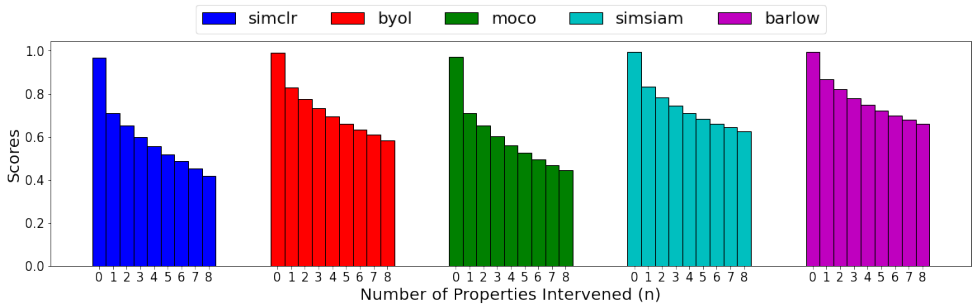
SSL Experiment Setup ResNet18[HZRS16a] is the backbone of the feature encoder f. Same augmentations in [CKNH20] are applied during training. An Adam optimizer with a learning rate at 0.0001 and a weight decay at 0.00001 is optimized for all discriminative SSL. Hyperparameters for each SSL is presented in 5.6.3. The dimension of the inferred representation space is set to 128. The network is trained for 20 epochs on intervened data. Then a linear classifier is trained on the frozen representations of the network with a same optimizer for 10 epochs.

To illustrate the simulation results in the deterioration of the downstream performance, for each testing data that has the same data variable distribution as the training data, we search in the ground truth representation space for 5 nearest representations in the hold-out test distribution when n dimension of the testing data is shifted to a random value in the hold-out range of corresponding data variable. Within the corresponding 5 images representing a shift in n dimensions that is not seen during the training, we record the lowest performance to fulfill the deterioration scenario. In Figure 5.4 the score and accuracy are averaged over all combinations of $\binom{8}{n}$. When $n = 0$, this indicates the performance of SSL on the testing data with the

[3]Only z is altered because most of deep vision models are translation invariant.



(a) Deterioration in Accuracy

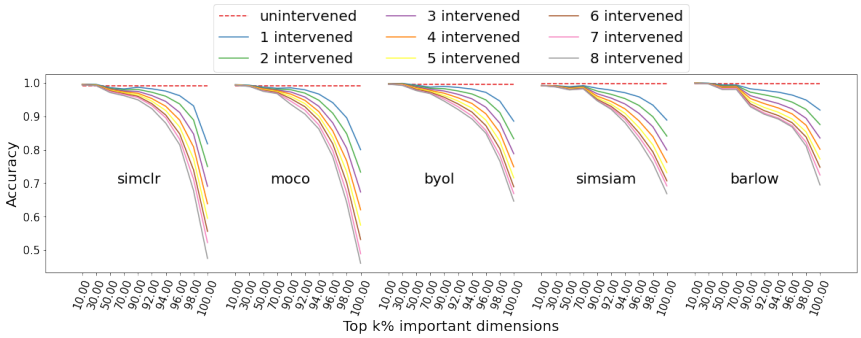


(b) Deterioration in Score

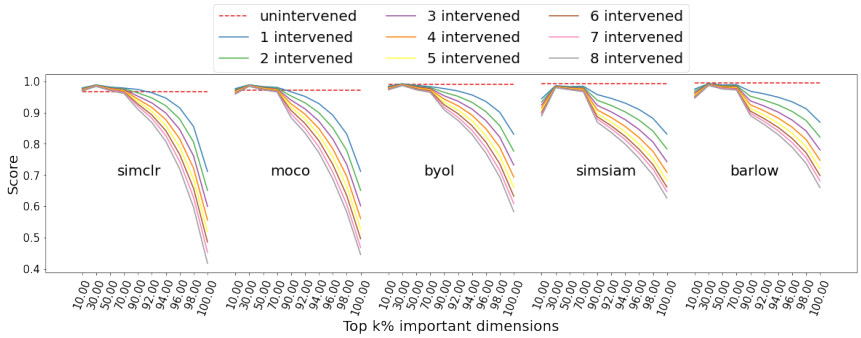
Figure 5.4: Deterioration of unstable changing in data variables on all SSL. The more unexpected changes occur in the data variables, the more severe the deterioration.

same data variable distribution as the training data. The performance is exceptional since hold-out values of some data variables may be covered by the augmentations applied (z covered by random cop, and hue colors covered by color distortions), $\mathbf{A}$ will be orthogonal to changes in these data variables. In fact in 5.6.3 we show that the performance difference between seen distribution and unseen distribution in testing data is comparatively small. As illustrated in Figure 5.4, even when only one variable is changed to a unseen value, there is a large drop in both accuracy and prediction score (20% in accuracy and 30% in prediction score for SimCLR). With more tangled changes in data variables, the unstable representations results in poorer downstream performance. Since the selection of the data variables may be too complex due to the dependency, we also validate the same issue on selecting only children nodes in Figure 5.3. See 5.6.3 for more results.

Solution 1 For each pair of testing data $\mathbf{x}_{stable}$ with seen data variable distribution and the selected data $\mathbf{x}_{unstable}$ among the 5 nearest neighbours when changing n dimensions to unseen distribution values, we apply the Jacobian of the linear classifier w.r.t the target class $W_c^\top$ on the stable representation $f(\mathbf{x})_{stable}$ to identify the top k most important dimensions and pass the same dimensions of $f(\tilde{\mathbf{x}})_{unstable}$ to D to evaluate the performance. As shown in Figure 5.5, the accuracy only deteriorate slightly when top 90% most important features of $f(\tilde{\mathbf{x}})_{unstable}$ are selected for the downstream task. This is true even when all 8 variables are shifted. This suggests that $\mathbf{A}$ is orthogonal to changes in ground truth representation $\mathbf{z}$ in most dimensions. This high percentage of dimensions may be due to f optimizes the SSL objectives to a high level and the augmentations applied covers some of the hold-out variable values. Interestingly, there are cases where passing the top k% (around 40%) important dimensions results in higher performance than stable representations (unitervened in Figure 5.5). However, as expected, including more less important dimensions where



(a) Improvement on Accuracy



(b) Improvement on Score

Figure 5.5: By identifying and passing top k % important dimensions, the deterioration in both accuracy and prediction score is significantly alleviated.

the unseen shift in the data variable results in non-zero adjustment ($\mathbf{A}\delta\mathbf{z} \neq 0$) initiates the deterioration phenomenon we observe.

Solution 2 As shown in 5.3.4, a linear transformation $\mathbf{F} : \mathbb{R}^{d_2} \to \mathbb{R}^{d_2}$ is trained to cancel the effect $\mathbf{A}\delta\mathbf{z}$ and also further improve the robustness of transformation on the ground truth representation $\mathbf{Az}$. In this experiment, we match each training data with a random member in the 5 nearest representations when only **one** dimension is changed to a unstable value. $(\mathbf{x}, \tilde{\mathbf{x}})$. A linear layer is trained with the same optimizer for 10 epochs on the training pairs. In Table 5.1, all accuracy for unstable examples increases significantly (except SimSiam) after learning $\mathbf{F}$. However, since the f is very close to the global minimizer of the alignment term, the improvement on stable examples cannot be observed.

Table 5.1: The effect of $\mathbf{F}$ on both stable and unstable samples.The accuracy is average over 3 random seeds.

	$\mathbf{x}_{stable}$		$\tilde{\mathbf{x}}_{unstable}$	
	w/o $\mathbf{F}$	w/ $\mathbf{F}$	w/o $\mathbf{F}$	w/ $\mathbf{F}$
SimCLR	0.996	0.998	0.833	0.889
MoCo	0.992	0.992	0.800	0.849
BYOL	0.996	0.998	0.886	0.920
SimSiam	0.998	0.999	0.919	0.928
Barlow Twins	0.991	0.995	0.818	0.862

5.4.2 ImageNet

To validate our findings on a larger scale and more realistic settings, we apply proposed solutions on ImageNet[DDS+09a] with various altered and synthetic datasets as unstable shift in the data variable. See Figure 5.6 and 5.6.4 for more information on **ObjectNet, Stylized-ImageNet, Synthetic dataset**.

ObjectNet[BMA+19] is a collection of objects that are intentionally placed at an unusual view angle and backgrounds, so that the bias learned by the model with usual

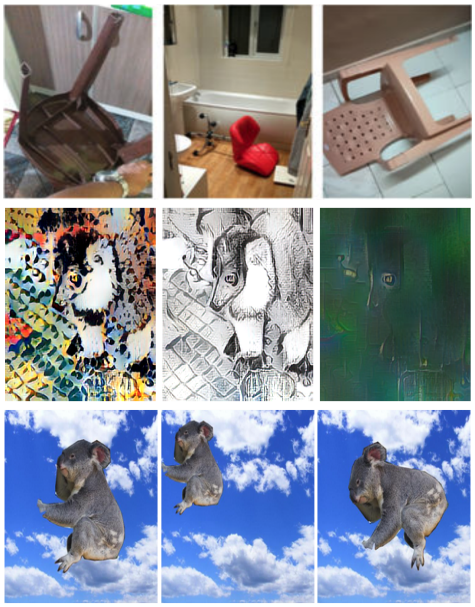


Figure 5.6: Examples of ObjectNet dataset (1st row), Stylized ImageNet dataset (2nd row), Synthetic dataset (3rd row).

data variables is more prominent when testing on ObjectNet. With focusing on 113 classes overlapping with ImageNet classes, we evaluate our second proposed solution explicitly on ObjectNet since the shift presented in ObjectNet is very unstable and it is very challenging to overcome the negative effect of the shift.

Stylized ImageNet[GRM+19] change the style of original ImageNet image to a random artistic style. With this drastic shift in the data variable, the analysis of the first solution is more insightful on the robustness of dimensions of $\mathbf{A}$ on a very different shift in data variable.

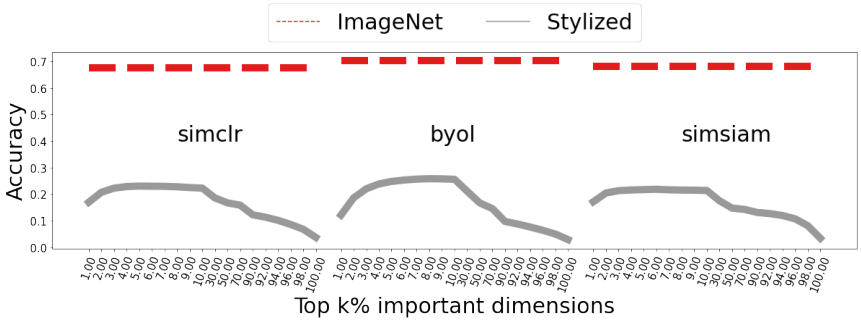
Synthetic Data follows synthetic procedure in [DYT+20] where object is masked on a background at a location with a rotation angle. We explore the benefit of this synthesized dataset at three modes: *background, location, rotation* where the target variable is randomly sampled with the other two variables fixed. Additionally, we explore *texture* as an independent variable by masking 'texturized' objects in [GMW+19]. We set total number of updating steps per epoch as 100 with batch size

256. This means we select total number of 512000 synthesized images every epoch.

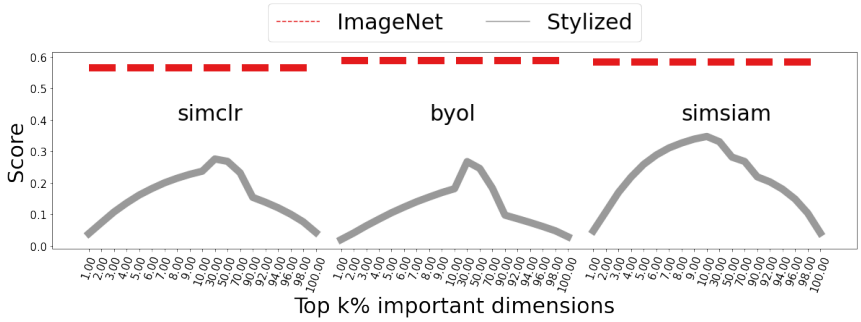
Experiment Setup ResNet 50 pretrained on ImageNet and a linear classifier fintuned via SimCLR, BYOL, and SimSiam are tested with both proposed solutions. For solution 2, a linear layer is optimized with an Adam optimizer for 10 epochs on the synthetic dataset.

Solution 1 For each of ImageNet validation data sample, we stylized the image to a random artistic fashion. We observe the dramatic performance difference between the ImageNet stable images and Stylized unstable ImageNet images $(\mathbf{x}_{stable}, \mathbf{x}_{unstable})$. The result is shown in Figure 5.7. For SimCLR, passing the top 10% important dimensions can close a small performance gap between stable representations and unstable representations. Nonetheless, all SSL seem to be sensible to the strong style change as they shorten the difference between ImageNet to an modest extent. This is as expected since the Stylized ImageNet changes multiple variables to an extreme value.

Solution 2 As described in **Synthetic Data**, we explore the benefit of our second proposed method to evaluate on a dataset that most of samples are unstable according to learned $\mathbf{A}$. At each training step, with other variables randomly fixed, 10 images with random target variable values are generated. The pair with maximum $m(D(\mathbf{x})) - m(D(\tilde{\mathbf{x}}))$ is selected to train the linear transformation. In Table 5.2, inferring $\mathbf{F}$ via controlling location produces least improvement. This is expected since the network is robust to translation by design. While background, rotation, and texture improves the performance considerably with the consideration on the training time. However, in 5.6.4 we show that training the model longer using **Solution 2** yield less favorable results since the improvement is less significant and starts to saturate at around 30 epochs.



(a) Improvement on Accuracy



(b) Improvement on Score

Figure 5.7: By identifying and passing top k % important dimensions, the deterioration in both accuracy and prediction score is alleviated.

Table 5.2: Comparison of inferring different **F** on ObjectNet with a target data variable random trials.

	SimCLR	BYOL	SimSiam
w/o F	10.12	14.04	11.15
background	12.34	15.43	12.45
location	10.35	14.13	11.17
rotation	12.04	15.88	12.38
texture	12.79	15.06	13.11

5.5 Conclusions

In conclusion, this paper has proposed a novel approach to address the issue of unstable behavior during the inference stage in SSL methods. By building on the previous theories of successful InfoNCE-facilitated contrastive SSL and extending it to recent SSL methods, we have demonstrated that a change in the data factor can result in a shift in the inferred representation, leading to a decline in downstream performance. We have proposed learning targeted transformations that regularize the violating shift and restore performance on the unseen data shift. Our experiments on both controlled and realistic datasets have shown the efficacy of our proposed solutions. These contributions provide a better understanding of SSL methods and offer a promising solution to the problem of unstable behavior during the inference stage. We hope that our work will inspire further research in this area and lead to improved SSL methods that are more robust to changes in data factors.

5.6 Appendix

5.6.1 Recent Descriminative SSL Formulations

Table 5.3: Notations for important elements in SSL. **Note that the extracted or projected representations are normalized to a unit sphere unless specified otherwise.**

Notation	Description
$\mathbf{x}_i$	An image sample
$\mathcal{A}_1(\mathbf{x}_*), \mathcal{A}_2(\mathbf{x}_*)$	Two random augmentations
$f_\theta(.)$	A feature encoder parameterized by θ
$f_\xi(.)$	An EMA encoder parameterized by ξ
τ	A temperature scaling term
$p_\theta(.)$	An MLP predictor parameterized by θ.
`sg`	`stop-gradient` operation.

SimCLR[CKNH20] optimizes InfoNCE to maximize similarity between positive pairs and minimize similarity between negative pairs. Positive pairs are two augmented views of an image sample, $\tilde{\mathbf{x}}_{2i}, \tilde{\mathbf{x}}_{2i+1} = \mathcal{A}_1(\mathbf{x}_i), \mathcal{A}_2(\mathbf{x}_i)$. Negative pairs are all other augmented samples in a mini training batch. For a batch with N image samples, the augmentation produces $2N$ augmented samples. A feature encoder $f_\theta(.)$ extracts representations of the batch data $\tilde{\mathbf{z}}_i = f_\theta(\tilde{\mathbf{x}}_i)$. SimCLR optimizes the following objective:

$$\mathcal{L}_{SimCLR} = \mathop{\mathbb{E}}_{i}[-log\frac{e^{\tilde{\mathbf{z}}_{2i}^\top \tilde{\mathbf{z}}_{2i+1}/\tau}}{\sum_{j=1,j\neq 2i}^{2B} e^{\tilde{\mathbf{z}}_{2i}^\top \tilde{\mathbf{z}}_{j}/\tau}}]$$

MoCo/MoCo V2 optimzes InfoNCE to maximize similarity between positive pairs and minimize similarity between negative pairs. Positive pairs are two augmented views of an image sample, $\tilde{\mathbf{x}}_{2i}, \tilde{\mathbf{x}}_{2i+1} = \mathcal{A}_1(\mathbf{x}_i), \mathcal{A}_2(\mathbf{x}_i)$. But negative pairs are representations learned via a moving-averaged network, $f_\xi(.)$, and stored in a memory bank with size K, $\mathcal{M}_K$. And $\xi = m\xi + (1-m)\theta$, where m is a momentum coefficient.

$\tilde{\mathbf{z}}_{2i} = f_\theta(\tilde{\mathbf{x}}_{2i})$, $\hat{\mathbf{z}}_{2i+1} = f_\xi(\tilde{\mathbf{x}}_{2i+1})$, $\hat{\mathbf{z}}_j = f_\xi(\tilde{\mathbf{x}}_j) \in \mathcal{M}_K$. MoCo optimizes the following objective:

$$\mathcal{L}_{MoCo} = \mathop{\mathbb{E}}_{i}[-log\frac{e^{\tilde{\mathbf{z}}_{2i}^\top \hat{\mathbf{z}}_{2i+1}/\tau}}{\sum_{j=1}^{K} e^{\tilde{\mathbf{z}}_{2i}^\top \hat{\mathbf{z}}_{j}/\tau}}]$$

BYOL aligns the projection of a representation of an augmented sample with an EMA representation of another augmented sample. The main difference between BYOL and SimCLR/MoCo is the claim that BYOL only formulates the objective on positive pairs. $\tilde{\mathbf{x}}_{2i}, \tilde{\mathbf{x}}_{2i+1} = \mathcal{A}_1(\mathbf{x}_i), \mathcal{A}_2(\mathbf{x}_i)$. An MLP predictor, $p_\theta(.)$, further projects the representation extracted by the feature encoder to an embedding space and the *EMA* representation predicts the projected embedding/representation by alignment. $\tilde{\mathbf{z}}_{2i} = f_\theta(\tilde{\mathbf{x}}_{2i})$,$\hat{\mathbf{z}}_{2i+1} = f_\xi(\tilde{\mathbf{x}}_{2i+1})$. BYOL optimizes the following objective:

$$\mathcal{L}_{BYOL} = \mathop{\mathbb{E}}_{i} \|p_\theta(\tilde{\mathbf{z}}_{2i}) - \hat{\mathbf{z}}_{2i+1}\|_2^2$$

SimSiam aligns the projection of a representation of an augmented sample with a detached representation of another augmented sample. Unlike BYOL or MoCo, SiamSiam omits the EMA encoder that the author deem to be unnecessary for a stable representation learning. An MLP predictor, $p_\theta(.)$, further projects the representation extracted by the feature encoder to an embedding space and the *detached* representation predicts the projected embedding/representation by alignment. $\tilde{\mathbf{z}}_{2i} = f_\theta(\tilde{\mathbf{x}}_{2i})$,$\hat{\mathbf{z}}_{2i+1} = f_\xi(\tilde{\mathbf{x}}_{2i+1})$[4]. SimSiam optimizes the following objective:

$$\mathcal{L}_{SimSiam} = \mathop{\mathbb{E}}_{i}[-\frac{p_\theta(\tilde{\mathbf{z}}_{2i})}{\|p_\theta(\tilde{\mathbf{z}}_{2i})\|_2} \cdot \frac{\hat{\mathbf{z}}_{2i+1}}{\|\hat{\mathbf{z}}_{2i+1}\|_2}]$$

Barlow Twins aligns positive representations of two augmented samples in feature dimensions and reduces redundancy cross different feature dimensions. Different

[4]The $\mathbf{z}$ is not normalized yet, since in the loss function both projected and extracted representations are normalized.

to all aforementioned SSL methods, the authors suggest to standard normalize the representation (zero mean and unit std) instead of unit sphere normalization. However, as stated in the paper, either normalization scheme works under Barlow Twin method. $\mathcal{Z}^A = \{\tilde{\mathbf{z}}_{2i}\}_{i=1}^N = \{f_\theta(\tilde{\mathbf{x}}_{2i})\}_{i=1}^N, \mathcal{Z}^B = \{\tilde{\mathbf{z}}_{2i+1}\}_{i=1}^N = \{f_\theta(\tilde{\mathbf{x}}_{2i+1})\}_{i=1}^N$. And $\mathcal{Z}^A$ and $\mathcal{Z}^B$ are normalized over the batch statistics. Barlow Twins optimizes the following objective:

$$\mathcal{L}_{BarlowTwins} = \sum_a (1 - \mathcal{C}_{aa})^2 + \lambda \sum_a \sum_{b \neq a} \mathcal{C}_{ab}{}^2 \tag{5.15}$$

$$\mathcal{C}_{ab} = \frac{\sum_{i=1}^N [\tilde{\mathbf{z}}_{2i}]_a [\tilde{\mathbf{z}}_{2i+1}]_b}{\sqrt{\sum_i ([\tilde{\mathbf{z}}_{2i}]_a)^2} \sqrt{\sum_i ([\tilde{\mathbf{z}}_{2i+1}]_b)^2}}$$

5.6.2 Extended Theory and Proofs

In sections 5.3.1 and 5.3.2 we introduce our data generation process and prove that all SSL methods benefit from the alignment term in their objectives. Here we extend the theory to include the output entropy(s) of the encoder network(s) and provide analysis on how SSL prevent representation collapse by maximizing the output entropy of the network.

Theorem 5.6.1. *With a data generation process described in 5.3.1, all discriminative SSL objectives have an alignment loss function between positive pairs from the network and output entropy loss function(s) of the network(s):*

$$\mathcal{L}_{SSL} = \|f(\mathbf{x}) - f(\mathbf{x})\|_2^2 - H(f(\mathbf{x}, \theta)) \tag{5.16}$$

For InfoNCE-driven SSL methods, the proof is shown as (5.6) and (5.7).

$$\lim_{K\to\infty} \mathcal{L}_{InfoNCE} - logK - logC_q(\mathbf{z}) =$$
$$\frac{1}{\tau} \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [\|f(\mathbf{x}) - f(\tilde{\mathbf{x}})\|_2^2/2 - 1] - H(f(\mathbf{x}))$$

For both EMA-driven and Siamese with predictor SSLs, we show that the loss function can be reformulated to three terms that first two are the alignment between positive pairs through the online/trainable network, and the alignment between same data sample from two networks. The third term can be further approximated via second order Taylor expansion around $\mathbf{z}$.

$$\begin{aligned}
&-2 \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [(p'(\tilde{\mathbf{x}},\theta) - p'(\mathbf{x},\theta))^\top (p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi))] \\
&= -2(1 - \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)] - \\
&\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}} [p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi))] + \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi))]) \\
&\approx -2(1 - log(\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [e^{p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)}]) + \frac{\mathbb{V}[p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)]}{2\mathbb{E}[p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)]^2} \\
&-log(\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}} [e^{p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi))}]) + \frac{\mathbb{V}[p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi))]}{2\mathbb{E}[p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi))]^2} \\
&+log(\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [e^{p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi))}]) - \frac{\mathbb{V}[p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi))]}{2\mathbb{E}[p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi))]^2})
\end{aligned} \tag{5.17}$$

Assuming that κ_2 is a large number (as set by $1/\tau$ in SimCLR and MoCo), then the variance terms $\mathbb{V}[.] \approx 0$.

$$
\begin{aligned}
\mathcal{L} &= \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} \|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta) + p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi)\|_2^2 \\
&= \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} \|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta)\|_2^2 + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}} \|p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi)\|_2^2 \\
&\quad -2 \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [(p'(\tilde{\mathbf{x}},\theta) - p'(\mathbf{x},\theta))^\top (p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi))] \\
&\approx \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} \|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta)\|_2^2 + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}} \|p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi)\|_2^2 \\
&\quad -2 + 2log(\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [e^{p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)}]) + 2log(\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}} [e^{p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi))}]) \\
&\quad -2log(\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [e^{p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi))}]) \\
&= \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} \|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta)\|_2^2 + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}} \|p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi)\|_2^2 \\
&\quad -2 - 2H(p'(\mathbf{x})) + 2logC_q(\mathbf{z}) + 2log(\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}} [e^{p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi))}]) \\
&\quad -2log(\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})} [e^{p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi))}])
\end{aligned}
\tag{5.18}
$$

Since $f(\mathbf{x},\theta)$ and $f(\tilde{\mathbf{x}},\xi)$ maps in the same space $\mathbb{R}^{d_2}$, p can be considered as a bijective linear transformation within $\mathbb{R}^{b_2}$. In [GK11] by change of variable: $H(\mathbf{Y}) = H(\mathbf{X}) + \mathbb{E}[log|\mathbf{J}_m|] - H(\mathbf{X}|\mathbf{Y})$ if $\mathbf{Y} = m\mathbf{X}$, where m is a projection matrix mapping $\mathbf{X} \to \mathbf{Y}$ and $\mathbf{J}_m$ is the Jacobian of m, $\frac{\delta m}{\delta \mathbf{x}}$. This relates the last two terms in (5.18) to maximizing the output cross entropy of p' and f w.r.t the same sample, and minimizing the output cross entropy of p' and f w.r.t positive samples. This also hints on the importance of the predictor in BYOL, since removing the p in $p \circ f$, $2log(\mathbb{E}_{(\mathbf{x},\tilde{\mathbf{x}})}[e^{p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)}])$ and $-2log(\mathbb{E}_{(\mathbf{x},\tilde{\mathbf{x}})}[e^{p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi))}])$ will cancel out and omit the output entropy maximization objective resulting in the representation collapse.

The same analysis applies to Siamese with predictor SSL. In case the predictor and/or `stop-gradient` is removed, the output entropy maximization objective will

be no longer available and lead to a trivial solution.

For Barlow Twins, the objective can be regarded as minimizing the information loss between two augmented examples. In Appendix.A in [ZJM+21], the author formulate such relation to the Information Bottleneck Principle:

$$\begin{aligned} \mathcal{IB}_\theta &= I(f_\theta(\mathbf{x}), \mathbf{x}) - \beta I(f_\theta(\mathbf{x}), \tilde{\mathbf{x}}) \\ &= H(f_\theta(\mathbf{x})|\mathbf{x}) + \frac{1-\beta}{\beta} H(f_\theta(\mathbf{x})) \end{aligned} \tag{5.19}$$

The first term in (5.19) is linked to the alignment term in (5.15) when [ZJM+21] assumes that $f(\mathbf{x})$ follows a Gaussian distribution. However, in our representation learning formulation, we assume the conditional distribution of positive pairs follows a vMF distribution (5.3). We can further decompose (5.19):

$$\mathcal{IB}_\theta = \mathcal{L}_{alignment} - H(f_\theta(\mathbf{x})) + \frac{1-\beta}{\beta} H(f_\theta(\mathbf{x})) \tag{5.20}$$

As suggested in [ZJM+21], when $\beta < 1$ the best solution of (5.20) is to set the representation to a constant, i.e. representation collapse. When $\beta > 1$, the last term in (5.20) is the same as maximizing the output entropy of the network.

Once we prove that all SSL objectives contain the alignment term and output entropy maximization term, we demonstrate that the cross entropy between $p(.|\mathbf{z})$ in (5.1) and $q_h(.|\mathbf{z})$ in (5.3) can be formulated with the $\mathcal{L}_{alignmet} - H(f_\theta(\mathbf{x}))$ as illustrated in [ZSS+21].

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{z}\sim p(\mathbf{z})}[H(p(.|\mathbf{z}), q_h(.|\mathbf{z}))] \\
&= -\mathop{\mathbb{E}}_{\mathbf{z}\sim p(\mathbf{z})}[\mathop{\mathbb{E}}_{\tilde{\mathbf{z}}\sim p(\tilde{\mathbf{z}}|\mathbf{z})}[log(q_h(\tilde{\mathbf{z}}|\mathbf{z}))]] \\
&= \kappa_2 \mathop{\mathbb{E}}_{(\mathbf{z},\tilde{\mathbf{z}})}[h(\tilde{\mathbf{z}})^\top h(\mathbf{z})] - \mathop{\mathbb{E}}_{\mathbf{z}}[log(\mathop{\mathbb{E}}_{\tilde{\mathbf{z}}}[e^{h(\tilde{\mathbf{z}})^\top h(\mathbf{z})\kappa_1}])] \\
&= \mathcal{L}_{alignment} - H(h(\mathbf{z}))
\end{aligned}
\tag{5.21}
$$

Since $H(.) \leq 0$, then the alignment loss will be a lower bound for $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[H(p(.|\mathbf{z}), q_h(.|\mathbf{z}))]$.

Finally, we can use **Proposition 1** and **Proposition 2** in [ZSS+21] to prove 5.3.2 and 5.3.4.

5.6.3 Causal3DIdent

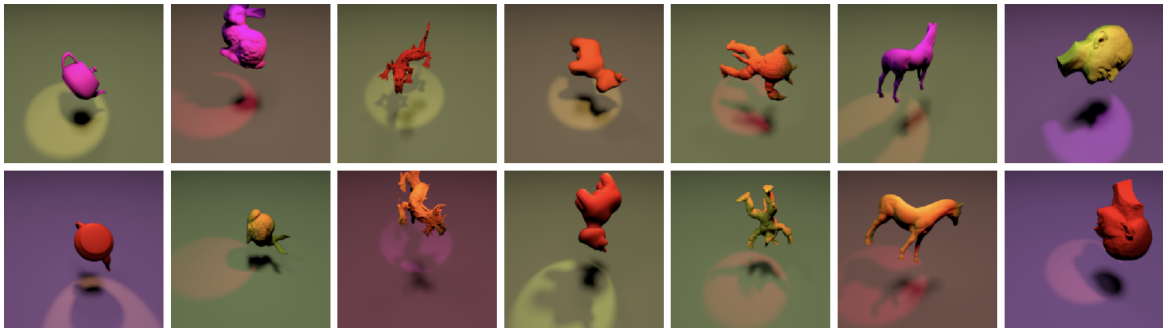


Figure 5.8: Examples of images in Causal3DIdent. Each image is associated with a 10-dimensional ground-truth latent representation. [$pos_{obj} = (x, y, z)$, $rot_{obj} = (\phi, \theta, \psi)$, hue_{obj}, pos_{spl}, hue_{spl}, hue_{bg}]

3DIdent contained 7 object classes: Teapot, Hare, Dragon, Cow, Armadillo, Horse, Head. For spotlight position, spotlight hue, and background hue, variable values are sampled from $U(-1, 1)$. The dependence is imposed by varying the mean (μ) of a truncated normal distribution with standard deviation $\sigma = 0.5$, truncated to the range $[-1, 1]$. See Appendix B in [VBS+21] for the dependency on μ.

To exclude variable range with extreme values, we exclude edges for uniformly sampled variables (≤ -0.8 and ≥ 0.8), and exclude smaller portion tail for dependent variables ($\leq \mu - 0.8$ if $\mu > 0$,$\geq \mu + 0.8$ if $\mu \geq 0$). To ensure the training data size is the same after sampling, the included samples are delicately sampled to match the original data size. Three illustrative examples are shown in Figure 5.9.

This section contains the values of hyperparameters for SSL methods. **Note the batch size is set to 128**.

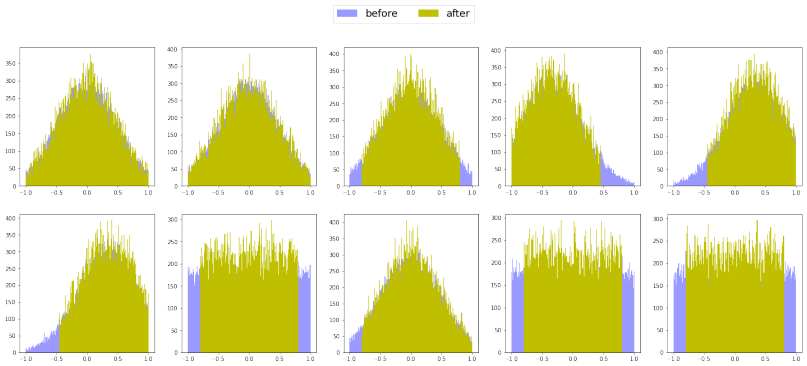
Table 5.4: Hyperparameters for SSL methods in training Causal3dIdent.

	Hyperparameters
SimCLR	$\tau = 0.07$
MoCo	$K = 65536$,$\tau = 0.07$,$\alpha = 0.99$
BYOL	$\alpha = 0.99$
SimSiam	None
Barlow Twins	$\lambda = 0.005$

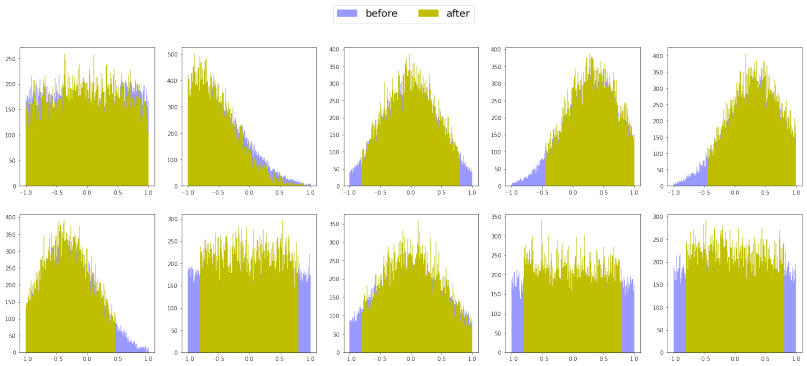
To verify that the difference between seen and unseen distribution does not induce large discrepancy in the performance, we evaluate the accuracy on all classes and report the difference between seen and unseen distributions. See Table 5.5 for results. On average, the difference is accuracy is about 2% to 3%, which is not comparable to the minimum reduction in accuracy $\approx 20\%$ reported in Figure 5.4.

Table 5.5: Accuracy discrepancy between seen and unseen distributions.

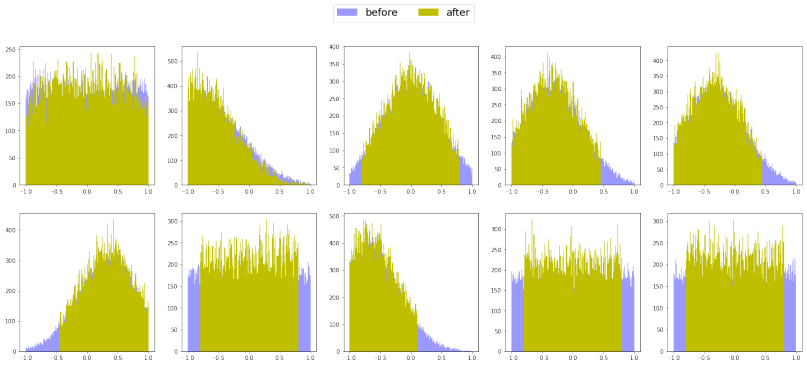
	simclr	moco	byol	simsiam	barlow
acc	0.0343	0.0370	0.0190	0.0168	0.0147
0	0.0860	0.0860	0.0392	0.0279	0.0335
1	0.0761	0.0866	0.0476	0.0320	0.0334
2	0.0241	0.0179	0.0178	0.0097	0.0095
3	0.0445	0.0449	0.0164	0.0261	0.0090
4	0.0543	0.0458	0.0300	0.0226	0.0126
5	0.0603	0.0488	0.0262	0.0234	0.0281
6	0.0542	0.0445	0.0420	0.0655	0.0387



(a) Teapot



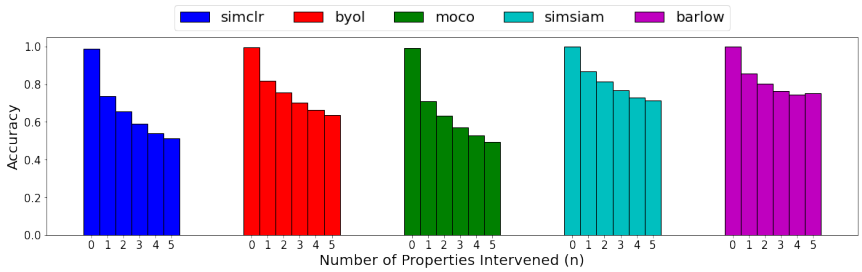
(b) Dragon



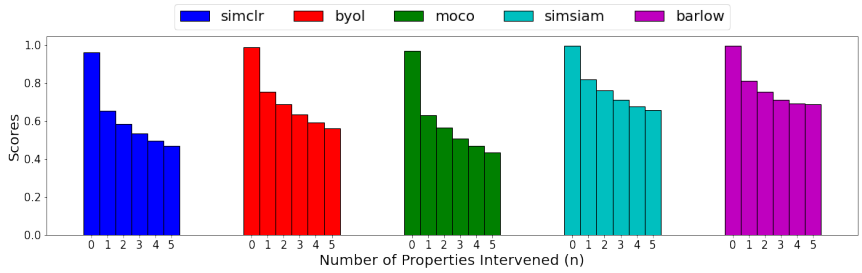
(c) Horse

Figure 5.9: Examples of sampling intervened data samples. Before sampling and After sampling.

We also verify the same deterioration in performance when only intervening the 5 children nodes in Figure 5.3. In Figure 5.10, we observe the same deterioration when only 5 variables are sampled to exclude the extreme edge(s).



(a) Deterioration in Accuracy



(b) Deterioration in Score

Figure 5.10: Deterioration of unstable changing in data variables on all SSL. Only 5 children nodes are intervened.

5.6.4 ImageNet

ObjectNet is a large crowdsourced test set for object recognition that includes controls for object rotations, viewpoints, and backgrounds. Objects are posed by workers in their own homes in natural settings according to specific instructions detailing what object class they should use, how and where they should pose the object, and where to image the scene from. Every image is annotated with these properties, allowing us to test how well object detectors work across these conditions. Each of these properties is randomly sampled leading to a much more varied dataset. There are 313 ObjectNet classes with 113 of them overlapping with the ImageNet

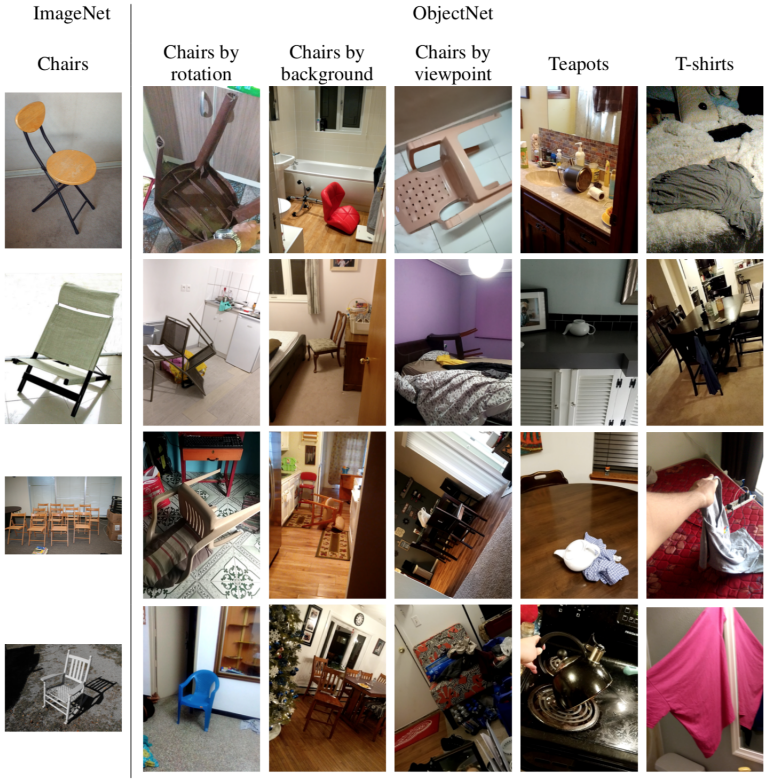


Figure 5.11: Illustration of ObjectNet controlling data variables.

Figure 5.12: Examples of Stylized ImageNet

classes. With each controlled variable, the changes in the variable poses challenges to identify the objects correctly due to the very unusual shift.

Starting from ImageNet **Stylized ImageNet** is constructed by stripping every single image of its original texture and replacing it with the style of a randomly selected painting through AdaIN style transfer. The original objective for Stylized-ImageNet is to help the network to learn more about shapes, and less about local textures. However, we regards this shift in the appearance as a change in the data variables(s). Since the stylized images appear drastically different to natural images, we assume this shift is very hard to counter in the representation space due to entangled transformations.

Synthetic Data follows synthetic procedure in [DYT+20] where object is masked on a background at a location with a rotation angle. Foreground object masks are cropped from OpenImages [KRA+20]. The object classes that overlap with ObjectNet classes are selected, and each class is selected with 10 object masks at highest area and not truncated by any other objects. The backgrounds are sampled from *pexel.com* with the same set of 867 backgrounds used in [DYT+20]. Initially there are three data variables to control with: *background*, *rotation*, **location**. The object class is randomly selected at each sythesizing step. At each training step, the other two variables are randomly fixed while the target variable is randomly selected across 10 values. This results in 10 images to compare with each other. For the special data

variable, *texture*, we change the style of the selected object masks to different textures in *Describable Textures Dataset* [CMK+14] based on formulation used in **Stylized ImageNet**. With all other three variables randomly fixed, the object mask with 10 different textures are sampled.

We carry the experiment on **Solution 2** to 50 epochs and observe the saturation of the improvement (Figure 5.13). To further improve the performance, more integrated interventions should be applied to make **F** more robust to shifts in the data variables.

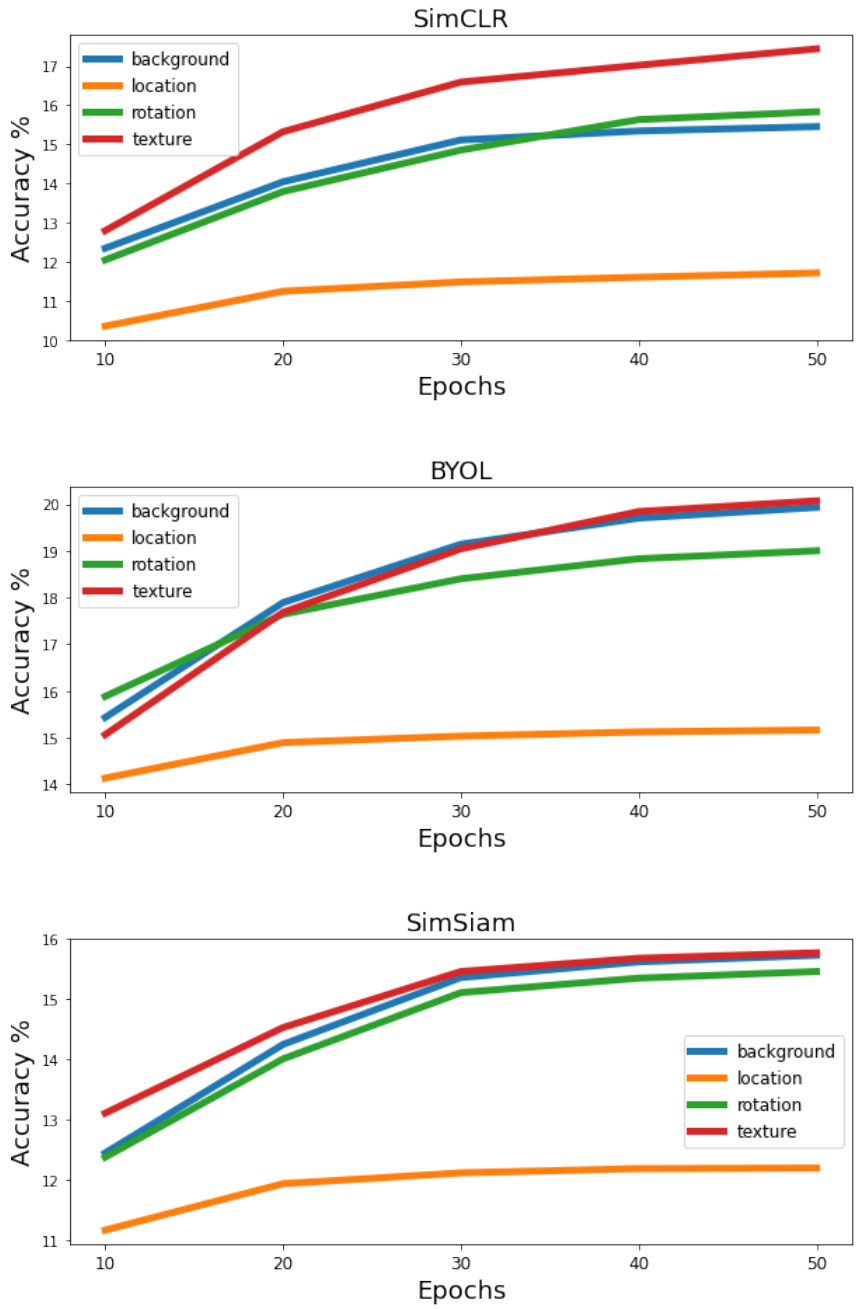


Figure 5.13: Training longer in **Solution 2** can improve the performance. But the improvement saturates after around 30 epochs and the improvement becomes less significant.

Chapter 6

Conclusions

Through research presented in this thesis, I demonstrate the connections between causality and deep vision models. Causality plays a crucial role in understanding how deep vision models learn representations, and this Ph.D. dissertation has formulated causal frameworks and discussed how to learn and interpret the representations with labeled, partially labeled, and unlabeled data. The proposed strategies in each chapter have demonstrated improvements in model interpretability, robustness, and generalization, contributing to the ongoing efforts to enhance visual representation learning. Specifically, I summarize the findings of this thesis as following:

- In Chapter 2, we focus on interpreting the prediction of a supervised deep vision model with causal pixels in the input image. We recognize the weakness of gradient-based saliency maps that tend to spurious correlations when the model struggles to improve the confidence w.r.t a specific target label. By 'backpopagting' the weight of the model instead of the gradient, causal pixels with highest contributions to the prediction score can be more accurately identified. In addition to explaining the prediction, we also propose causally intervene the image samples with the weight-based saliency maps so that positive and negative counterfactuals are generated to improve the prediction. By comparing with gradient-based saliency maps, the proposed methods learn more generalized representations on medical images.

- In Chapter 3, we weigh the significance of the quality and quantity of the acquired data in the supervised object detection application. We perceive the challenge

in collecting manual labels for objects in all public object detection datasets and notice many objects are missing labels. These missing labels oppose the learning signal during the training and hence result in worse performance. By reforming a Positive-Unlabeled (PU) loss function instead of a Positive-Negative one, we can alliviate the opposing signals from these 'unlabeled' instances via inferring a prior on the positive instances. In this particular case, we intervene the effect of unlabeled instances with a statistical prior that is estimated dynamically.

- Self-supervised learning is a type of unsupervised learning where the model is trained to predict or reconstruct certain features or inputs of the data itself, without the need for human-labeled data. In Chapter 4, we study a more difficult representation learning problem with unlabeled data in a decentralized setting. Prior chapters demonstrated that deep vision models is able to learn generalized representations from centralized data. But for specificity of the applications, strong influencing factors like labels guide the downstream models to generate specific output. Federated learning learns a generalized global model and desires personalized local models. We address both generalization and personalization on basis of causal understanding of self-supervised learning. By extracting local style representations and infusing them in the latent space during FL training, a more generalized global model can be aggregated since the model is not only robust to distortions in image space, but also robust to interventions in the latent space. With the extracted style representations, more customized local representation improves the personalization of local models.

- In Chapter 5, we analyse the unstable behaviour of the representation learned via descriminative SSL methods under a carefully constructed causal framework. We observe the deterioration on the downstream performance from some subtle

changes in the data variable. By analytically understand the causal factors of the successful alignment and entropy regularization in all descriminative SSL objectives, we identify the root reasons for these unstable behaviour. Novel solutions are proposed to counter the shift caused by the change in data variables and the effectiveness are verified on both controlled image dataset and natural image dataset.

Overall, this Ph.D. dissertation contributes to the ongoing research in visual representation learning, specifically in enhancing the interpretability, robustness, and generalization of deep vision models. The proposed strategies and frameworks can potentially be applied to other computer vision tasks, ultimately leading to improved performance and accuracy.

Bibliography

[AASC19] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[ABGLP19] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[AFS03] FirstName Alpher and FirstName Fotheringham-Smythe. Frobnication revisited. *Journal of Foo*, 13(1):234–778, 2003.

[AFSG04] FirstName Alpher, FirstName Fotheringham-Smythe, and FirstName Gamow. Can a machine frobnicate? *Journal of Foo*, 14(1):234–778, 2004.

[AGM+18] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018.

[AHB+18] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *Conference on Computer Vision and Pattern Recognition*, 2018.

[AIMB+11] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.

[AL76] Ibrahim Ahmad and Pi-Erh Lin. A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.). *IEEE Transactions on Information Theory*, 22(3):372–375, 1976.

[ALA+15] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. VQA: Visual Question Answering. *International Conference on Computer Vision*, 2015.

[Alp02] FirstName Alpher. Frobnication. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):234–778, 2002.

[BBM+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[BDBCP06] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

[BFA20] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020.

[BHB19] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

[BHS19] Sergiy V Borodachov, Douglas P Hardin, and Edward B Saff. *Discrete energy on rectifiable sets*, volume 3. Springer, 2019.

[BMA+19] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019.

[BSEvLC+17] David S Boyer, Ursula Schmidt-Erfurth, Menno van Lookeren Campagne, Erin C Henry, and Christopher Brittain. The pathophysiology of geographic atrophy secondary to age-related macular degeneration and the complement pathway as a therapeutic target. *Retina (Philadelphia, Pa.)*, 37(5):819, 2017.

[BV16] Hakan Bilen and Andrea Vedaldi. Weakly Supervised Deep Detection Networks. *Conference on Computer Vision and Pattern Recognition*, 2016.

[CCGD18] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations*, 2018.

[CFGH20] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[CH21a] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[CH21b] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[CHMS21] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021.

[CK07] Henry Cohn and Abhinav Kumar. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, 20(1):99–148, 2007.

[CKNH20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[CLW18] Yue Cao, Mingsheng Long, and Jianmin Wang. Unsupervised domain adaptation with distribution matching machines. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[CMK+14] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[CMM+20] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[CRC+20] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

[CSHB18] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based

visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.

[CSKX15] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730, 2015.

[CTM+21a] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[CTM+21b] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. *Proceedings of the IEEE International Conference on Computer Vision*, pages 9630–9640, 2021.

[CXH21] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[DCDGL99a] Francesco De Comité, François Denis, Rémi Gilleron, and Fabien Letouzey. Positive and Unlabeled Examples Help Learning. *Algorithmic Learning Theory*, 1999.

[DCDGL99b] Francesco De Comité, François Denis, Rémi Gilleron, and Fabien Letouzey. Positive and unlabeled examples help learning. In *International Conference on Algorithmic Learning Theory*, pages 219–230. Springer, 1999.

[DCL+18] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, pages 592–603, 2018.

[DDCP88] Elizabeth R DeLong, David M DeLong, and Daniel L Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, pages 837–845, 1988.

[DDS+09a] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE*

conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[DDS+09b] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. *Conference on Computer Vision and Pattern Recognition*, 2009.

[Den98] François Denis. PAC Learning from Positive Statistical Queries. *Algorithmic Learning Theory*, 1998.

[Den12] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[DG17] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pages 6967–6976, 2017.

[DI09] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[DKB14] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[DKM20] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

[DLHS16] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *Advances In Neural Information Processing Systems*, 2016.

[DLSH18] Mengnan Du, Ninghao Liu, Qingquan Song, and Xia Hu. Towards explanation of dnn-based prediction with guided feature inversion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1358–1367, 2018.

[DPNS14] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *Advances in neural information processing systems*, pages 703–711, 2014.

[DPNS15] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *International Conference on Machine Learning*, pages 1386–1394, 2015.

[DS14] Marthinus Christoffel Du Plessis and Masashi Sugiyama. Class prior estimation from positive and unlabeled data. *IEICE Transactions on Information and Systems*, pages 1358–1362, 2014.

[DS19] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *Advances in neural information processing systems*, 32, 2019.

[DVT+21] Canh T Dinh, Tung T Vu, Nguyen H Tran, Minh N Dao, and Hongyu Zhang. Fedu: A unified framework for federated multi-task learning with laplacian regularization. *arXiv preprint arXiv:2102.07148*, 2021.

[DXT12] Lixin Duan, Dong Xu, and Ivor Tsang. Learning with augmented features for heterogeneous domain adaptation. *arXiv preprint arXiv:1206.4660*, 2012.

[DYT+20] Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, Sylvain Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On robustness and transferability of convolutional neural networks, 2020.

[DZM+18] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-shot Object Detection. *Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[EBCV09] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[EHL19] Stavros Efthymiou, Jack Hidary, and Stefan Leichenauer. Tensornetwork for machine learning. *arXiv preprint arXiv:1906.06329*, 2019.

[EHRP04] Deniz Erdogmus, Kenneth E Hild, Yadunandana N Rao, and Jose C Principe. Minimax mutual information approach for independent component analysis. *Neural Computation*, 16(6):1235–1252, 2004.

[EN08] Charles Elkan and Keith Noto. Learning Classifiers from Only Positive and Unlabeled Data. *International Conference on Knowledge Discovery and Data Mining*, 2008.

[EVW+10] Mark Everingham, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 2010.

[FA20] Abe Fetterman and Josh Albrecht. Understanding self-supervised and contrastive learning with bootstrap your own latent (byol). *Untitled AI, August*, 2020.

[FHYF19] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10705–10714, 2019.

[Fis53] Ronald Aylmer Fisher. Dispersion on a sphere. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 217(1130):295–305, 1953.

[FMO20] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[FPV19] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2950–2958, 2019.

[FV17] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.

[GDDM14a] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[GDDM14b] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Conference on Computer Vision and Pattern Recognition*, 2014.

[GH10] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[GHK+20] Mathilde Guillemot, Catherine Heusele, Rodolphe Korichi, Sylvianne Schnebert, and Liming Chen. Breaking batch normalization for better explainability of deep neural networks through layer-wise relevance propagation. *arXiv preprint arXiv:2002.11018*, 2020.

[Gir15] Ross Girshick. Fast R-CNN. *International Conference on Computer Vision*, 2015.

[GK11] Bernhard C. Geiger and Gernot Kubin. On the information loss in memoryless systems: The multivariate case. *CoRR*, abs/1109.4856, 2011.

[GMW+19] Robert Geirhos, Claudio Michaelis, Felix A. Wichmann, Patricia Rubisch, Matthias Bethge, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *7th International Conference on Learning Representations, ICLR 2019*, pages 1–22, 2019.

[GMX+13] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. *arXiv preprint*, 2013.

[GPAM+14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *Advances In Neural Information Processing Systems*, 2014.

[GRM+19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[GSA+20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[GSK18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[GSKZ17] AmirEmad Ghassami, Saber Salehkaleybar, Negar Kiyavash, and Kun Zhang. Learning causal structures using regression invariance. In *Advances in Neural Information Processing Systems*, pages 3011–3021, 2017.

[GSR+18] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici,

Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions. *Conference on Computer Vision and Pattern Recognition*, 2018.

[GWE+19] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *ICML*, 2019.

[Hay94] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[HB17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[HCH+15] Tzu Ming Harry Hsu, Wei Yu Chen, Cheng-An Hou, Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Unsupervised domain adaptation with imbalanced cross-domain data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4121–4129, 2015.

[HCX+22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[HEKK19] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9737–9748, 2019.

[HFLM+18] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[HFW+20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *International Conference on Computer Vision*, 2017.

[HGT+14] Judy Hoffman, Sergio Guadarrama, Eric Tzeng, Jeff Donahue, Ross B. Girshick, Trevor Darrell, and Kate Saenko. LSDA: Large Scale Detection Through Adaptation. *Advances In Neural Information Processing Systems*, 2014.

[HH00] Aapo Hyvärinen and Patrik Hoyer. Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural computation*, 12(7):1705–1720, 2000.

[HHH+15] Kai-Lung Hua, Che-Hao Hsu, Shintami Chusnul Hidayati, Wen-Huang Cheng, and Yu-Jen Chen. Computer-aided classification of lung nodules on computed tomography images via deep learning technique. *OncoTargets and therapy*, 8, 2015.

[HLVDMW17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[HM16] Aapo Hyvärinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and Nonlinear ICA. *Advances in Neural Information Processing Systems*, pages 3772–3780, 2016.

[HM17] Aapo Hyvärinen and Hiroshi Morioka. Nonlinear ICA of temporally dependent stationary sources. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, 2017.

[Hol86] Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960, 1986.

[HP99] Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

[HPW+22] Sungwon Han, Sungwon Park, Fangzhao Wu, Sundong Kim, Chuhan Wu, Xing Xie, and Meeyoung Cha. Fedx: Unsupervised federated learning with cross knowledge distillation. In *European Conference on Computer Vision*, pages 691–707. Springer, 2022.

[HRS+17] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *Conference on Computer Vision and Pattern Recognition*, 2017.

[HST20] Aapo Hyvärinen, Hiroaki Sasaki, and Richard E. Turner. Nonlinear ICA using auxiliary variables and generalized contrastive learning. *AISTATS 2019 - 22nd International Conference on Artificial Intelligence and Statistics*, 89, 2020.

[Hul94] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[HWGM21] Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable Guarantees for Self-Supervised Deep Learning with Spectral Contrastive Loss. *Advances in Neural Information Processing Systems*, 7(NeurIPS):5000–5011, 2021.

[HYM+21] Chaoyang He, Zhengyu Yang, Erum Mushtaq, Sunwoo Lee, Mahdi Soltanolkotabi, and Salman Avestimehr. Ssfl: Tackling label deficiency in federated learning via personalized self-supervision. *arXiv preprint arXiv:2110.02470*, 2021.

[HZ93] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6, 1993.

[HZRS16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[HZRS16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *Conference on Computer Vision and Pattern Recognition*, 2016.

[IZJ+17] Haroon Idrees, Amir R. Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 2017.

[JBZ+20] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[JSS16] Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029, 2016.

[JT20] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

[JWR16] Shantanu Jain, Martha White, and Predrag Radivojac. Estimating the class prior and posterior from noisy positives and unlabeled data. In *Advances in neural information processing systems*, pages 2693–2701, 2016.

[JZL+18] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. *International Conference on Machine Learning*, 2018.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KBT19] Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. Adaptive gradient-based meta-learning methods. *Advances in Neural Information Processing Systems*, 32, 2019.

[KC10] Christopher Kanan and Garrison Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2472–2479. IEEE, 2010.

[KDA+16] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. OpenImages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2016.

[KGC+18] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[KJYH19] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation.

In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4893–4902, 2019.

[KKMH20] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

[KNDS17] Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. Positive-Unlabeled Learning with Non-Negative Risk Estimator. *Advances In Neural Information Processing Systems*, 2017.

[KPR+17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 2017.

[KRA+18] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint*, 2018.

[KRA+20] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[KRMW14] Diederik P Kingma, Danilo J Rezende, Shakir Mohamed, and Max Welling. Semi-supervised Learning with Deep Generative Models. *Advances In Neural Information Processing Systems*, 2014.

[KRPM+18] Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus Maier-Hein, SM Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. In *Advances in Neural Information Processing Systems*, pages 6965–6975, 2018.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 2012.

[KSH17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[KSS+20] David Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

[KTW+20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[KVB88] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.

[KZG+17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal Computer Vision*, 2017.

[LAE+16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot MultiBox Detector. *European Conference on Computer Vision*, 2016.

[LAE+19] Curtis P Langlotz, Bibb Allen, Bradley J Erickson, Jayashree Kalpathy-Cramer, Keith Bigelow, Tessa S Cook, Adam E Flanders, Matthew P Lungren, David S Mendelson, Jeffrey D Rudie, et al. A roadmap for foundational research on artificial intelligence in medical imaging: from the 2018 nih/rsna/acr/the academy workshop. *Radiology*, 291(3):781–791, 2019.

[Las14a] FirstName LastName. The frobnicatable foo filter, 2014. Face and Gesture submission ID 324. Supplied as additional material `fg324.pdf`.

[Las14b] FirstName LastName. Frobnication tutorial, 2014. Supplied as additional material `tr.pdf`.

[LBD+89] Y. Lecun, B. Boser, J.S Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1989.

[LDG00] Fabien Letouzey, François Denis, and Rémi Gilleron. Learning from Positive and Unlabeled Examples. *Algorithmic Learning Theory*, 2000.

[LDG+17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. *Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[LDXT13] Wen Li, Lixin Duan, Dong Xu, and Ivor W Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1134–1148, 2013.

[LGG+17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. *International Conference on Computer Vision*, 2017.

[LGH+17] Rebecca Sawyer Lee, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy, and Daniel L. Rubin. A curated mammography data set for use in computer-aided detection and diagnosis research. *Nature Scientific Data*, 2017.

[LHBS21] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[LHG+18] Kevin J Liang, Geert Heilmann, Christopher Gregory, Souleymane O. Diallo, David Carlson, Gregory P. Spell, John B. Sigman, Kris Roe, and Lawrence Carin. Automatic Threat Recognition of Prohibited Items at Aviation Checkpoint with X-ray Imaging: A Deep Learning Approach. *SPIE Anomaly Detection and Imaging with X-Rays (ADIX) III*, 2018.

[LHL+20] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. Fedvision: An online visual object detection platform powered by federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13172–13179, 2020.

[Lin88] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[LJP+22] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and Philip Yu. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[LKHS20] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *Ieee Access*, 8:193907–193934, 2020.

[LL03] Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, volume 3, pages 448–455, 2003.

[LLSZ21] Jason D. Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting What You Already Know Helps: Provable Self-Supervised Learning. *Advances in Neural Information Processing Systems*, 1(NeurIPS):309–323, 2021.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C Lawrence Zitnick, and Piotr Dolí. Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision*, 2014.

[LMRPC+22] Shuo Liu, Adria Mallol-Ragolta, Emilia Parada-Cabaleiro, Kun Qian, Xin Jing, Alexander Kathan, Bin Hu, and Bjoern W Schuller. Audio self-supervised learning: A survey. *Patterns*, 3(12):100616, 2022.

[LMX+19] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, et al. Privacy-preserving federated brain tumour segmentation. In *International workshop on machine learning in medical imaging*, pages 133–141. Springer, 2019.

[LRS+19] Afshan Latif, Aqsa Rasheed, Umer Sajid, Jameel Ahmed, Nouman Ali, Naeem Iqbal Ratyal, Bushra Zafar, Saadat Hanif Dar, Muhammad Sajid, and Tehmina Khalil. Content-based image retrieval and feature extraction: a comprehensive review. *Mathematical problems in engineering*, 2019, 2019.

[LSD15a] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *Conference on Computer Vision and Pattern Recognition*, 2015.

[LSD15b] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[LSW+13] Jessica N Leuschen, Stefanie G Schuman, Katrina P Winter, Michelle N McCall, Wai T Wong, Emily Y Chew, Thomas Hwang, Sunil Srivastava, Neeru Sarin, Traci Clemons, et al. Spectral-domain optical

coherence tomography characteristics of intermediate age-related macular degeneration. *Ophthalmology*, 120(1):140–150, 2013.

[LSZ+20] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[LWD+14] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1410–1417, 2014.

[LWP+18] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9215–9223, 2018.

[LYZ+21] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021.

[LZH+21] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2021.

[LZYN11] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR 2011*, pages 3361–3368. IEEE, 2011.

[MAD+12] Inês C. Moreira, Igor Amaral, Inês Domingues, Ant´ónio Cardoso, Maria Jo˜ão Cardoso, and Jaime S. Cardoso. INbreast: Toward a Full-field Digital Mammographic Database. *Academic Radiology*, 2012.

[MBL+19] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209. Springer, 2019.

[MC89] Michael McCloskey and Neal J Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *The Psychology of Learning and Motivation*, 1989.

[MIM20] Mostafa Neo Mohsenvand, Mohammad Rasool Izadi, and Pattie Maes. Contrastive representation learning for electroencephalogram classification. In *Machine Learning for Health*, pages 238–253. PMLR, 2020.

[MLB+17] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

[MMKI18] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[MMR+17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[MMW+20] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*, 2020.

[MNB+21] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 34:15434–15447, 2021.

[Mon19] Grégoire Montavon. Gradient-based vs. propagation-based explanations: an axiomatic comparison. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 253–265. Springer, 2019.

[MSM18] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[MV16] Aravindh Mahendran and Andrea Vedaldi. Salient deconvolutional networks. In *European Conference on Computer Vision*, pages 120–135. Springer, 2016.

[NF16] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, pages 69–84. Springer, 2016.

[NLBT17] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational Continual Learning. *International Conference on Learning Representations*, 2017.

[NV06] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. *International Conference on Pattern Recognition*, 2006.

[NWC+11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[OBLS15] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? Weakly-supervised learning with convolutional neural networks. *Conference on Computer Vision and Pattern Recognition*, 2015.

[OCC+20] Matthew O'Shaughnessy, Gregory Canal, Marissa Connor, Christopher Rozell, and Mark Davenport. Generative causal explanations of black-box classifiers. *Advances in Neural Information Processing Systems*, 33, 2020.

[OLV18a] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[OLV18b] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[OOR+18] Avital Oliver, Augustus Odena, Colin A. Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. *Advances In Neural Information Processing Systems*, 2018.

[PDS18] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[Pea09a] Judea Pearl. *Causality*. Cambridge university press, 2009.

[Pea09b] Judea Pearl. *Causality*. Cambridge university press, 2009.

[PGH+16] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational Autoencoder for Deep Learning of Images, Labels and Captions. *Advances In Neural Information Processing Systems*, 2016.

[PKD+16] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[Rat90] Roger Ratcliff. Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions. *Psychology Review*, 1990.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *Conference on Computer Vision and Pattern Recognition*, 2016.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[RF17] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger. *Conference on Computer Vision and Pattern Recognition*, 2017.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer Assisted Intervention*, 2015.

[RFJV20] Sylvestre-Alvise Rebuffi, Ruth Fong, Xu Ji, and Andrea Vedaldi. There and back again: Revisiting backpropagation saliency methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8848, 2020.

[RGBYO18] Alexandre Rame, Emilien Garreau, Hedi Ben-Younes, and Charles Ollion. OMNIA Faster R-CNN: Detection in the wild through dataset merging and soft distillation. *arXiv preprint*, 2018.

[RHDV17] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2662–2670, 2017.

[RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, 2015.

[RHU+18] Dezs o Ribli, Anna Horváth, Zsuzsa Unger, Péter Pollner, and István Csabai. Detecting and classifying lesions in mammograms with Deep Learning. *Nature Scientific Reports*, 2018.

[RIZ+17] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[RNZ18] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. In *Classification in BioApps*, pages 323–350. Springer, 2018.

[RS21] Tobias Ringwald and Rainer Stiefelhagen. Adaptiope: A modern benchmark for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 101–110, January 2021.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[RSMY20] Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International Conference on Machine Learning*, pages 8116–8126. PMLR, 2020.

[RW15] Mrigank Rochan and Yang Wang. Weakly supervised localization of novel objects using appearance transfer. *Conference on Computer Vision and Pattern Recognition*, 2015.

[Saj06] Paul Sajda. Machine learning for detection and diagnosis of disease. *Annu. Rev. Biomed. Eng.*, 8:537–565, 2006.

[SBSL18] Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual Learning in Generative Adversarial Nets. *arXiv preprint*, 2018.

[SCD+17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[Sch19] Bernhard Schölkopf. Causality for machine learning. *arXiv preprint arXiv:1911.10500*, 2019.

[SD20] Raghavendra Selvan and Erik B Dam. Tensor networks for medical image classification. In *Medical Imaging with Deep Learning*, 2020.

[SD22] Saeed Shurrab and Rehab Duwairi. Self-supervised learning methods and applications in medical imaging analysis: A survey. *PeerJ Computer Science*, 8:e1045, 2022.

[SGBB20] Mark P Sendak, Michael Gao, Nathan Brajer, and Suresh Balu. Presenting machine learning model information to clinical end users with model facts labels. *NPJ Digital Medicine*, 3(1):1–4, 2020.

[SGG16] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training Region-based Object Detectors with Online Hard Example Mining. *Conference on Computer Vision and Pattern Recognition*, 2016.

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153, 2017.

[SGZ21] Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive learning of general-purpose audio representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3875–3879. IEEE, 2021.

[SLC+18] Jonathan Schwarz, Jelena Luketina, Wojciech M Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & Compress: A scalable framework for continual learning. *International Conference on Machine Learning*, 2018.

[SMKLM15] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[SMS20] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

[SMSB19] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, pages 6056–6065. PMLR, 2019.

[SOO19] Dasom Seo, Kanghan Oh, and Il-Seok Oh. Regional multi-scale approach for visually pleasing explanations of deep neural networks. *IEEE Access*, 8:8572–8582, 2019.

[SRM+16] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder Variational Autoencoders. *Advances in Neural Information Processing Systems*, 2016.

[SRS22] Madeline C Schiappa, Yogesh S Rawat, and Mubarak Shah. Self-supervised learning for videos: A survey. *ACM Computing Surveys*, 2022.

[SSL20] Amitojdeep Singh, Sourya Sengupta, and Vasudevan Lakshminarayanan. Explainable deep learning models in medical image analysis. *Journal of Imaging*, 2020.

[SSLC20] John B. Sigman, Gregory P. Spell, Kevin J Liang, and Lawrence Carin. Background Adaptive Faster R-CNN for Semi-supervised Convolutional Object Detection of Threats in X-ray Images. *SPIE Anomaly Detection and Imaging with X-Rays (ADIX) V*, 2020.

[SSSG17] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. *International Conference on Computer Vision*, 2017.

[STK+17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[SVI+16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[SVZ13] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[SZS12] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. T. *Technical Report CRCV-TR-12-01, University of Central Florida*, 2012.

[TCG21] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

[TDR+19] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

[TKI19] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multi-view coding. *arXiv preprint arXiv:1906.05849*, 2019.

[TL21] Mamatha Thota and Georgios Leontidis. Contrastive domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2209–2218, 2021.

[TNW06] Yee Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. *Advances in neural information processing systems*, 19, 2006.

[TSdC+19] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An Empirical Study of Example Forgetting during Deep Neural Network Learning. *International Conference on Learning Representations*, 2019.

[TV17] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, 2017.

[TWG+16] Yuxing Tang, Josiah Wang, Boyang Gao, Emmanuel Dellandrea, Robert Gaizauskas, and Liming Chen. Large Scale Semi-Supervised Object Detection Using Visual and Semantic Knowledge Transfer. *Conference on Computer Vision and Pattern Recognition*, 2016.

[VAC+17] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning From Noisy Large-Scale Datasets With Minimal Supervision. *Conference on Computer Vision and Pattern Recognition*, 2017.

[VBS+21] Kügelgen Von, Julius and, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-Supervised Learning with Data Augmentations Provably Isolates Content from Style Luigi Gresele. *Yash Sharma*, 3(NeurIPS):4, 2021.

[VdMH08] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[VECP17] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

[VKSG+21] Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

[VLL+10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[VMB+20] Sebastian Vollmer, Bilal A Mateen, Gergo Bohner, Franz J Király, Rayid Ghani, Pall Jonsson, Sarah Cumbers, Adrian Jonas, Katherine SL McAllister, Puja Myles, et al. Machine learning and artificial intelligence research for patient benefit: 20 critical questions on transparency, replicability, ethics, and effectiveness. *bmj*, 368, 2020.

[VMSM18] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5495–5504, 2018.

[WBS+19] Zhe Wu, Navaneeth Bodla, Bharat Singh, Mahyar Najibi, Rama Chellappa, and Larry S. Davis. Soft Sampling for Robust Object Detection. *British Machine Vision Conference*, 2019.

[WBW+11] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[WD18] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[WFX+22] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022.

[WHLX19] Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. *InInternational Conference on Learning Representations*, 2019.

[WHZS20] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense representation learning via causal inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 378–379, 2020.

[WI20] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *37th International Conference on Machine Learning, ICML 2020*, PartF168147-13:9871–9881, 2020.

[WKG+19] Jorg Wagner, Jan Mathias Kohler, Tobias Gindele, Leon Hetzel, Jakob Thaddaus Wiedemer, and Sven Behnke. Interpretable and fine-grained visual explanations for convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9097–9107, 2019.

[WKH+20] Martin J Willemink, Wojciech A Koszek, Cailin Hardell, Jie Wu, Dominik Fleischmann, Hugh Harvey, Les R Folio, Ronald M Summers, Daniel L Rubin, and Matthew P Lungren. Preparing medical imaging data for machine learning. *Radiology*, 295(1):4–15, 2020.

[WPL+17] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. ChestX-ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. *Conference on Computer Vision and Pattern Recognition*, 2017.

[WWD+20] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–25, 2020.

[WWK+19] Lezi Wang, Ziyan Wu, Srikrishna Karanam, Kuan-Chuan Peng, Rajat Vikram Singh, Bo Liu, and Dimitris N Metaxas. Sharpen focus: Learning with attention separability and consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 512–521, 2019.

[WWW+22] Rui Wang, Zuxuan Wu, Zejia Weng, Jingjing Chen, Guo-Jun Qi, and Yu-Gang Jiang. Cross-domain contrastive learning for unsupervised domain adaptation. *IEEE Transactions on Multimedia*, 2022.

[WWZ+21] Yawen Wu, Zhepeng Wang, Dewen Zeng, Meng Li, Yiyu Shi, and Jingtong Hu. Federated contrastive representation learning with feature fusion and neighborhood matching. 2021.

[XBG+19] Mengmeng Xu, Yancheng Bai, Bernard Ghanem, Boxiao Liu, Yan Gao, N Guo, X Ye, F Wan, H You, D Fan, et al. Missing labels in object detection. In *CVPR workshops*, volume 3, page 5, 2019.

[XKTS21] Detai Xin, Tatsuya Komatsu, Shinnosuke Takamichi, and Hiroshi Saruwatari. Disentangled speaker and language representations using mutual information minimization and domain adaptation for cross-lingual tts. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6608–6612. IEEE, 2021.

[XLY+21] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021.

[XXXT17] Yixing Xu, Chang Xu, Chao Xu, and Dacheng Tao. Multi-positive and unlabeled learning. In *IJCAI*, pages 3182–3188, 2017.

[YCBL14] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances In Neural Information Processing Systems*, 2014.

[YLW09] Junsong Yuan, Zicheng Liu, and Ying Wu. Discriminative subvolume search for efficient action detection. *Conference on Computer Vision and Pattern Recognition*, 2009.

[YWLS17] Ke Yan, Xiaosong Wang, Le Lu, and Ronald M. Summers. DeepLesion: Automated Deep Mining, Categorization and Detection of Significant Radiology Image Findings using Large-Scale Clinical Lesion Annotations. *arXiv preprint*, 2017.

[ZBB+17] Chiyuan Zhang, Samy Bengio, Google Brain, Moritz Hardt, Benjamin Recht, Oriol Vinyals, and Google Deepmind. Understanding Deep Learning Requires Re-thinking Generalization. *International Conference on Learning Representations*, 2017.

[ZCS+23] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023.

[ZF14] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[ZGW+21] Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4912–4921, 2021.

[ZIE16] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016.

[ZJM+21] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[ZKL+16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[ZKY+20] Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li. Federated unsupervised representation learning. *arXiv preprint arXiv:2010.08982*, 2020.

[ZPG17] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence. *International Conference on Machine Learning*, 2017.

[ZSS+21] Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pages 12979–12990. PMLR, 2021.

[ZZL20] Cheng Zhang, Kun Zhang, and Yingzhen Li. A causal view on robustness of neural networks. *arXiv preprint arXiv:2005.01095*, 2020.

[ZZZ+18] Yitian Zhao, Yalin Zheng, Yifan Zhao, Yonghuai Liu, Zhili Chen, Peng Liu, and Jiang Liu. Uniqueness-driven saliency analysis for automated lesion detection with applications to retinal diseases. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 109–118. Springer, 2018.

Biography

Yuewei Yang was born in Chengdu, Sichuan Province, China. He received the B.A. degree and M.Eng. degree in Electrical and Elctronics Engineering from University of Cambridge, United Kingdom in 2015. In August 2018, he obtained the M.S degree in Electrical and Computer Engineering at Duke University, Durham, NC, where he was offered to further pursue a doctoral degree, under the guidance of Professor Yiran Chen. His research interests focus on causal interpretation on deep vision models and unsupervised visual representation learning.