
Causal View of Generalization



Zhongqi Yue

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

21/07/2023

Date

$$\frac{h}{\lambda}$$

Zhongqi Yue

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

21/07/2023

.....

Date

U NTU NTU NTU NTU NTU |
TU NTU NTU NTU NTU NTU
U NTU NTU NTU NTU NTU
U NTU NTU NTU NTU NTU |
.....

Prof Hanwang Zhang

Authorship Attribution Statement

This thesis contains material from 3 papers published in the following peer-reviewed conferences and 3 papers that are submitted to a conference, in which I am listed as the first author for all 6 papers.

The first part of Chapter 3 is published as [Zhongqi Yue, Hanwang Zhang, Qianru Sun, Xian-Sheng Hua. Interventional Few-Shot Learning. 2020. *Conference on Neural Information Processing Systems \(NeurIPS\)*.](#)

The contributions of the co-authors are as follows:

- Prof. Hanwang Zhang pointed out the research direction and participated in the paper writing.
- Qianru Sun, Xian-Sheng Hua joined the early discussion and polished the paper before submission.
- I designed the algorithm, took charge of the entire code implementation and wrote the draft paper for other co-authors to refine.

The second part of Chapter 3 is published as [Zhongqi Yue, Qianru Sun, Xian-Sheng Hua, Hanwang Zhang. Transporting Causal Mechanisms for Unsupervised Domain Adaptation. 2021. *International Conference on Computer Vision \(ICCV\)*. \(Oral Presentation 210/6236\).](#)

The contributions of the co-authors are as follows:

- Prof. Hanwang Zhang pointed out the research direction and participated in the paper writing.
- Qianru Sun, Xian-Sheng Hua joined the early discussion and helped with polishing the paper before submission.
- I designed the algorithm, took charge of the entire code implementation and wrote the draft paper for other co-authors to refine.

The first part of Chapter 4 is submitted as [Zhongqi Yue, Qianru Sun, Xian-Sheng Hua, Hanwang Zhang. Make the U in UDA Matter: Invariant Consistency Learning for Unsupervised Domain Adaptation. 2022. \(WILDS 2.0 Leaderboard 1st Place as of application submission date\).](#)

The contributions of the co-authors are as follows:

- Prof. Hanwang Zhang pointed out the research direction and participated in the paper writing.
- Qianru Sun, Xian-Sheng Hua polished the paper before submission.

- I designed the algorithm, took charge of the entire code implementation and wrote the draft paper for other co-authors to refine.

The second part of Chapter 4 is submitted as [Zhongqi Yue, Xinting Hu, Yew Soon Ong, Qianru Sun, Hanwang Zhang. Debiased Pseudo-Labeling for Semi-Supervised Semantic Segmentation. 2023.](#)

The contributions of the co-authors are as follows:

- Prof. Hanwang Zhang pointed out the research direction and participated in the paper writing.
- Yew Soon Ong, Qianru Sun polished the paper before submission.
- Xinting Hu polished the paper and executed part of the experiments.
- I designed the algorithm, took charge of the entire code implementation and wrote the draft paper for other co-authors to refine.

The first part of Chapter 5 is published as [Zhongqi Yue*, Tan Wang*, Qianru Sun, Xian-Sheng Hua, Hanwang Zhang. Counterfactual Zero-Shot and Open-Set Visual Recognition. 2021. *IEEE Conference on Computer Vision and Pattern Recognition \(CVPR\)*.](#)

The contributions of the co-authors are as follows:

- Prof. Hanwang Zhang pointed out the research direction and participated in the paper writing.
- Qianru Sun and Xian-Sheng Hua joined discussion about the algorithm and evaluation, and helped with polishing the paper before submission.
- Tan Wang was in charge of the implementation and execution of Open-Set Recognition experiments.
- I designed the core algorithm, took charge of algorithm implementation, executed the Zero-Shot Learning experiments and wrote the draft paper for other co-authors to refine.

The second part of Chapter 5 is submitted as [Zhongqi Yue, Jiankun Wang, Qianru Sun, Lei Ji, Eric I-Chao Chang, Hanwang Zhang. Exploring Diffusion Time-steps for Unsupervised Representation Learning. 2023.](#)

The contributions of the co-authors are as follows:

- Prof. Hanwang Zhang pointed out the research direction and participated in the paper writing.
- Qianru Sun, Lei Ji and Eric I-Chao Chang participated in early discussion, tracked and guided the progress of the project, and helped with polishing the paper before submission.
- Jiankun Wang executed the experiments about counterfactual generation and image manipulation.

.....

$$\frac{1}{2} \frac{1}{\sqrt{2}}$$

Acknowledgements

First of all, I would like to thank my supervisor Prof. Hanwang Zhang for his hands-on guidance throughout my Ph.D. study. When I first started my Ph.D., I was a complete novice in research. He patiently taught me everything—from the grand scheme of things (*e.g.*, idea formulation) to the nitty-gritty details (*e.g.*, best practice to format a LaTeX project). When things got tough and I thought about quitting, he was right by my side and motivated me to keep going. Even when I encounter personal difficulties, he was compassionate and offered tremendous support that helped me cope. Thanks to him, I gained so much in this journey and I can't wait to climb the next mountain.

I would like to thank my collaborators Ph.D. Tan Wang, Dr. Hui Lv, Dr. Xinting Hu, Yanghao Wang, Jiali Ma, Jiankun Wang for their support. I am also grateful for the insightful comments from Prof. Qianru Sun and Xian-Sheng Hua. I want to thank Dr. Kaihua Tang, Dr. Jiabin Qi, Ph.D. Beier Zhu, Dr. Dong Zhang, Dr. Yaoyao Liu, and Dr. Chi Zhang for their helpful discussion and precious advice. In particular, I would like to thank my roommate Ph.D. Weishi Wang and Yiming Wang when we huddle for warmth in the most difficult times.

I would like to thank Alibaba-NTU Singapore Joint Research Institute for the amazing Ph.D. opportunity and School of Computer Science and Engineering at Nanyang Technological University for its great facilities, knowledgeable faculty members, and kind administration staff. I also want to thank Alibaba DAMO Academy, particularly City Brain Group, for their collaboration and generous sharing of computational resources.

I am extremely grateful to my kind and loving parents—Chunsheng Yue and Haiying Zhang, who comforted me by letting me know it's ok to fail. I also thank my beloved wife Shuangxiu Li for being my panacea to sorrow, anxiety, and upset, for her constant tolerance of a busy man of few words, and for her heartwarming company on the long nights near the deadlines.

Last but certainly not least, I would like to praise and thank God, the Almighty, who has granted me countless blessings, knowledge, and opportunity to accomplish the thesis. To God be the glory!

Zhongqi Yue, July 2023

Abstract

Causal reasoning, an essential cognitive ability in human intelligence, allows us to generalize past learning to solve present problems. Unfortunately, while machine learning has prospered over the past decade by training powerful deep neural networks (DNN) on massive data, it still lacks the generalization ability like that of humans and performs poorly when the deployment environment differs from that of training.

Inspired by the important role of causality in human generalization, we take a causal view of machine generalization by introducing the classic notion of causal representation learning: Each data sample is generated from the **causal feature** $\mathbf{c}$ determining its class identity and the class-invariant **environmental feature** $\mathbf{e}$. However, the spurious correlation between $\mathbf{c}$ and $\mathbf{e}$ in the training data is inevitable (*e.g.*, $\mathbf{c}$ = “bird” is often in $\mathbf{e}$ = “sky”) due to the ubiquitous bias in real-world data, which makes the training data a **confounder**. As a result, the conventional maximum likelihood objective (*e.g.*, cross-entropy loss) introduces confounding effect, where the trained model recklessly exploits the correlation for prediction (*e.g.*, classifying any sample in “sky” as “bird”). The confounded model generalizes poorly when the correlation no longer holds in deployment.

To tackle this, it has been theoretically proven that the confounding effect can only be eliminated by causal intervention, which pursues the causal effect from $\mathbf{c}$ to class identity. In this thesis, we study three categories of causal intervention and contribute practical implementations to improve generalization:

1. **Backdoor adjustment** assumes that we can observe and stratify the confounder, which is however elusive in real-world machine learning tasks. We devise an approximation of confounder stratification based on two inherent properties of pre-trained DNN: feature channels and pre-trained classifier and perform a case study in Few-Shot Learning setting. Furthermore, we propose an

unsupervised method to identify the confounder stratum by disentangling independent causal mechanisms, and successfully apply it in Unsupervised Domain Adaptation (UDA) setting.

2. **Invariant learning** stems on multiple disjoint environments where the spurious correlation between $\mathbf{c}$ and $\mathbf{e}$ differs. In this way, seeking an invariant classifier that is simultaneously optimal across all environments provably captures the causal effect. We show that the labeled data and unlabeled data in UDA and Semi-Supervised Learning naturally form such environments. Hence we learn an invariant classifier that is simultaneously consistent with the labeled data and clusters in the unlabeled data, removing the confounding bias.
3. **Disentangled representation.** An alternative way to achieve causal intervention is to learn $\mathbf{c}$ and $\mathbf{e}$ as two disjoint modular features (*i.e.*, a disentangled representation), hence a classifier can capture the causal effect by using only $\mathbf{c}$ for prediction. First, we consider supervised feature disentanglement in Zero-Shot Learning and Open-Set Recognition, where $\mathbf{c}$ is labeled in the training data, and we design a counterfactual inference algorithm to leverage the disentangled feature. Then we relax the need for labeled $\mathbf{c}$ and explore unsupervised disentanglement by using the inductive bias from a pre-trained Denoising Diffusion Probabilistic Model.

The proposed practical implementations are extensively evaluated by standard benchmarks and demonstrate state-of-the-art generalization performance in Few-Shot Learning, Unsupervised Domain Adaptation, Semi-Supervised Learning, Zero-Shot Learning, Open-Set Recognition, and Unsupervised Representation Learning.

Contents

Acknowledgements	xi
Abstract	xiii
List of Figures	xix
List of Tables	xxv
1 Introduction	1
1.1 Causal View of Generalization	1
1.2 Confounder Prevents Generalization	3
1.3 Causal Intervention De-Confounds	5
1.4 Thesis Outline	9
2 Literature Review	13
2.1 Causal Inference	13
2.2 Few-Shot Learning	14
2.3 Unsupervised Domain Adaptation	14
2.4 Semi-Supervised Learning	16
2.5 Zero-Shot and Open-Set Recognition	17
2.6 Disentangled Representation Learning	18
3 De-confound by Backdoor Adjustment	19
3.1 Preliminaries	20
3.2 Case Study: Few-Shot Learning	21
3.2.1 Introduction	21
3.2.2 Problem Formulations	23
3.2.3 Interventional Few-Shot Learning	27
3.2.4 Experiments	30
3.2.5 Supplementary Information	34
3.3 Case Study: Unsupervised Domain Adaptation	38
3.3.1 Introduction	38
3.3.2 Disentangled Causal Mechanisms Discovery	41
3.3.3 Representing U by Proxy Variables	45

3.3.4	Experiment	48
3.3.5	Supplementary Information	53
3.4	Conclusion	61
4	De-confound by Invariant Learning	63
4.1	Preliminaries	63
4.2	Case Study: Unsupervised Domain Adaptation	65
4.2.1	Introduction	65
4.2.2	Problem Formulations	68
4.2.3	Step 1: Identify Clusters in T	69
4.2.4	Step 2: Invariant Consistency (ICON)	70
4.2.5	Invariant Learning in UDA	71
4.2.6	Experiment	73
4.2.7	Supplementary Information	79
4.3	Case Study: Semi-Supervised Learning	81
4.3.1	Introduction	81
4.3.2	Problem Formulations	84
4.3.3	Clustering Assumption in SSL	86
4.3.4	DePL Objective	88
4.3.5	Experiments	89
4.3.6	Supplementary Information	96
4.4	Conclusion	98
5	De-confound by Learning Disentangled Representation	101
5.1	Preliminaries	101
5.2	Case Study: Supervised Learning	102
5.2.1	Introduction	102
5.2.2	Problem Formulations	105
5.2.3	Generative Causal Model	106
5.2.4	Counterfactual Generation and Inference	108
5.2.5	Counterfactual-Faithful Training	110
5.2.6	Experiments	112
5.2.7	Results on ZSL	114
5.2.8	Results on OSR	115
5.2.9	Supplementary Information	117
5.3	Case Study: Unsupervised Learning	120
5.3.1	Introduction	120
5.3.2	Problem Formulations	122
5.3.3	Inductive Bias of Diffusion Time-steps	123
5.3.4	Proposed Approach	125
5.3.5	Experiments	127
5.3.6	Inference Results	128
5.3.7	Counterfactual Generation	131

5.3.8	Supplementary Information	133
5.4	Conclusion	135
6	Summary	137
	List of Author's Awards, and Publications	141
	Bibliography	143

List of Figures

1.1	An example of human generalization, when understanding the causality behind ball motion on earth enables counterfactual inference about its motion on the moon.	1
1.2	Examples of machine generalization.	2
1.3	A model trained on only red digits generalizes to colorful digits in testing by learning the causal feature (<i>i.e.</i> , digit shape), as the environmental feature (<i>i.e.</i> , digit color) is not discriminative in training. We color the “0”s and “1”s in MNIST [1] differently between training and test data, and learn a linear classifier. Features are visualized by t-SNE [2], where ● / ● denotes “0” / “1”. The classification boundary is rendered blue and red.	3
1.4	An example where the model fails to generalize. Red : training, Blue : testing. ●: “0”, ▲: “1”. The red line denotes the classifier trained in S . Wrongly classified samples are circled in black.	3
1.5	SCM for machine generalization. (a) Without causal intervention. (b) With intervention.	4
1.6	A real-world example of a confounded model that captures the spurious correlation in training. Left: Confusion matrix on VisDA-2017 [3] test split. Right: GradCAM [4] visualization of the model prediction on the train and test split. The text on top of an image denotes the model prediction. Red text means that the model makes the wrong prediction.	5
1.7	Example of performing backdoor adjustment on training data with spurious correlation. $D = d_1$ or $D = d_2$ denotes two digit colors. Circles and triangles denote “0”s and “1”s, respectively.	6
1.8	An example of the SCM where an image classifier is trained on real-world photos to generalize to clipart images.	6
1.9	An invariant learning example. Red : 1st environment, Blue : 2nd environment. ●: “0”, ▲: “1”. We color the optimal regions of the classifier in red and blue for the two environments, respectively.	7
1.10	A disentangled representation (right) achieves higher Average Precision (AP) in attribute prediction than an entangled one (left). It also enables manipulation of a specific feature $\mathbf{c}$ =“young”/“brown hair”/“mustache” without affecting other features.	8
1.11	Overall thesis outline.	9

3.1	SCM for machine generalization.	19
3.2	The relationships among different FSL paradigms (color green and orange). Our goal is to remove the deficiency introduced by Pre-Training.	21
3.3	Quantitative and qualitative evidences of pre-trained knowledge misleading the fine-tune FSL paradigm. (a) <i>mini</i> ImageNet fine-tuning accuracy on 1-/5-/10-shot FSL using weak and strong backbones: ResNet-10 and WRN-28-10. $\mathcal{S} \sim \mathcal{Q}$ (or $\mathcal{S} \not\sim \mathcal{Q}$) denotes the pre-trained classifier scores of the query is similar (or dissimilar) to that of the support set. “Average” is the mean of both. The dissimilarity is measured using query hardness defined in Section 5.1. (b) An example of 5-shot $\mathcal{S} \not\sim \mathcal{Q}$	22
3.4	(a) Causal Graph for FSL; (b) Feature-wise illustration of $D \rightarrow C$: Feature channels of pre-trained network (<i>e.g.</i> 1 . . . 512 for ResNet-10). $X \rightarrow C$: Per-channel response to an image (“school bus”) visualized by CAM[5]; (c) Class-wise illustration for $D \rightarrow C$: features are clustered according to the pre-training semantic classes (colored t-SNE plot[6]). $X \rightarrow C$: An image (“school bus”) can be represented in terms of the similarities among the base classes (“ashcan”, “unicycle”, “sign”).	25
3.5	Causal graphs with sampling process. (a) Many-Shot Learning, where $P(Y X) \approx P(Y do(X))$; (b) Few-Shot Learning where $P(Y X) \not\approx P(Y do(X))$; (c) Interventional Few-Shot Learning where we directly model $P(Y do(X))$	27
3.6	IFSL algorithms for fine-tuning and meta-learning.	29
3.7	Accuracy across query hardness on 5-shot fine-tuning and meta-learning.	32
3.8	Some <i>mini</i> ImageNet visualizations of Grad-Cam [4] activation of query images and the CAM-Acc (%) table of using linear classifier and MAML. Categories with red text represent failed cases.	33
3.9	A DA example from source “Real World” to target “Clipart” domain in the Office-Home benchmark [7].	38
3.10	Causal graph of domain adaptation.	39
3.11	(a) Our DCMs $\{(M_i, M_i^{-1})\}_{i=1}^k$, where each $M_i : X_s \rightarrow X_t$ and $M_i^{-1} : X_t \rightarrow X_s$ correspond to intervention on the disentangled attribute U_i . (b) CycleGAN [8] loss $\mathcal{L}_{CycleGAN}^i$ on each (M_i, M_i^{-1})	43
3.12	Transformation between “Art” and “Clipart” domain with 4 trained DCMs $\{(M_i, M_i^{-1})\}_{i=1}^4$, where each pair of mapping functions specializes in brightness, exposure, color temperature and sharpness, respectively.	43
3.13	Causal graph with the proxy $\hat{X}$ and Z	45
3.14	Plot of k against Accuracy (%) on the 6 UDA tasks from ImageCLEF-DA [9].	51

3.15	t-SNE [6] plot of features in $S = s$ (red) and in $S = t$ (blue) after training using method: (a) GVB-GD [10], (b) Our TCM. Only samples from the first 15 classes are plotted to avoid clutter.	51
3.16	Class Activation Maps (CAMs) [5] of GVB-GD [10], Baseline and TCM on the Artistic($S = s$) $\rightarrow$ Clipart($S = t$) task in Office-Home dataset [7]. The left column shows two samples in each domain, whose class name is indicated on the left. In $S = t$, we show two samples predicted wrongly by GVB-GD and Baseline, but correctly with TCM.	52
3.17	Supplementary to Figure 4. Transformation between “Real World” (R) and “Clipart” (C) domain with 4 trained DCMs $\{(M_i, M_i^{-1})\}_{i=1}^4$. The winning DCM is outlined in red.	59
3.18	Class Activation Maps (CAMs) [5] of GVB-GD [10], Baseline and TCM on the Real World($S=s$) $\rightarrow$ Clipart($S = t$) task in Office-Home dataset [7]. The left column shows two samples in each domain, whose class name is indicated on the top. In $S = t$, we show two samples predicted wrongly by GVB-GD and Baseline, but correctly with TCM. Supplementary to Figure 8	60
3.19	Class Activation Maps (CAMs) [5] of GVB-GD [10], Baseline and TCM on the Real World($S=s$) $\rightarrow$ Clipart($S = t$) task in Office-Home dataset [7]. The left column shows two samples in each domain, whose class name is indicated on the left. All methods fail on the two samples. Supplementary to Figure 8.	60
3.20	UDA accuracy (%) using DANN, CDAN, SymNets and our TCM, on different training iterations. Batch size is 32.	61
4.1	We color the “0”s and “1”s in MNIST [1] differently to form S and T , and train a linear classifier in S . Features are visualized by t-SNE [2], where $\bullet$ / $\bullet$ denotes “0” / “1”. The classification boundary is rendered blue and red.	66
4.2	Red: S , Blue: T . $\bullet$: “0”, $\blacktriangle$: “1”. (a) The red line denotes the classifier trained in S . (b) Classifier (black line) can be corrected by respecting the BCE loss between S (red line) and T (blue line) (c) A failure case of minimizing the combined BCE losses in S and T (black line is wrongly tilted). Due to the low variance of $\mathbf{e}_S$ in each class, the optimal classifier <i>w.r.t.</i> the BCE loss in S (red line) looks “flat”, <i>i.e.</i> , it has large weight on $\mathbf{e}$ [11]. (d) Our ICON prevents such a failure case (the intersection area of the red and blue candidate classifier regions). Best viewed in color.	66
4.3	S : Clipart and T : Real-world from OFFICE-HOME [7] dataset. Without loss of generality, we give an example where the pair of samples is from different classes in S and the same cluster in T , respectively. The black arrows on the similarity bars denotes that minimizing the BCE losses in S and T will decrease the similarity in S (as $y_1 \neq y_2$) and increase that in T (as $y_1 = y_2$).	69

4.4	Red: S , Blue: T . $\bullet$: “0”, $\blacktriangle$: “1”. Black lines denote the invariant classifier satisfying the ICON objective. (a) Assumption 1 is violated. (b) Assumption 2 is violated.	72
4.5	Ablations of α, β on the classic UDA datasets.	76
4.6	(a) Solid lines: test accuracy against training epochs for FixMatch , CST and ICON . Dotted black lines: average loss in each epoch with matching markers to denote the three methods. (b) Among the feature pairs where f fails, the percentage of pairs where g succeeds for the three methods in each epoch.	77
4.7	CST and ICON confusion matrix on VisDA-2017.	77
4.8	GradCAM on VisDA-2017 [3] using CST [12], the cluster head g and the proposed ICON. Ground-truth knives and boards are in solid and dotted boxes, respectively. Predictions from f, g are on top of each row. Red denotes wrong predictions. Pale blue and red background denotes training and testing, respectively. Note that g is trained without supervision in T	78
4.9	Results of UniMatch [13] (Section 4.3.5) trained on 92 images from PASCAL VOC 2012 [14]. (a) The ground-truth (GT) and predicted (Pred) semantic labels on “sofa” and “chair” images in $\mathcal{D}, \mathcal{U}$ and testing. (b) The class-wise IoU of pseudo-labels and in testing. (c) The confusion matrix in testing. We show a subset of the classes to avoid clutter. Full results in Section 4.3.6.	82
4.10	Circle denotes the pixel-level features in $\mathcal{D}$ and $\mathcal{U}$ in a 2D feature space. (a) An illustrated failure case of pseudo-labeling, where context bias in $\mathcal{D}$ leads to wrong pseudo-labels on $\mathcal{U}$. (b) Our DePL learns a debiased model (black line) to prevent such failure. Best viewed in color.	83
4.11	Clustering assumption on different intermediate outputs of the model. We used 4 images in $\mathcal{U}$ of PASCAL VOC 2012 dataset. GT denotes the ground-truth mask. $\bullet$: background; $\bullet$ and $\bullet$: the first and second non-background semantic classes in each image. (a) Local variations between each pixel and its 8 neighbors using the backbone feature. (b) Using decoder feature. (c) Using decoder feature after average pooling (7×7 kernel).	87
4.12	Pipeline of our proposed DePL. On the left, we show how $\mathcal{U}_f$ is constructed to compute $\text{BCE}(\mathcal{U}_f)$ using two features in $\mathcal{U}$ from the same Cluster 1 (C1) as an example. On the right, we show how CE is computed for a “sofa” feature and “chair” feature in $\mathcal{D}$. The optimal classification boundary for the CE loss and BCE loss are shown with the top and bottom diagram in the middle, respectively, which are combined to learn the unbiased one (black line in the center diagram). We omit 1) the learning of cluster head g as the process is similar with $\text{BCE}(\mathcal{U}_f)$; 2) pseudo-labeling loss as the process is similar with CE.	87
4.13	Ablation on α, β	93

4.14	Predictions from UniMatch [13] and UniMatch+DePL (Ours). GT denotes the ground-truth mask. We used the models trained in 1/16 (92) on VOC <i>classic</i> . We highlight the label name when necessary. .	95
4.15	Confusion matrix for UniMatch [13] trained on 1/16 PASCAL VOC 2012 [15] <i>classic</i> setting. Supplementary to Figure 4.9	96
4.16	Confusion matrix for our DePL trained on 1/16 PASCAL VOC 2012 [15] <i>classic</i> setting. Supplementary to Figure 4.9	96
4.17	Class-wise IoU of UniMatch [13] and DePL on 1/16 PASCAL VOC 2012 [15] <i>classic</i> setting. Supplementary to Figure 1. The last plot shows the %improvements of DePL over UniMatch on each class. .	97
4.18	Clustering assumption on different intermediate outputs of the model. We used 6 images in $\mathcal{U}$ of PASCAL VOC 2012 dataset. Supplementary to Figure 4.12.	97
4.19	Predictions from UniMatch [13] and UniMatch+DePL (Ours). GT denotes the ground-truth mask. We used the models trained in 1/16 (92) on VOC <i>classic</i> . Supplementary to Figure 4.14	98
5.1	Disentangled representation enables generalization.	102
5.2	Our counterfactual framework for ZSL and OSR. Here $\oplus$ denotes vector concatenation, $\mathcal{Y}_U$ and $\mathcal{Y}_S$ denote the set of unseen and seen class attributes, respectively.	103
5.3	(a) t-SNE [6] plot of the CUB [16] samples in ZSL using a conventional unseen generation method [17], where a single pair of seen- and unseen-class is shown to avoid clutter. Due to the out-of-distribution generation (green), the decision boundary (black dashed lines) is imbalanced between the true seen (red) and unseen (blue) samples. (b) Illustration of our counterfactual generation. (c) t-SNE plot of the CUB using our counterfactual generation. The decision boundary is balanced. See Figure 5.5 for more examples of OSR.	104
5.4	Our GCM for ZSL and OSR.	106
5.5	OSR using (a) our counterfactual framework ($Z=z(X)$); (b) existing generative approach ($Z \sim \mathcal{N}(0, I)$) when the test sample is from seen-classes (left) and unseen-classes (right). Red dashed line denotes the true samples from the seen-class and blue dash line as those from the unseen-class. Green dash line denotes generated samples using the seen-class attribute. $\star$ and $\star$ denote the samples from seen and unseen class, respectively.	107
5.6	Counterfactual generation conditioned on an unseen sample (denoted as the star). $Z_U \oplus Y_S$ and $Z_U \oplus Y_U$ represent counterfactual images conditioned on the unseen sample, using seen and unseen attribute respectively.	110
5.7	The Seen-Unseen accuracy Curve (SUC) together with the Area Under SUC (AUSUC) on the four datasets. The rectangle denotes the ZSL Accuracy when calibration factor $\omega=0$	116

5.8	Comparison of the reconstructed images using CGDL [18] ($y(X) \oplus \mathcal{N}(0, I)$) and the counterfactual images generated from our GCM-CF ($y \oplus z(X)$) on the seen- and unseen-class samples. The red box on a generated image denotes that it is similar to the original. For CIFAR10, due to the conflict between visual perception and discriminative training [19], we first train a classifier on seen-classes images and then use CAM [5] to show the sensible yet discriminative features: although the pixel-level generation is not sensible, the pixel-level distance remains discriminative as shown in Table 5.4.	116
5.9	Openness-F1 Plot where 4 non-animal classes from CIFAR10 [20] were used as seen-classes and various classes were drawn from CIFAR100 [20] as unseen-classes.	117
5.10	Visualization of the covariance matrix in AWA2 dataset of the (a) seen-classes and (b) unseen-classes attributes.	119
5.11	(a) Illustration of attribute loss as time-step t increases in the forward diffusion process. The two axes depict a two-dimensional sample space. (b) DM reconstructed $\mathbf{x}_0$, denoted as $\hat{\mathbf{x}}_0$, from randomly sampled $\mathbf{x}_t$ at various time-steps t . DM is pre-trained on CelebA, from where $\mathbf{x}_0$ is drawn.	121
5.12	Illustration of our DiTi. On the left, we break down the learning with Eq. (5.17) at each time-step, and the loss at t is denoted as $\mathcal{L}_t$. On the right, we show the detailed network design. We follow PDAE [21] for the decoder design. $\hat{\mathbf{x}}_0$ denotes the reconstructed $\mathbf{x}_0$ by the pre-trained DM.	125
5.13	Improvements in attribute classification precision of our DiTi over PDAE (top) and SimCLR (bottom).	129
5.14	Counterfactual generations on FFHQ. $\mathbf{x}_0$ and $\mathbf{x}'_0$ are marked with a green box.	130
5.15	Counterfactual generation on CelebA.	131
5.16	Ablation on evaluation methods.	133
5.17	The PDFs of $q(\mathbf{x}_t \mathbf{x}_0)$ (green) and $q(\mathbf{y}_t \mathbf{y}_0)$ (blue). Without loss of generality, we consider the 1-D case of $y_0 > x_0$. Their means are computed from Eq. (2).	134
6.1	Existing paradigm of generalization tasks.	138
6.2	Our proposed paradigm that incorporates causal intervention. We include the relevant chapter on the top.	139

List of Tables

3.1	Acc (%) averaged over 2000 5-way FSL tasks before and after applying IFSL. We obtained the results by using official code and our backbones for a fair comparison across methods. We also implemented SIB in both transductive and inductive setting to facilitate fair comparison. For IFSL, we reported results of combined adjustment as it almost always outperformed feature-wise and class-wise adjustment. See Section 3.2.5 for Acc and 95% confidence intervals on all 3 types of adjustment.	31
3.2	Comparison with state-of-the-arts of 5-way 1-/5-shot Acc (%) on <i>miniImageNet</i> and <i>tieredImageNet</i>	33
3.3	Results of cross-domain evaluation: <i>miniImageNet</i> $\rightarrow$ CUB.	33
3.4	Supplementary to Table 3.1 Acc (%) and 95% confidence intervals averaged over 2000 5-way FSL tasks before and after applying three proposed implementations of adjustment. Specifically, “ <i>ft</i> ” refers to feature-wise adjustment, “ <i>cl</i> ” refers to class-wise adjustment and “ <i>ft+cl</i> ” refers to combined adjustment.	35
3.5	Accuracy (%) on the Office-Home dataset [7] with 12 UDA tasks, where all methods are fine-tuned from ResNet-50 [22] pre-trained on ImageNet [23].	48
3.6	Accuracy (%) and the standard deviation on the ImageCLEF-DA dataset [9] with 6 UDA tasks, where all methods are fine-tuned from ResNet-50 [22] pre-trained on ImageNet [23].	49
3.7	Accuracy (%) of Synthetic $\rightarrow$ Real task on the VisDA-2017 dataset [3].	51
3.8	Ablation on the number of DCMs k with accuracy (%) on the 6 UDA tasks from ImageCLEF-DA [9].	51
3.9	GFLOPS of feature extractor backbone and additional modules (<i>e.g.</i> , discriminator networks) as well as #parameters of those additional modules on Office-Home.	61

4.1	Dataset details and the results of our proposed ICON compared with existing methods. Details on * in Section 4.2.6. WILDS 2.0 [24] datasets are highlighted in yellow, where models were evaluated on both the test data in S (first number) and in T (second number). As of thesis submission deadline, ICON dominates the WILDS 2.0 leaderboard (https://wilds.stanford.edu/leaderboard/), with the performance gain in T highlighted in red. “ERM (labeled T)” has full supervision in T . -/- means that the method is not applicable to the dataset.	73
4.2	Accuracy (%) on the OFFICE-HOME dataset [7] with 12 UDA tasks.	75
4.3	Ablations on each ICON component.	76
4.4	Comparison with <i>state-of-the-art</i> methods on the Pascal VOC 2012 dataset in the <i>classic</i> setting. We reproduced UniMatch [13] with its official code on the same machine and code environment with where we conducted our DePL experiments. We indicate the improvements from our DePL with red numbers. The best result in each column is in bold	90
4.5	Comparison with <i>state-of-the-art</i> methods on the Pascal VOC 2012 dataset in the <i>blended</i> setting.	91
4.6	Comparison on the Cityscapes dataset. We list the AEL results reproduced by U ² PL [25].	93
4.7	Ablation on the kernel size for average pooling. 1×1 corresponds to using no average pooling.	94
4.8	Ablation on the similarity measure. $\text{cosine} > s$ denotes that $\mathbb{1}(\bar{\mathbf{x}}_i \sim \bar{\mathbf{x}}_j)$ in Eq. (4.12) returns 1 (<i>i.e.</i> , similar) if their cosine similarity is larger than s , and 0 otherwise. Correspondingly, $\text{top-}k$ denotes that two features are similar if their $\text{top-}k$ feature values have the same indices.	94
4.9	Comparison of class-wise IoU on 1/32 VOC <i>blended</i> . We shorten the name of some semantic classes. bg: background, pln: aeroplane, botl: bottle, tabl: table, hrs: horse, mbk: motorbike, prs: person, shp: sheep, trn: train.	94
5.1	ZSL Accuracy ($U\%$, $S\%$, $H\%$) on the four datasets, where Inf. means inference-based methods and Gen. means generation-based methods. Note that PS V2.0 was released recently in https://drive.google.com/file/d/1p9gtkuHCCCyjkyezSarCw-1siCSXUyKH/view to fix a test-data leaking bug. This can have large impacts on the performance of existing methods, such as GDAN [26]. Therefore all our evaluations were conducted on PS V2.0. † indicates that the results are taken from the PS V2.0 report [27], and we reproduced the results on all other methods using the official code. For our method GCM-CF, we used AREN [28] for Conventional ZSL on CUB, and otherwise used TF-VAEGAN [17] for supervised classification and Conventional ZSL.	113
5.2	CVb (%) of generative models on all datasets.	114

5.3	Comparison of the two-stage inference performance on AWA2 [29] using TF-VAEGAN [17] and our GCM-CF as the stage-one binary classifier. The results on other datasets are shown in Appendix, where using GCM-CF as stage-one binary classifier improves all the methods.	115
5.4	Comparison of the F1 score averaged over 5 random splits in OSR. We used the official code on CGDL [18] and implemented Softmax and OpenMax [30] for evaluation. For GCM-CF, after binary classification, we used CGDL for supervised classification on the seen-classes.	116
5.5	AP (%) on CelebA attribute classification and Pearson's r, MSE on LFW attribute regression. The first column shows the training dataset.	128
5.6	Ablations on k , partition and optimization strategy. Models were trained on CelebA. <i>Imbalance</i> and <i>Detach</i> means using our devised partition and optimization strategy.	128

Chapter 1

Introduction

1.1 Causal View of Generalization

Causal reasoning is a fundamental aspect of human thinking that enables understanding and prediction of the world around us. The basic premise is to unravel the causal structure of the world [31, 32], *i.e.*, identifying the modular factors (*e.g.*, gravity, friction) that generate observations (*e.g.*, the motion of a falling ball on earth) and their cause-and-effect relationships (*e.g.*, Newton’s second law of motion). This enables generalization by knowledge transfer to novel situations, allowing us to predict outcomes by counterfactual inference (*e.g.*, predicting the motion of a ball on the moon), learn new tasks efficiently (*e.g.*, studying a new

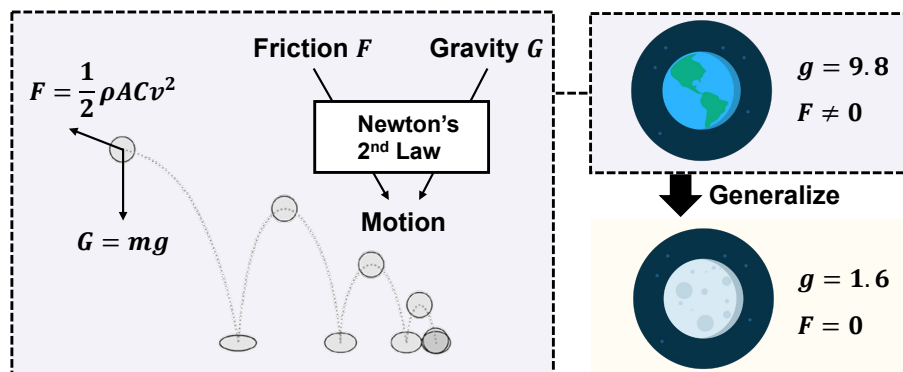


FIGURE 1.1: An example of human generalization, when understanding the causality behind ball motion on earth enables counterfactual inference about its motion on the moon.

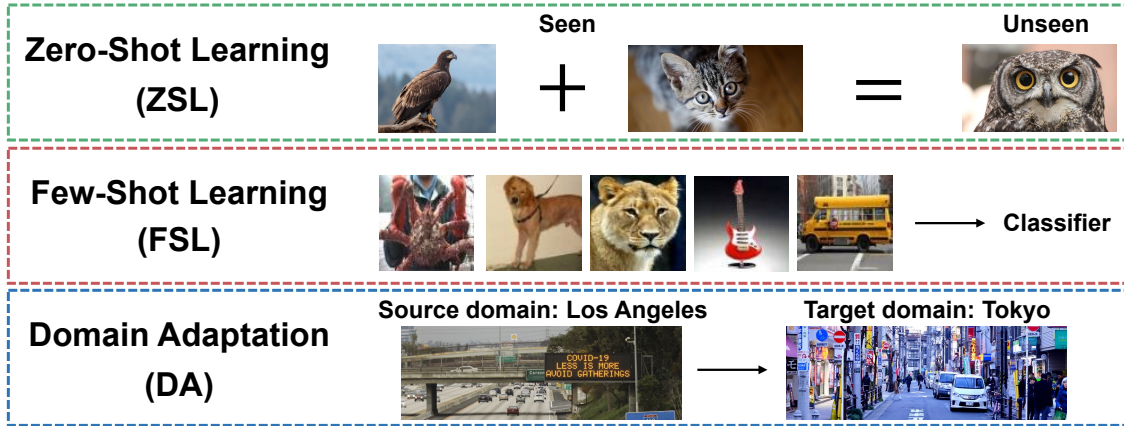


FIGURE 1.2: Examples of machine generalization.

language), and adapt to changing domains (*e.g.*, understanding social interactions of different cultures).

In fact, the aforementioned ability to generalize is nothing short of a panacea for any machine learning model that is practical in real-world applications, as models are trained on a finite dataset to perform well on *unseen* data in deployment. Similar to human generalization, machines that generalize embody the capability of knowledge transfer as shown in Figure 1.2, enabling 1) Zero-Shot Learning (ZSL) [29, 33], the capacity to correctly infer information about samples from novel classes, *e.g.*, a model trained on “eagle” and “cat” images interprets “owl” as an eagle with a cat face; 2) Few-Shot Learning (FSL) [34, 35], the ability to grasp new concepts from a few training samples, *e.g.*, learning “Yeti crab” with only 1 image; 3) Domain Adaptation (DA) [36, 37], the capability of adapting to different (but related) domains from the training one, *e.g.*, adapting an autopilot system trained in one country to another with different road signs.

To understand when a model is expected to generalize, we introduce the classic notion of causal representation learning [38, 39]: any observed sample $\mathbf{x}$ (*e.g.*, digit image) from class y (*e.g.*, digit) is generated by $\mathbf{x} = \Phi(\mathbf{c}, \mathbf{e})$, where Φ is the (hidden) generator, $\mathbf{c}$ is the causal feature (*e.g.*, digit shape) determining the class identity y , and $\mathbf{e}$ is the environmental feature (*e.g.*, digit color). A model generalizes when it captures the causal effect, *i.e.*, predicting y from $\mathbf{c}$, as such a model is an invariant predictor of y regardless of the distribution of $\mathbf{e}$. For example, Figure 1.2 shows a digit classification task. As $\mathbf{e} = \text{“red”}$ is non-discriminative in the training data (left), even by naive empirical risk minimization (ERM), the

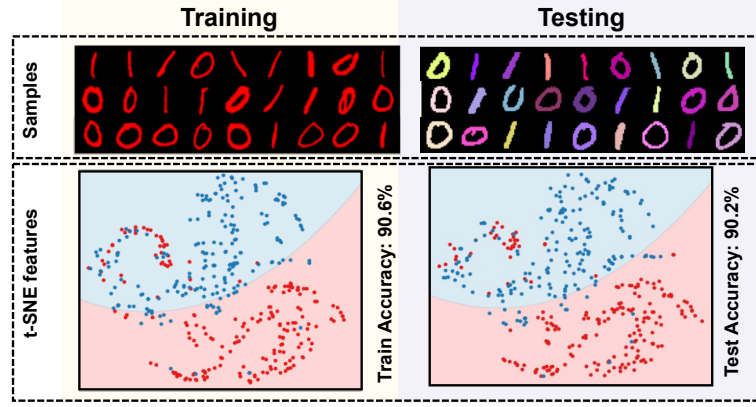


FIGURE 1.3: A model trained on only red digits generalizes to colorful digits in testing by learning the causal feature (*i.e.*, digit shape), as the environmental feature (*i.e.*, digit color) is not discriminative in training. We color the “0”s and “1”s in MNIST [1] differently between training and test data, and learn a linear classifier. Features are visualized by t-SNE [2], where $\bullet$ / $\bullet$ denotes “0” / “1”. The classification boundary is rendered blue and red.

model still easily disentangles $\mathbf{c}$ out of $\Phi(\mathbf{c}, \mathbf{e})$ for prediction. Hence despite never having seen $\mathbf{e} = \text{“colorful”}$, the model still generalizes well to colorful digits during testing (right).

1.2 Confounder Prevents Generalization

Unfortunately, in general, disentangling $\mathbf{c}$ by ERM is impossible as $\mathbf{e}$ can be also discriminative in the training data, *i.e.*, spuriously correlated with $\mathbf{c}$. Such spurious correlation can arise from selection bias. For example, if the training data is about real object photos, texture can be a useful cue to tell different objects apart (*e.g.*, mattress from desk surface). However, texture becomes useless to tell apart clip-art images where textures are seldom depicted. Another common source of spurious correlation is contextual bias, *e.g.*, cow usually appear with grass, hence a classifier trained on

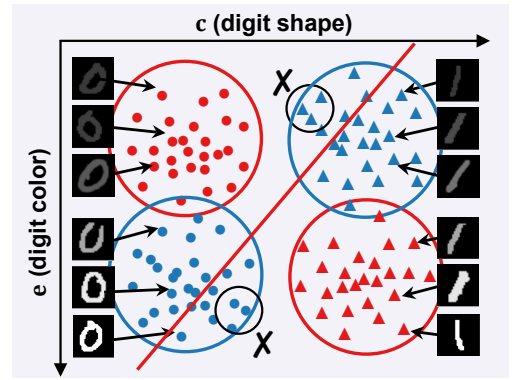


FIGURE 1.4: An example where the model fails to generalize. **Red**: training, **Blue**: testing. $\bullet$: “0”, $\blacktriangle$: “1”. The red line denotes the classifier trained in S . Wrongly classified samples are circled in black.

usual cow images may fail on a “cow on beach” image.

We illustrate the spurious correlation in Figure 1.4 with our digit examples, color $\mathbf{e}$ is correlated with shape $\mathbf{c}$, *i.e.*, $\mathbf{c}_1 = \text{“0”}$ and $\mathbf{c}_2 = \text{“1”}$ in S tend to have $\mathbf{e}_1 = \text{dark}$ and $\mathbf{e}_2 = \text{light}$ digit colors, respectively. As both $\mathbf{c}$ and $\mathbf{e}$ are discriminative, a trained classifier by ERM will inevitably capture both, *e.g.*, the red line in Figure 1.4. This leads to poor generalization when the correlation between $\mathbf{e}$ and $\mathbf{c}$ is different in testing, *i.e.*, the colors of $\mathbf{c}_1 = \text{“0”}$ and $\mathbf{c}_2 = \text{“1”}$ in T tend to be $\mathbf{e}_2 = \text{light}$ and $\mathbf{e}_1 = \text{dark}$ instead.

The causal relations in the above analysis are depicted with a Structural Causal Model (SCM) in Figure 1.5a. We denote X as the sample and Y as the predictive effect (*e.g.*, logits in classification). D denotes the environmental feature that spuriously correlates with labels in the training data. $D \rightarrow X$ denotes that samples are subject to the correlation (*e.g.*, most “0”s are “dark”). In particular, $D \rightarrow Y \leftarrow X$ is because, besides the causal feature embedded in X , D additionally provides useful contextual cues for predicting Y in training (*e.g.*, color is discriminative in classifying “0” and “1”).

An ideal model should capture the true causality between X and Y to generalize to unseen samples. For example, in Figure 1.4, we expect that the digit prediction is caused by the digit shape feature *per se*, but not the color. However, from the SCM in Figure 1.5a, the conventional objective that maximizes $P(Y|X)$ fails to do so, because the increased likelihood of Y given X is not only due to “ X causes Y ” via $X \rightarrow Y$, but also the spurious correlation

via the backdoor path $X \leftarrow D \rightarrow Y$, *e.g.*, “0”s tend to be darker. In fact, D as a common cause of X and Y is known as their confounder, which misleads the model to exploit the spurious correlation and prevents it to generalize.

In Figure 1.6, we show an example of a confounded model. The confusion matrix on the left reveals that the model predicted most “knife”s as “skateboard”s

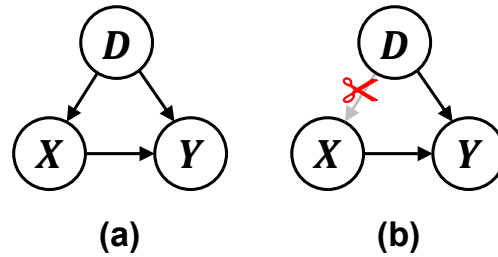


FIGURE 1.5: SCM for machine generalization. (a) Without causal intervention. (b) With intervention.

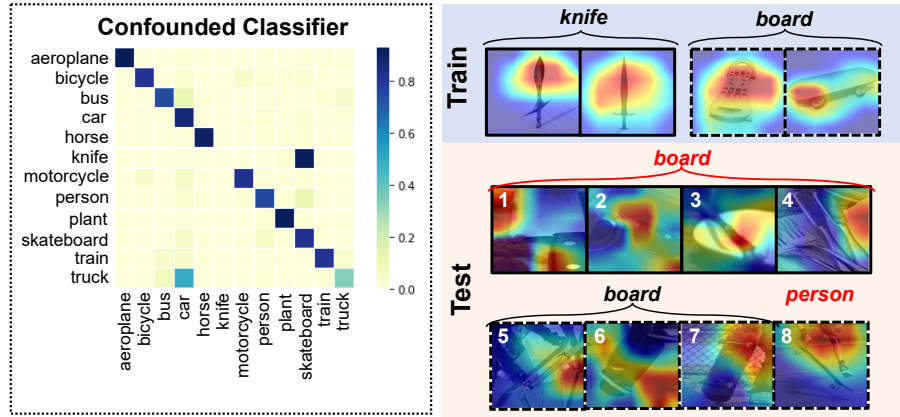


FIGURE 1.6: A real-world example of a confounded model that captures the spurious correlation in training. Left: Confusion matrix on VisDA-2017 [3] test split. Right: GradCAM [4] visualization of the model prediction on the train and test split. The text on top of an image denotes the model prediction. Red text means that the model makes the wrong prediction.

(“board” for short) in testing. We probed the reason by visualizing the GradCAM [4] response of the model. In the first row, we notice that the model predicts the two categories perfectly in training by leveraging their apparent differences: the curved blade versus the flat board ending with semi-circles. However, this leads to poor generalization on “knife” in testing (second row), where the background (image 1 and 4), the knife component (2), or even the blade (3) can have board-like appearances. Moreover, even if the model predicts correctly in testing (*i.e.*, 5-7), it coincidentally did so as it still leveraged the environmental feature (*e.g.*, background) for predicting “board”, which is prone to error, *e.g.*, legs of a person in the background of image 8 mislead the model to predict a “board” as “person”.

1.3 Causal Intervention De-Confounds

To pursue the true causality between X and Y and achieve generalization, we need to use *causal intervention* $P(Y|do(X))$ instead of the likelihood $P(Y|X)$ as theoretically proven in [40]. In particular, $P(Y|do(X))$ is computed by measuring $P(Y|X)$ among data generated by a modified SCM, where all incoming arrows to X are removed, as shown in Figure 1.5b. Note that in the new causal graph, X is no longer governed by the spurious correlation D , *e.g.*, we can observe diverse digit samples with “0”s and “1”s in dark and light color. Hence learning on diverse X free from spurious correlation removes the confounding effect (*i.e.*, de-confounds) and leads

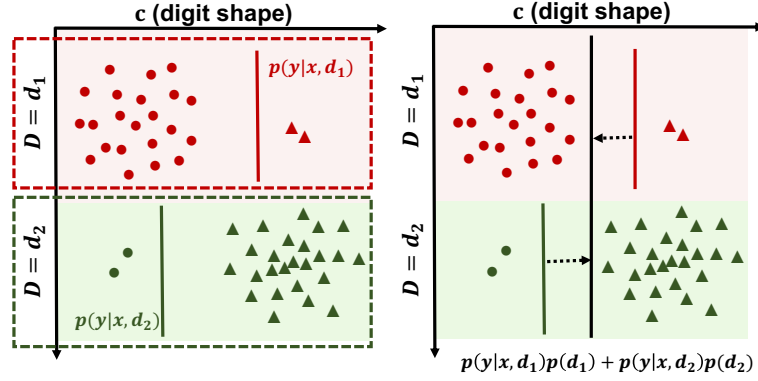


FIGURE 1.7: Example of performing backdoor adjustment on training data with spurious correlation. $D = d_1$ or $D = d_2$ denotes two digit colors. Circles and triangles denote “0”s and “1”s, respectively.

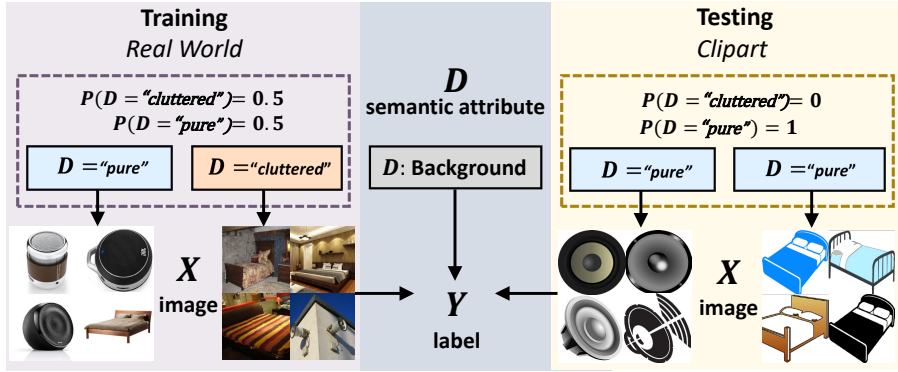


FIGURE 1.8: An example of the SCM where an image classifier is trained on real-world photos to generalize to clipart images.

to the desired causal effect (*e.g.*, only the causal feature “shape” is discriminative). We illustrate this further by drawing a connection between $P(Y|do(X))$ and many-shot learning in Section 3.2.

Unfortunately, collecting perfect training data with no spurious correlation is impossible, *e.g.*, the naturally long-tailed real-world observations [41, 42], or the ubiquitous label bias due to labelers’ preference. Hence in this thesis, we are interested in methods that perform causal intervention *without* requiring such training data. These methods can be divided into three categories:

Backdoor Adjustment [40] breaks down the interventional term $P(Y|do(X))$ (*i.e.*, with *do*) to only observational terms (*i.e.*, without *do*), which can be estimated from data distribution. For the SCM in Figure 1.5a, the formula is given by:

$$P(Y|do(X = \mathbf{x})) = \sum_d P(Y|X = \mathbf{x}, D = d) P(D = d), \quad (1.1)$$

which assumes that the confounder D is observable, and fairly adjusts the contribution from each stratum of D according to a fixed prior $P(D)$. We illustrate backdoor adjustment in Figure 1.7. Note that digit shape is spuriously correlated with color D : most “0”s (circles) have color d_1 , and most “1”s (triangles) have color d_2 . To perform backdoor adjustment, on the left, we first train separate classifiers $p(y|\mathbf{x}, d_1)$ and $p(y|\mathbf{x}, d_2)$ for each stratum of D . As the value of D is the same in each stratum (*i.e.*, non-discriminative), both classifiers can only leverage digit shape for prediction (*i.e.*, upright). On the right, we then combine them according to prior $p(d_1)$ and $p(d_2)$, which both correspond to 0.5 in this example. This leads to an overall classifier depicted by the black line, which is the desired classifier that separates “0”s from “1”s. In the example of Figure 1.8, a naively-trained classifier will recklessly exploit the background feature (*i.e.*, “pure”=“speaker”, “cluttered”=“bed”) and hence downplay the causal feature shape. This leads to poor generalization when “background”=“pure” is no longer discriminative in testing. In contrast, by performing backdoor adjustment in training, every stratum of the confounder D is fairly adjusted subject to the prior $P(D)$, *i.e.*, forcing the model to respect “shape” as it is the only “invariance” that distinguishes between “speaker” and “bed” in controlled “cluttered” or “pure” backgrounds, enabling generalization.

Invariant Learning [43]. Given a partition of the training dataset into subsets and a loss function, the goal is to learn a model that simultaneously minimizes the loss in all the subsets of the partition. We extend the digit classification example in Figure 1.9 to illustrate. The training data (“0”s with circle dot and “1”s with triangular dot) is divided into two environments colored red and blue, respectively. For the classification loss in each of the environments, we color the candidate regions where the classifier is optimal (*i.e.*, gradients ≈ 0) with red and blue, respectively. The classifier that is simultaneously optimal in both environments lies in the intersected region colored in purple, which generalizes to any digit classification task

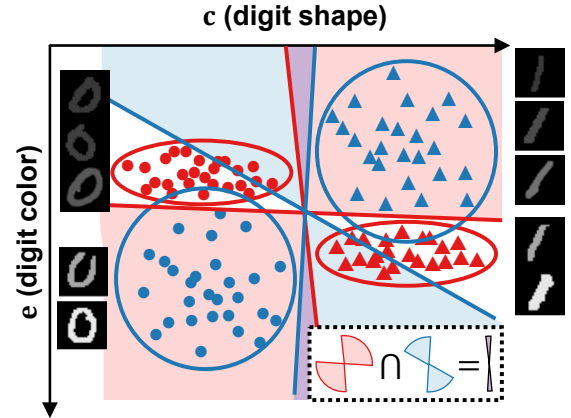


FIGURE 1.9: An invariant learning example. **Red**: 1st environment, **Blue**: 2nd environment. **●**: “0”, **▲**: “1”. We color the optimal regions of the classifier in red and blue for the two environments, respectively.

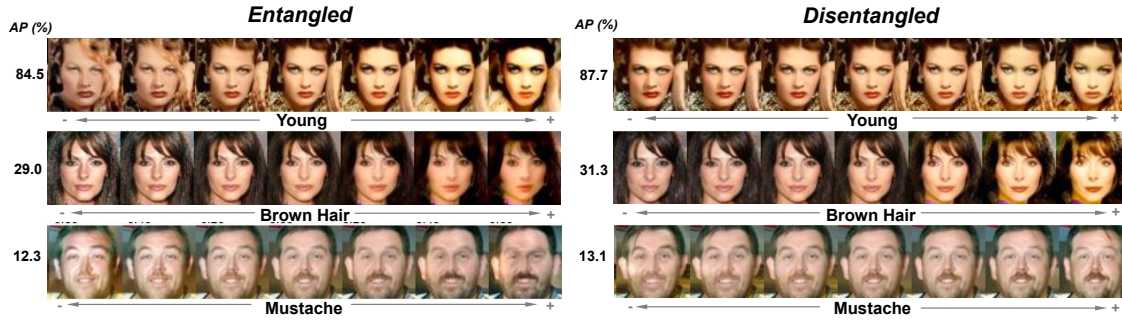


FIGURE 1.10: A disentangled representation (right) achieves higher Average Precision (AP) in attribute prediction than an entangled one (left). It also enables manipulation of a specific feature $\mathbf{c}$ = “young” / “brown hair” / “mustache” without affecting other features.

by only using the causal feature (*i.e.*, digit shape) for prediction.

Disentangled Representation [38] is an injective mapping $f : \mathcal{X} \rightarrow \mathcal{C} \times \mathcal{E}$ that inverts Φ , *i.e.*, given each $\mathbf{x}$ in the sample space $\mathcal{X}$, $f(\mathbf{x})$ recovers the causal feature $\mathbf{c} \in \mathcal{C}$ and environmental one $\mathbf{e} \in \mathcal{E}$ as modular attributes. A disentangled representation facilitates robust inference in downstream tasks about the causal feature $\mathbf{c}$, *e.g.*, a linear classifier can eliminate task-irrelevant environmental feature $\mathbf{e}$ to avoid spurious correlations by assigning 0 to the corresponding dimensions of its weight parameter, *e.g.*, Figure 1.10 shows that a disentangled representation achieves higher accuracy in attribute prediction than its entangled counterpart (details in Chapter 5.3). Moreover, when Φ is available (*e.g.*, a learned generative model), a disentangled representation enables extrapolation [44] by manipulating only the causal feature $\mathbf{c}$ without affecting $\mathbf{e}$, *e.g.*, manipulating only $\mathbf{c}$ = “young” / “brown hair” / “mustache” without affecting other features (*e.g.*, person’s identity) in Figure 1.10.

Yet the three causal intervention methods are limited by the following practical challenges: 1) The confounder D is diverse and unobserved [45]. Hence it is still unknown how to implement its stratification and representation, making backdoor adjustment impractical. 2) Meaningful data partition that enables invariant learning to de-confound is often not available in real-world datasets [46]. 3) Learning disentangled features is difficult, or even impossible without proper supervision [47]. We aim to address these challenges in this thesis, as outlined in the next section.

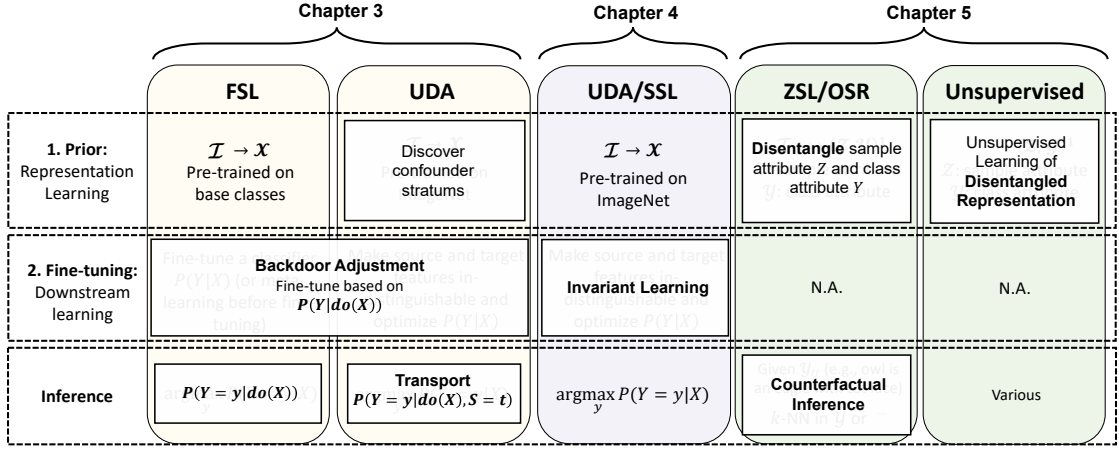


FIGURE 1.11: Overall thesis outline.

1.4 Thesis Outline

In this chapter, we begin by introducing the notion of machine generalization and formalizing it from a causal perspective. We then show that the presence of confounding effect prevents generalization, which must be removed by causal intervention. Next, we discuss three methods that perform intervention: 1) backdoor adjustment, 2) invariant learning, and 3) learning disentangled representation, yet deploying them in real-world tasks is limited by practical challenges.

The goal of this thesis is to break the limitation by devising practical implementation in each of the three aforementioned categories, thereby improving generalization in Zero-Shot Learning (ZSL), Few-Shot Learning (FSL), Semi-Supervised Learning (SSL), Unsupervised Domain Adaptation (UDA) and Open-Set Recognition (OSR) [48]. The outline of the thesis is illustrated in Figure 1.11 and explained below:

- In Chapter 2, we first review causal inference, with a special focus on its application in machine learning. Then we introduce the related works for FSL, UDA, SSL, ZSL, and OSR.
- In Chapter 3, we start by deriving the backdoor adjustment formula. Then we apply it to the following scenarios:

FSL: We uncover an ever-overlooked deficiency in the prevailing Few-Shot Learning (FSL) methods: the pre-trained knowledge is indeed a confounder that limits the performance. This finding is rooted in our causal assumption:

a Structural Causal Model (SCM) for the causalities among the pre-trained knowledge, sample features, and labels. Thanks to it, we propose a novel FSL paradigm: Interventional Few-Shot Learning (IFSL). Specifically, we develop three effective IFSL algorithmic implementations based on the back-door adjustment, which is essentially a causal intervention towards the SCM of many-shot learning: the upper bound of FSL in a causal view.

UDA: Existing Unsupervised Domain Adaptation (UDA) literature adopts the covariate shift and conditional shift assumptions, which essentially encourage models to learn common features across domains. However, due to the lack of supervision in the target domain, they suffer from semantic loss: the feature will inevitably lose non-discriminative semantics in the source domain, which is however discriminative in the target domain. We use a causal view—transportability theory [49]—to identify that such loss is in fact a confounding effect, which can only be removed by causal intervention. However, the theoretical solution provided by transportability is far from practical for UDA, because it requires the stratification and representation of the unobserved confounder that is the cause of the domain gap. To this end, we propose a practical solution: Transporting Causal Mechanisms (TCM), to identify the confounder stratum and representations by using the domain-invariant disentangled causal mechanisms, which are discovered in an unsupervised fashion.

- In Chapter 4, we give a short preliminary on invariant learning, followed by two case studies:

UDA: Domain Adaptation (DA) is always challenged by the spurious correlation between the domain-invariant features (*e.g.*, class identity) and the domain-specific ones (*e.g.*, environment) that does not generalize to the target domain. Unfortunately, even enriched with additional unsupervised target domains, existing Unsupervised DA (UDA) methods still suffer from it. This is because the source domain supervision only considers the target domain samples as auxiliary data (*e.g.*, by pseudo-labeling), yet the inherent distribution in the target domain—where the valuable de-correlation clues hide—is disregarded. We propose to make the U in UDA matter by giving equal status to the two domains. Specifically, we learn an invariant classifier whose prediction is simultaneously consistent with the labels in the source

domain and clusters in the target domain, hence the spurious correlation inconsistent in the target domain is removed.

SSL: Pseudo-labeling is the prevailing method in semi-supervised semantic segmentation. However, it suffers from the well-known confirmation bias, where the model overfits inaccurate pseudo-labels. The reason is two-fold: 1) The labeled images tend to follow a different distribution from the unlabeled ones due to labeling preference. 2) The model is inevitably biased to the contextual cues specific to the labeled images (*e.g.*, “sofa” is brown), and generates noisy pseudo-labels (*e.g.*, misclassifying white “sofa”) that reinforce the bias. Based on the fundamental clustering assumption, we point out that the pixel-level feature clusters of the unlabeled images contain valuable debiasing hints, which become ever-lost in pseudo-labeling. In particular, a classifier that captures the labeled-specific context bias breaks up the unlabeled-specific clusters. Hence by its contra-position, we propose to train a model that simultaneously separates the labeled classes and unlabeled clusters, which removes the context bias.

- In Chapter 5, we formally define disentangled representation. Then we present practical approaches to learn it, in both supervised setting and unsupervised one:

Supervised Learning: We present a novel counterfactual framework for both Zero-Shot Learning (ZSL) and Open-Set Recognition (OSR), whose common challenge is generalizing to the unseen-classes by only training on the seen-classes. Our idea stems from the observation that the generated samples for unseen-classes are often out of the true distribution, which causes severe recognition rate imbalance between the seen-class (high) and unseen-class (low). We show that the key reason is that the generation is not **Counterfactually Faithful**, and thus we propose a faithful one, whose generation is from the sample-specific counterfactual question: What would the sample look like, if we set its class attribute to a certain class, while keeping its sample attribute unchanged? Thanks to the faithfulness, we can apply the **Consistency Rule** to perform unseen/seen binary classification, by asking: Would its counterfactual still look like itself? If “yes”, the sample is from a certain class, and “no” otherwise. Through extensive experiments on ZSL and OSR, we demonstrate that our framework effectively mitigates the seen/unseen imbalance and hence significantly improves the overall performance. Note that

this framework is orthogonal to existing methods, thus, it can serve as a new baseline to evaluate how ZSL/OSR models generalize.

Unsupervised Learning: Representation learning is all about discovering the hidden modular attributes that generate the data faithfully. We explore the potential of Denoising Diffusion Probabilistic Model (DM) in unsupervised learning of the modular attributes. We build a theoretical framework that connects the diffusion time-steps and the hidden attributes, which serves as an effective inductive bias for unsupervised learning. Specifically, the forward diffusion process incrementally adds Gaussian noise to samples at each time-step, which essentially collapses different samples into similar ones by losing attributes, *e.g.*, fine-grained attributes (*e.g.*, texture) are lost with less noise added (*i.e.*, early time-steps), while coarse-grained ones such as shape are lost by adding more noise (*i.e.*, late time-steps). To disentangle the modular attributes, at each time-step t , we learn a t -specific feature to compensate for the newly lost attribute, and the set of all $\{1, \dots, t\}$ -specific features, corresponding to the cumulative set of lost attributes, are trained to make up for the reconstruction error of a pre-trained DM at time-step t .

Chapter 2

Literature Review

2.1 Causal Inference

Causal inference [40] was recently introduced to machine learning [50, 51] and has been applied to various fields in computer vision. [52] proposes a retrospective for image captioning and other applications include image classification [53, 54], imitation learning [55], long-tailed recognition [42] and semantic segmentation [56]. We would like to highlight that data-augmentation based methods can also be considered as approximated intervention. These methods learn to generate additional support samples with image deformation [57, 58] or generative models [59, 60]. This can be view as physical interventions on the image features. Regarding the causal relation between image X and label Y , some works adopted anti-causal learning [37], *i.e.*, $Y \rightarrow X$, where the assumption is that labels Y are disentangled enough to be treated as Independent Mechanism (IM) [61, 62], which generates observed images X through $Y \rightarrow X$. However, our work targets at the more general case where labels can be entangled (*e.g.* “lion” and “dog” share the semantic “soft fur”) and the IM assumption may not hold. Therefore, we use causal prediction $X \rightarrow Y$ as it is essentially a reasoning process, where the IM is captured by D , which is engineered to be disentangled through CNN (*e.g.*, the conv-operations are applied independently). In this way, D generates visual features through $D \rightarrow X$ and emulates human’s naming process through $D \rightarrow Y$ (*e.g.*, “fur”, “four-legged” $\rightarrow$ “meerkat”). In fact, the causal direction $X \rightarrow Y$ (NOT anti-causal $Y \rightarrow X$) has been empirically justified in complex CV tasks [42, 63–65].

2.2 Few-Shot Learning

Few-Shot Learning. FSL has a wide spectrum of methods, including fine-tuning [35, 66], optimizing model initialization [67, 68], generating model parameters [69, 70], learning a feature space for a better separation of sample categories [71, 72], feature transfer [73, 74], and transductive learning that additionally uses query set data [35, 75, 76]. Thanks to them, the classification accuracy has been drastically increased [72, 75, 77, 78]. Recent works explore more powerful multi-modal CLIP as the pre-trained model, and use the few-shot training examples to fine-tune either CLIP visual embedding [79] or text embedding [80]. Our work offers an answer from a causal standpoint by showing that pre-training is a confounder. We not only further improve the accuracy of various FSL methods, but also explain the reason behind the improvements. In fact, the perspective offered by our work can benefit all the tasks that involve pre-training—any downstream task can be seen as FSL compared to the large-scale pre-training data.

Negative Transfer. The above phenomenon is also known as the negative transfer, where learning in source domain contributes negatively to the performance in target domain [81]. Many research works have been focused on when and how to conduct this transfer learning [82–84]. Yosinski *et al.* [85] split ImageNet according to man-made objects and natural objects as a test bed for feature transferability. Other work also revealed that using deeper backbone might lead to degraded performance when the domain gap between training and test is large [86]. Some similar findings are reported in the few-shot setting [87] and NLP tasks [88]. Unfortunately, they didn’t provide a theoretical explanation why it happens.

2.3 Unsupervised Domain Adaptation

Existing UDA works fall into the three categories: **1) Sample Reweighting** [89, 90]. This line-up adopts the covariate shift assumption. We denote the source domain as $S = s$ and target domain as $S = t$. It first models $P(X|S = s)$ and $P(X|S = t)$. When minimizing the classification loss, each sample in $S = s$ is assigned an importance weight $P(X|S = t)/P(X|S = s)$, *i.e.*, a target-like sample has a larger weight, which effectively encourages $P(X|S = s) = P(X|S = t)$. **2) Domain Mapping** [91, 92]. This approach focuses on the conditional

shift assumption. It first learns a mapping function that transforms samples from $S = s$ to $S = t$ through unpaired image-to-image translation techniques such as CycleGAN [8]. Then, the transformed source domain samples are used to train a classifier. **3) Invariant Feature Learning.** This is the most popular approach in recent literature [93–95]. It maps the samples in $S = s$ and $S = t$ to a common feature space, denoted as domain $S = c$, where the source and target domain samples are indistinguishable. Some works [36, 96] adopted the covariate shift assumption, and aim to minimize the differences in $P(X|S = c)$ using distance measures like Maximum Mean Discrepancy [97, 98] or using adversarial training to fool a domain discriminator [93, 99]. Others used the conditional shift assumption and aim to align $P(Y|X, S = c)$ [94, 100]. As the target domain samples have no labels, their $P(Y|X, S = c)$ is either estimated through clustering [101, 102] or using a classifier trained with the source domain samples [103, 104]. **4) Self-training** methods [105–107] are the mainstream in semi-supervised learning (SSL), and recently became a promising alternative in UDA, which centers on generating accurate pseudo-labels of samples in T with a classifier trained in S . Starting from a simple classifier design based on k-NN [108], this line of works later explored more complex designs, such as an ensemble of classifiers [109], a teacher-student network [110] or an exponential moving average model [111]. Yet the pseudo-labels are error-prone due to the spurious correlations in S , hence the model performance is still inconsistent [12].

Alleviate Semantic Loss. There are existing works that attempt to do so by learning to capture U in unsupervised fashion: One line exploits that D generates X (via $D \rightarrow X$) in the SCM and learns a latent variable to reconstruct X [99, 112]. The other line [113, 114] aims to learn a domain-specific representation for each $S = s, S = t$. However, this leads to an ever-illusory trade-off between the adversarial discrimination and reconstruction objectives. Our approach in Chapter 3.3 aims to fundamentally eliminate the semantic loss by providing a practical implementation of the transportability theory [49] based on causal intervention [115].

Alleviating Spurious Correlation has been studied in several existing works of UDA [12, 116, 117]. They train a target domain specific classifier to generate pseudo-labels [116, 117], or to regularize the learned features such that the target domain classifier performs well in $S = s$ [12]. However, their classifiers are still initialized based on a classifier trained in $S = s$, hence the spurious correlation in

$S = s$ are introduced to the classifiers right in the beginning. On the contrary, our method in Chapter 4.2 learns a cluster head in $S = t$ without supervision in $S = s$, which is solely for discovering the sample clusters in $S = t$. Hence the spurious correlation in $S = s$ inconsistent with the clusters in $S = t$ can be removed by our method (Section 4.2).

2.4 Semi-Supervised Learning

Semi-Supervised Learning (SSL). Semi-Supervised Learning (SSL) has been extensively studied to utilize unlabeled data to improve models trained with limited labeled data. The cluster assumption [118, 119], which posits that high-density regions in the input space are more likely to share the same label, forms the foundation of many SSL frameworks, such as label propagation, co-training, and self-training [120–122]. Prevailing SSL methods adopt the self-training paradigm but differ in the regularization of unlabeled data. Pseudo-labeling methods [123–125] predict labels for unlabeled data using the model trained on labeled data and add them to the training data for re-training, while consistency-regularization methods [122, 126, 127] apply a random perturbation to an unlabeled image and penalize the model’s inconsistency between the original and augmented predictions. Recently, state-of-the-art SSL methods like FixMatch [121] combines these two techniques to predict improved pseudo-labels and achieve better performance on various benchmarks.

Semi-Supervised Semantic Segmentation. In the context of semantic segmentation, acquiring pixel-level annotations on large-scale datasets is prohibitively expensive [128, 129], making SSL particularly relevant. Early methods [130, 131] in semi-supervised semantic segmentation incorporate Generative Adversarial Networks (GANs) [132] as an auxiliary supervision signal on unlabeled data. However, these methods suffer from unstable training [133], and often produce low-quality segmentation results. Recent methods [134–136] strive for simpler training paradigms, and pseudo-labeling [123] becomes prominent. However, pseudo-labeling methods rely heavily on high-quality pseudo-labels, yet they are often noisy and biased toward the labeled data [121, 137]. This leads to negative transfer [138], which may even hurt the performance. To address this, regularization techniques leveraging strong image augmentations—following FixMatch [121] in SSL—have

been adopted to improve the pseudo-labeling robustness. Strong perturbations are often applied on image-level [135, 139–141] or feature-level [13, 142] for improved diversity of the augmented views. Yet we show in Section 4.3 that context bias in the labeled data can lead to the failure of these techniques, and we propose a debiasing regularization to address this.

Debiasing in Semi-Supervised Learning. Most works tackle the bias implicitly by improving the tolerance to inaccurate pseudo labels, *e.g.*, generating pseudo-labels by averaging the previous predictions [143] or from an exponential moving average model [144]. Some works specifically target the class imbalance in SSL [135, 145, 146], which aim to improve the under-performing classes, *e.g.*, by re-weighting [147, 148]. Recent work [149] aims to remove the bias in SSL explicitly, which minimizes an estimated worst-case training bias through a minimax game. However, minimax optimization can be challenging, especially for the hard task of semantic segmentation that involves dense classification. Furthermore, it is difficult to estimate the ever-elusive training bias accurately. In contrast, our proposed method in Chapter 4.3 stems from the fundamental clustering assumption in SSL, and is based on the simple Binary Cross-Entropy (BCE) loss that is easy to optimize (Section 4.3).

2.5 Zero-Shot and Open-Set Recognition

Zero-Shot Learning (ZSL) is usually provided with an auxiliary set of class attributes to describe each seen- and unseen-class [29, 150]. Therefore, ZSL can be approached by either inferring a sample’s attribute and finding the closest match in the attribute space [33, 151–153], or generating features using the attributes and matching in the feature space [17, 154–156]. **Open-Set Recognition (OSR)** is a more challenging open environment setting with no information on the unseen-classes [157, 158], and the goal is to build a classifier for seen-classes that additionally rejects unseen-class samples as outliers. The inference methods in OSR calibrate a discriminative model by adjusting the classification logits [30, 159], and the generation methods estimate the seen-class density and reject test samples in the low-density area as outliers [18, 48]. **Out-Of-Distribution (OOD)** detection [43, 160, 161] also focuses on unseen detection, where seen and unseen samples

are usually from different domains [162]. However, in OOD, the unseen-class information is available. In particular, some works employ OOD techniques in ZSL to distinguish seen- and unseen-classes [163, 164]. Yet existing methods are still easily biased towards the idiosyncrasies of seen-classes. Our proposed approach in Chapter 5.2 fundamentally resolve this issue by disentangling class attributes of both seen- and unseen-classes.

2.6 Disentangled Representation Learning

This notion dates back to [165], and henceforward becomes a high-level goal of separating the factors of variations in the data [166–169]. Several works aim to provide a more precise description [170–172] by adopting an information-theoretic view [170, 173] and measuring the properties of a disentangled representation explicitly [171, 172]. We adopt the recent group-theoretic definition from Higgins *et al.* [38], which not only unifies the existing, but also resolves the previous controversial points [47, 174]. Although supervised learning of disentangled representation is a well-studied field [175–179], unsupervised disentanglement based on GAN [173, 180–182] or VAE [183–186] is still believed to be theoretically challenging [47]. We propose to leverage the inductive bias from pre-trained diffusion model to learn a disentangled representation in Chapter 5.3.

Recent works explore Diffusion Models (DMs) for representation learning. There are three main approaches: 1) DDIM [187] inversion aims to find a latent $\mathbf{x}_T$ as a feature, by DDIM sampling with the learned DM, can reconstruct the image $\mathbf{x}_0$. However, this process is time-consuming, and the resulting latent is difficult to interpret [188]. 2) [189] uses the bottleneck feature of the U-Net at all time-steps. However, the feature is not compact and difficult for downstream leverage. 3) The closest to our works are [21, 190]. However, they learn a time-step-independent $\mathbf{z}$ that entangles all the attributes. In contrast, we learn a compact, time-step-aware feature that disentangles the hidden generative attributes in Chapter 5.3.

Chapter 3

De-confound by Backdoor Adjustment ¹

Recall that $P(Y|do(X))$ capture the causal effect from sample X to prediction Y free from the spurious correlations in the training data D for the Structural Causal Model (SCM) in Figure 3.1. In Section 3.1, we start by deriving the backdoor adjustment for the SCM using the three rules of *do*-

calculus [191], which shows that when D is the only confounder between X and Y , the spurious correlation can be mitigated when D is observed. Then in Section 3.2 and Section 3.3, we show how Few-Shot Learning (FSL) and Unsupervised Domain Adaptation (UDA) can be formulated by this causal graph, and devise practical implementation of backdoor adjustment in each case.

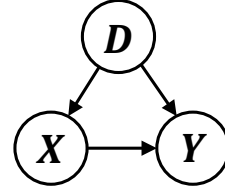


FIGURE 3.1: SCM for machine generalization.

¹The work in this chapter has been published in the following papers: **Zhongqi Yue**, Hanwang Zhang, Qianru Sun, Xian-Sheng Hua. Interventional Few-Shot Learning. 2020. *Conference on Neural Information Processing Systems (NeurIPS)*. **Zhongqi Yue**, Qianru Sun, Xian-Sheng Hua, Hanwang Zhang. Transporting Causal Mechanisms for Unsupervised Domain Adaptation. 2021. *International Conference on Computer Vision (ICCV)*.

3.1 Preliminaries

For a causal directed acyclic graph $\mathcal{G}$, let X, Y, Z and W be arbitrary disjoint sets of nodes. We use $\mathcal{G}_{\overline{X}}$ to denote the manipulated graph where all incoming arrows to node X are deleted. Similarly $\mathcal{G}_{\underline{X}}$ represents the graph where outgoing arrows from node X are deleted. We use lower case x, y, z and w for specific values taken by each set of nodes: $X = x, Y = y, Z = z$ and $W = w$. For any interventional distribution compatible with $\mathcal{G}$, we have the following three rules [40]:

Rule 1 Insertion/deletion of observations:

$$P(y|do(x), z, w) = P(y|do(x), w), \text{ if } (Y \perp\!\!\!\perp Z | X, W)_{\mathcal{G}_{\overline{X}}} \quad (3.1)$$

Rule 2 Action/observation exchange:

$$P(y|do(x), do(z), w) = P(y|do(x), z, w), \text{ if } (Y \perp\!\!\!\perp Z | X, W)_{\mathcal{G}_{\overline{XZ}}} \quad (3.2)$$

Rule 3 Insertion/deletion of actions:

$$P(y|do(x), do(z), w) = P(y|do(x), w), \text{ if } (Y \perp\!\!\!\perp Z | X, W)_{\mathcal{G}_{\overline{XZ(W)}}}, \quad (3.3)$$

where $Z(W)$ is the set of nodes in Z that are not ancestors of any W -node in $\mathcal{G}_{\overline{X}}$.

In our causal graph, the desired interventional distribution $P(Y|do(X = \mathbf{x}))$ can be derived by:

$$P(Y|do(\mathbf{x})) = \sum_d P(Y|do(X = \mathbf{x}), D = d)P(D = d|do(X = \mathbf{x})) \quad (3.4)$$

$$= \sum_d P(Y|do(X = \mathbf{x}), D = d)P(D = d) \quad (3.5)$$

$$= \sum_d P(Y|X = \mathbf{x}, D = d)P(D = d) \quad (3.6)$$

where Eq. (3.4) follow the law of total probability; Eq. (3.5) uses Rule 3 given $D \perp\!\!\!\perp X$ in $\mathcal{G}_{\overline{X}}$; Eq. (3.6) uses Rule 2 to change the intervention term to observation as $(Y \perp\!\!\!\perp X | D)$ in $\mathcal{G}_{\underline{X}}$. Eq. (3.6) is known as the backdoor adjustment. In this chapter, we discuss how to identify D to enable backdoor adjustment in Few-Shot Learning (FSL) and Unsupervised Domain Adaptation (UDA).

3.2 Case Study: Few-Shot Learning

3.2.1 Introduction

Few-Shot Learning (FSL) — the task of training a model using very few samples — is nothing short of a panacea for any scenario that requires fast model adaptation to new tasks [192], such as minimizing the need for expensive trials in reinforcement learning [193] and saving computation resource for light-weight neural networks [194, 195]. Although we knew that, more than a decade ago, the crux of FSL is to imitate the human ability of transferring prior knowledge to new tasks [196], not until the recent advances in pre-training techniques, had we yet reached a consensus on “what & how to transfer”: a powerful neural network Ω pre-trained on a large dataset $\mathcal{D}$. In fact, the prior knowledge learned from pre-training prospers today’s deep learning era, *e.g.*, $\mathcal{D}$ = ImageNet, Ω = ResNet in visual recognition [22, 197]; $\mathcal{D}$ = Wikipedia, Ω = BERT in natural language processing [198, 199].

In the context of pre-trained knowledge, we denote the original FSL training set as *support* set $\mathcal{S}$ and the test set as *query* set $\mathcal{Q}$, where the classes in $(\mathcal{S}, \mathcal{Q})$ are unseen (or new) in $\mathcal{D}$. Then, we can use Ω as a backbone (fixed or partially trainable) for extracting sample representations $\mathbf{x}$, and thus FSL can be achieved simply by *fine-tuning* the target model on $\mathcal{S}$ and test it on $\mathcal{Q}$ [35, 66]. However, the fine-tuning

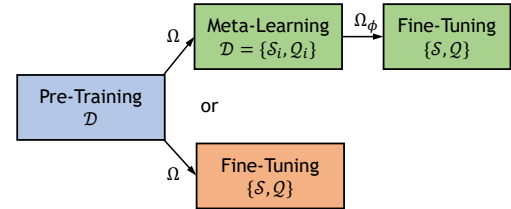


FIGURE 3.2: The relationships among different FSL paradigms (color green and orange). Our goal is to remove the deficiency introduced by Pre-Training.

only exploits the $\mathcal{D}$ ’s knowledge on “what to transfer”, but neglects “how to transfer”. Fortunately, the latter can be addressed by applying a post-pre-training and pre-fine-tuning strategy: *meta-learning* [200]. Different from fine-tuning whose goal is the “model” trained on $\mathcal{S}$ and tested on $\mathcal{Q}$, meta-learning aims to learn the “meta-model” — a learning behavior — trained on many learning episodes $\{(\mathcal{S}_i, \mathcal{Q}_i)\}$ sampled from $\mathcal{D}$ and tested on the target task $(\mathcal{S}, \mathcal{Q})$. In particular, the behavior can be parametrized by ϕ using model parameter generator [201, 202] or initialization [67]. After meta-learning, we denote Ω_ϕ as the new model starting

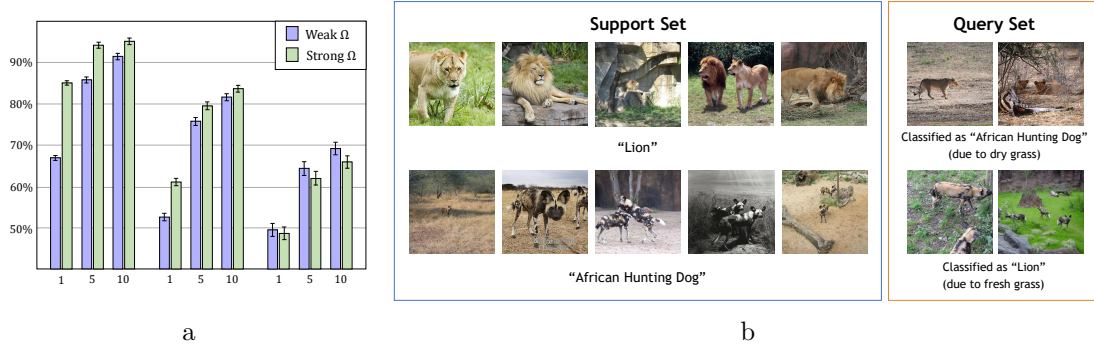


Figure 3.3: Quantitative and qualitative evidences of pre-trained knowledge misleading the fine-tune FSL paradigm. (a) *miniImageNet* fine-tuning accuracy on 1-/5-/10-shot FSL using weak and strong backbones: ResNet-10 and WRN-28-10. $\mathcal{S} \sim \mathcal{Q}$ (or $\mathcal{S} \not\sim \mathcal{Q}$) denotes the pre-trained classifier scores of the query is similar (or dissimilar) to that of the support set. “Average” is the mean of both. The dissimilarity is measured using query hardness defined in Section 5.1. (b) An example of 5-shot $\mathcal{S} \not\sim \mathcal{Q}$.

point for the subsequent fine-tuning on target task $(\mathcal{S}, \mathcal{Q})$. Figure 3.2 illustrates the relationships among the above discussed FSL paradigms.

It is arguably a common sense that the stronger the pre-trained Ω is, the better the downstream model will be. However, we surprisingly find that this may not be always the case in FSL. As shown in Figure 3.3a, we can see a paradox: though stronger Ω improves the performance on average, it indeed degrades that of samples in $\mathcal{Q}$ dissimilar to $\mathcal{S}$. To illustrate the “dissimilar”, we show a 5-shot learning example in Figure 3.3b, where the prior knowledge on “green grass” and “yellow grass” is misleading. For example, the “Lion” samples in $\mathcal{Q}$ have “yellow grass”, hence they are misclassified as “Dog” whose $\mathcal{S}$ has major “yellow grass”. If we use stronger Ω , the seen old knowledge (“grass” & “color”) will be more robust than the unseen new knowledge (“Lion” & “Dog”), and thus the old becomes even more misleading. We believe that such a paradox reveals an unknown systematic deficiency in FSL, which has been however hidden for years by our gold-standard “fair” accuracy, averaged over all the random $(\mathcal{S}, \mathcal{Q})$ test trials, regardless of the similarity between $\mathcal{S}$ and $\mathcal{Q}$ (*c.f.*, Figure 3.3a). Though Figure 3.3 only illustrates the fine-tune FSL paradigm, the deficiency is expected in the meta-learning paradigm, as fine-tune is also used in each meta-train episode (Figure 3.2). We will analyze them thoroughly in Section 3.2.4.

In this case study, we first point out that the cause of the deficiency: pre-training can do evil in FSL, and then propose a novel FSL paradigm: Interventional Few-Shot Learning (IFSL), to counter the evil. Our theory is based on the assumption

of the *causalities* among the pre-trained knowledge, few-shot samples, and class labels. Specifically, our contributions are summarized as follows.

- We begin with a Structural Causal Model (SCM) assumption in Section 3.2.2, which shows that the pre-trained knowledge is essentially a *confounder* that causes spurious correlations between the sample features and class labels in support set. As an intuitive example in Figure 3.3b, even though the “grass” feature is not the cause of the “Lion” label, the prior knowledge on “grass” still confounds the classifier to learn a correlation between them.
- In Section 3.2.2, we illustrate a causal justification of why the proposed IFSL fundamentally works better: it is essentially a causal approximation to many-shot learning. This motivates us to develop three effective implementations of IFSL using the backdoor adjustment [115] in Section 3.2.3.
- Thanks to the causal intervention, IFSL is naturally orthogonal to the downstream fine-tuning and meta-learning based FSL methods [67, 71, 75]. In Section 3.2.4, IFSL improves all baselines by a considerable margin, achieving new 1-/5-shot state-of-the-arts: 73.51%/83.21% on *miniImageNet* [71], 83.07%/88.69% on *tieredImageNet* [203], and 50.71%/64.43% on cross-domain CUB [16].
- We further diagnose the detailed performances of FSL methods across different similarities between $\mathcal{S}$ and $\mathcal{Q}$. We find that IFSL outperforms all baselines in every inch.

3.2.2 Problem Formulations

Few-Shot Learning

We are interested in a prototypical FSL: train a K -way classifier on an N -shot support set $\mathcal{S}$, where N is a small number of training samples per class (*e.g.*, $N=1$ or 5); then test the classifier on a query set $\mathcal{Q}$. As illustrated in Figure 3.2, we have the following two paradigms to train the classifier $P(y|\mathbf{x};\theta)$, predicting the class $y \in \{1, \dots, K\}$ of a sample $\mathbf{x}$:

Fine-Tuning. We consider the prior knowledge as the sample feature representation $\mathbf{x}$, encoded by the pre-trained network Ω on dataset $\mathcal{D}$. In particular, we refer

$\mathbf{x}$ to the output of the frozen sub-part of Ω and the rest trainable sub-part of Ω (if any) can be absorbed into θ . We train the classifier $P(y|\mathbf{x};\theta)$ on the support set $\mathcal{S}$, and then evaluate it on the query set $\mathcal{Q}$ in a standard supervised way.

Meta-Learning. Yet, Ω only carries prior knowledge in a way of “representation”. If the dataset $\mathcal{D}$ can be re-organized as the training episodes $\{(\mathcal{S}_i, \mathcal{Q}_i)\}$, each of which can be treated as a “sandbox” that has the same N -shot- K -way setting as the target $(\mathcal{S}, \mathcal{Q})$. Then, we can model the “learning behavior” from $\mathcal{D}$ parameterized as ϕ , which can be learned by the above fine-tuning paradigm for each $(\mathcal{S}_i, \mathcal{Q}_i)$. Formally, we denote $P_\phi(y|\mathbf{x};\theta)$ as the enhanced classifier equipped with the learned behavior. For example, ϕ can be the classifier weight generator [202], distance kernel function in k -NN [71], or even θ ’s initialization [67]. Considering $L_\phi(\mathcal{S}_i, \mathcal{Q}_i; \theta)$ as the loss function of $P_\phi(y|\mathbf{x};\theta)$ trained on $\mathcal{S}_i$ and tested on $\mathcal{Q}_i$, we can have $\phi \leftarrow \arg \min_{(\phi, \theta)} \mathbb{E}_i [L_\phi(\mathcal{S}_i, \mathcal{Q}_i; \theta)]$, and then we fix the optimized ϕ and fine-tune for θ on $\mathcal{S}$ and test on $\mathcal{Q}$.

Structural Causal Model

From the above discussion, we can see that (ϕ, θ) in meta-learning and θ in fine-tuning are both dependent on the pre-training. Such “dependency” can be formalized with a Structural Causal Model (SCM) [115] proposed in Figure 3.4a, where the nodes denote the abstract data variables and the directed edges denote the (functional) causality, *e.g.*, $X \rightarrow Y$ denotes that X is the cause and Y is the effect. Now we introduce the graph and the rationale behind its construction at a high-level. Please see Section 3.2.3 for the detailed functional implementations.

$D \rightarrow X$. We denote X as the feature representation and D as the pre-trained knowledge, *e.g.*, the dataset $\mathcal{D}$ and its induced model Ω . This link assumes that the feature X is extracted by using Ω .

$D \rightarrow C \leftarrow X$. We denote C as the transformed representation of X in the low-dimensional manifold, whose base is inherited from D . This assumption can be rationalized as follows. 1) $D \rightarrow C$: a set of data points are usually embedded in a low-dimensional manifold. This finding can be dated back to the long history of dimensionality reduction [204, 205]. Nowadays, there are theoretical [44, 206] and empirical [5, 207] evidences showing that disentangled semantic manifolds emerge during training deep networks. 2) $X \rightarrow C$: features can be represented using (or projected onto) the manifold base linearly [208, 209] or non-linearly [210]. In

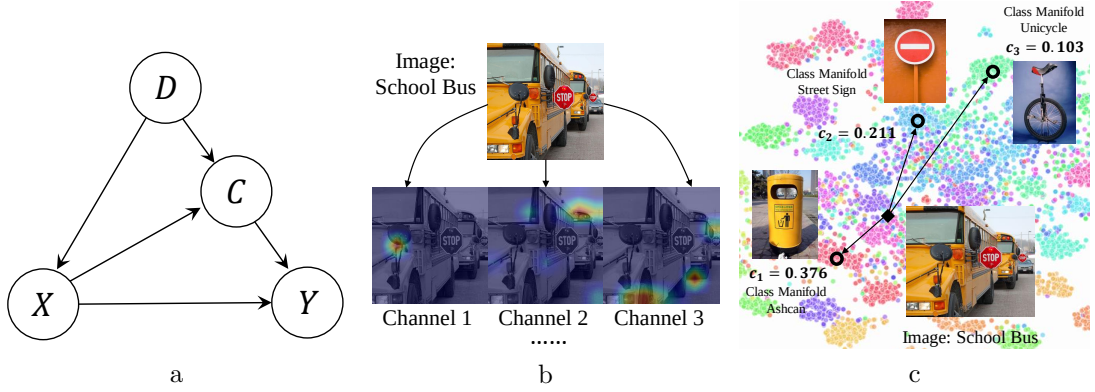


Figure 3.4: (a) Causal Graph for FSL; (b) Feature-wise illustration of $D \rightarrow C$: Feature channels of pre-trained network (*e.g.* $1 \dots 512$ for ResNet-10). $X \rightarrow C$: Per-channel response to an image (“school bus”) visualized by CAM[5]; (c) Class-wise illustration for $D \rightarrow C$: features are clustered according to the pre-training semantic classes (colored t-SNE plot[6]). $X \rightarrow C$: An image (“school bus”) can be represented in terms of the similarities among the base classes (“ashcan”, “unicycle”, “sign”).

particular, as later discussed in Section 3.2.3, we explicitly implement the base as feature dimensions (Figure 3.4b) and class-specific mean features (Figure 3.4c).

$\mathbf{X} \rightarrow \mathbf{Y} \leftarrow \mathbf{C}$. We denote Y as the classification effect (*e.g.*, logits), which is determined by X via two ways: 1) the direct $X \rightarrow Y$ and 2) the mediation $X \rightarrow C \rightarrow Y$. In particular, the first way can be removed if X can be fully represented by C (*e.g.*, feature-wise adjustment in Section 3.2.3). The second way is inevitable even if the classifier does not take C as an explicit input, because any X can be inherently represented by C . To illustrate, suppose that X is a linear combination of two base vectors plus a noise residual: $\mathbf{x} = c_1 \mathbf{b}_1 + c_2 \mathbf{b}_2 + \mathbf{e}$, any classifier $f(\mathbf{x}) = f(c_1 \mathbf{b}_1 + c_2 \mathbf{b}_2 + \mathbf{e})$ will implicitly exploit the C representation in terms of $\mathbf{b}_1$ and $\mathbf{b}_2$. In fact, this assumption also fundamentally validates unsupervised representation learning [211]. To see this, if $C \not\rightarrow Y$ in Figure 3.4a, uncovering the latent knowledge representation from $P(Y|X)$ would be impossible, because the only path left that transfers knowledge from D to Y : $D \rightarrow X \rightarrow Y$, is cut off by conditioning on X : $D \not\rightarrow X \rightarrow Y$.

An ideal FSL model should capture the true causality between X and Y to generalize to unseen samples. For example, as illustrated in Figure 3.3b, we expect that the “Lion” prediction is caused by the “lion” feature *per se*, but not the background “grass”. However, from the SCM in Figure 3.4a, the conventional correlation $P(Y|X)$ fails to do so, because the increased likelihood of Y given X is not

only due to “X causes Y” via $X \rightarrow Y$ and $X \rightarrow C \rightarrow Y$, but also the spurious correlation via 1) $D \rightarrow X$, *e.g.*, the “grass” knowledge generates the “grass” feature, and 2) $D \rightarrow C \rightarrow Y$, *e.g.*, the “grass” knowledge generates the “grass” semantic, which provides useful context for “Lion” label. Therefore, to pursue the true causality between X and Y , we need to use the *causal intervention* $P(Y|do(X))$ [40] instead of the likelihood $P(Y|X)$ for the FSL objective.

Causal Intervention via Backdoor Adjustment

By now, an astute reader may notice that the causal graph in Figure 3.4a is also valid for Many-Shot Learning (MSL), *i.e.*, conventional learning based on pre-training. Compared to FSL, the $P(Y|X)$ estimation of MSL is much more robust. For example, on *miniImageNet*, a 5-way-550-shot fine-tuned classifier can achieve 95% accuracy, while a 5-way-5-shot one only obtains 79%. We used to blame FSL for insufficient data by the law of large numbers in point estimation [212]. However, it does not answer why MSL converges to the true causal effects as the number of samples increases infinitely. In other words, why $P(Y|do(X)) \approx P(Y|X)$ in MSL while $P(Y|do(X)) \not\approx P(Y|X)$ in FSL?

To answer the question, we need to incorporate the endogenous feature sampling $\mathbf{x} \sim P(X|I)$ into the estimation of $P(Y|X)$, where I denotes the sample ID. We have $P(Y|X = \mathbf{x}_i) := \mathbb{E}_{\mathbf{x} \sim P(X|I)} P(Y|X = \mathbf{x}, I = i) = P(Y|I)$, *i.e.*, we can use $P(Y|I)$ to estimate $P(Y|X)$. In Figure 3.5a, the causal relation between I and X is purely $I \rightarrow X$, *i.e.*, $X \rightarrow I$ does not exist, because tracing the X ’s ID out of many-shot samples is like to find a needle in a haystack, given the nature that a DNN feature is an abstract and diversity-reduced representation of many samples [213]. However, as shown in Figure 3.5b, $X \rightarrow I$ persists in FSL, because it is much easier for a model to “guess” the correspondence, *e.g.*, the 1-shot extreme case that has a trivial 1-to-1 mapping for $X \leftrightarrow I$. Therefore, the key causal difference between MSL and FSL is: MSL essentially makes I an *instrumental variable* [214] that achieves $P(Y|X) := P(Y|I) \approx P(Y|do(X))$. Intuitively, we can see that all the causalities between I and D in MSL are all blocked by colliders¹, making I and D independent. So, the feature X is essentially “intervened” by I , no longer dictated by D , *e.g.*, neither “yellow grass” nor “green grass” dominates

¹In causal graph, the junction $A \rightarrow B \leftarrow C$ is called a “collider”, making A and C independent even though A and C are linked via B [115]. For example, A = “Quality”, C = “Luck”, and B = “Paper Acceptance”.

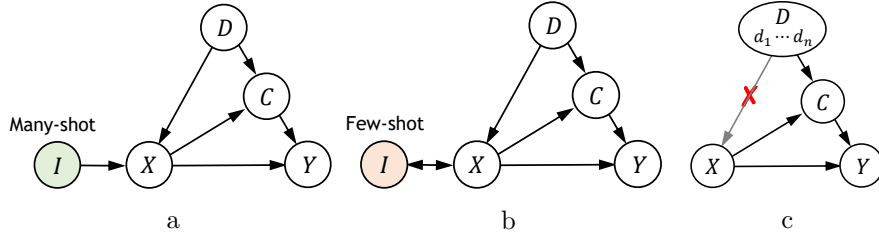


Figure 3.5: Causal graphs with sampling process. (a) Many-Shot Learning, where $P(Y|X) \approx P(Y|do(X))$; (b) Few-Shot Learning where $P(Y|X) \not\approx P(Y|do(X))$; (c) Interventional Few-Shot Learning where we directly model $P(Y|do(X))$.

“Lion” in Figure 3.3b, mimicking the casual intervention by controlling the use of pre-trained knowledge.

In this paper, we propose to use the backdoor adjustment [115] to achieve $P(Y|do(X))$ without the need for many-shot, which certainly undermines the definition of FSL. The backdoor adjustment assumes that we can observe and stratify the confounder, *i.e.*, $D = \{d\}$, where each d is a stratification of the pre-trained knowledge. Formally, the backdoor adjustment for the graph in Figure 3.4a is:

$$P(Y|do(X = \mathbf{x})) = \sum_d P(Y|X = \mathbf{x}, D = d, C = g(\mathbf{x}, d)) P(D = d), \quad (3.7)$$

where g is a function defined later. However, it is not trivial to instantiate d , especially when D is a 3rd-party delivered pre-trained network where the dataset is unobserved [215]. Next, we will offer three practical implementations of Eq. (3.7) for Interventional FSL.

3.2.3 Interventional Few-Shot Learning

Our implementation idea is inspired from the two inherent properties of any pre-trained DNN. First, each feature dimension carries a semantic meaning, *e.g.*, every channel in convolutional neural network is well-known to encode visual concepts [5, 207]. So, each feature dimension represents a piece of knowledge. Second, most prevailing pre-trained models use a classification task as the objective, such as the 1,000-way classifier of ResNet [22] and the token predictor of BERT [199]. Therefore, the classifier can be considered as the distilled knowledge, which has

been already widely adopted in literature [195]. Next, we will detail the proposed Interventional FSL (IFSL) by providing three *different* implementations² for $g(\mathbf{x}, d)$, $P(Y|X, D, C)$, and $P(D)$ in Eq. (1.1). In particular, the exact forms of $P(Y|\cdot)$ across different classifiers are given in Section 3.2.5.

Feature-wise Adjustment. Suppose that $\mathcal{F}$ is the index set of the feature dimensions of $\mathbf{x}$, *e.g.*, from the last-layer of the pre-trained network Ω . We divide $\mathcal{F}$ into n equal-size disjoint subsets, *e.g.*, the output feature dimension of ResNet-10 is 512, if $n = 8$, the i -th set will be a feature dimension index set of size $512/8 = 64$, *i.e.*, $\mathcal{F}_i = \{64(i-1) + 1, \dots, 64i\}$. The stratum set of pre-trained knowledge is defined as $D := \{d_1, \dots, d_n\}$, where each $d_i = \mathcal{F}_i$.

(i) $g(\mathbf{x}, d_i) := \{k | k \in \mathcal{F}_i \cap \mathcal{I}_t\}$, where $\mathcal{I}_t$ is an index set whose corresponding absolute values in $\mathbf{x}$ are larger than the threshold t . The reason is simple: if a feature dimension is inactive in $\mathbf{x}$, its corresponding adjustment can be omitted. We set $t=1e-3$ in this paper.

(ii) $P(Y|X, D, C) = P(Y|[\mathbf{x}]_c)$, where $c = g(\mathbf{x}, d_i)$ is implemented as the index set defined above, $[\mathbf{x}]_c = \{x_k\}_{k \in c}$ is a feature selector which selects the dimensions of $\mathbf{x}$ according to the index set c . The classifier takes the adjusted feature $[\mathbf{x}]_c$ as input. Note that d is already absorbed in c , so $[\mathbf{x}]_c$ is essentially a function of (X, D, C) .

(iii) $P(d_i) = 1/n$, where we assume a uniform prior for the adjusted features.

(iv) The overall feature-wise adjustment is:

$$P(Y|do(X = \mathbf{x})) = \frac{1}{n} \sum_{i=1}^n P(Y|[\mathbf{x}]_c), \quad \text{where } c = \{k | k \in \mathcal{F}_i \cap \mathcal{I}_t\}. \quad (3.8)$$

It is worth noting that the feature-wise adjustment is always applicable, as we can always have the feature representation $\mathbf{x}$ from the pre-trained network. Interestingly, our feature-wise adjustment sheds some light on the theoretical justifications for the multi-head trick in transformers [198]. We will explore this in future work.

Class-wise Adjustment. Suppose that there are m pre-training classes, denoted as $\mathcal{A} = \{a_1, \dots, a_m\}$. In class-wise adjustment, each stratum of pre-trained knowledge is defined as a pre-training class, *i.e.*, $D := \{d_1, \dots, d_m\}$ and each $d_i = a_i$.

²We assume that the combinations of the feature dimensions or classes are linear, otherwise the adjustment requires prohibitive $\mathcal{O}(2^n)$ sampling. We will relax this assumption in future work.

Algorithm 1: Fine-tuning + IFSL

Input : D , Support set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_s}$
Output : Fine-tuned classifier parameters θ
Initialize θ ;
while not converged **do**
 for $i = 1, \dots, n_s$ **do**
 for $d \in D$ **do**
 Calculate $c = g(\mathbf{x}, d)$;
 Obtain $P(Y|\mathbf{x}_i, c, d; \theta), P(d)$
 Prediction $\hat{y}_i = P(y|do(\mathbf{x}); \theta)$;
 Update θ using $\mathcal{L}(\hat{y}_i, y_i)$
return θ

Algorithm 2: Meta-Learning + IFSL

Input : D , training dataset $\mathcal{D}$
Output : Optimized meta-parameters ϕ
Initialize ϕ ;
while not converged **do**
 Sample $(\mathcal{S}_i, \mathcal{Q}_i)$ from $\mathcal{D}$;
 Initialize classifier θ with $\phi, \mathcal{S}_i$;
 Fine-tune θ using **Algorithm 1**
 conditioned on ϕ ;
 Predict query based on $P_\phi(y|do(\mathbf{x}); \theta)$;
 Update ϕ using $\mathcal{L}_\phi(\mathcal{S}_i, \mathcal{Q}_i; \theta)$
return ϕ

FIGURE 3.6: IFSL algorithms for fine-tuning and meta-learning.

- (i) $g(\mathbf{x}, d_i) := P(a_i|\mathbf{x})\bar{\mathbf{x}}_i$, where $P(a_i|\mathbf{x})$ is the pre-trained classifier's probability output that $\mathbf{x}$ belongs to class a_i , and $\bar{\mathbf{x}}_i$ is the mean feature of pre-training samples from class a_i . Note that unlike feature-wise adjustment where c is an index set, here $c = g(\mathbf{x}, d_i)$ is implemented as a real vector.
- (ii) $P(Y|X, D, C) = P(Y|\mathbf{x} \oplus g(\mathbf{x}, d_i))$, where $\oplus$ denotes vector concatenation.
- (iii) $P(d_i) = 1/m$, where we assume a uniform prior of each class.
- (iv) The overall class-wise adjustment is:

$$P(Y|do(X = \mathbf{x})) = \frac{1}{m} \sum_{i=1}^m P(Y|\mathbf{x} \oplus P(a_i|\mathbf{x})\bar{\mathbf{x}}_i) \approx P(Y|\mathbf{x} \oplus \frac{1}{m} \sum_{i=1}^m P(a_i|\mathbf{x})\bar{\mathbf{x}}_i), \quad (3.9)$$

where we adopt the Normalized Weighted Geometric Mean (NWGM) [52, 216] approximation to move the outer sum $\sum P$ into the inner $P(\sum)$. This greatly reduces the network forward-pass consumption as m is usually large in pre-training dataset.

Combined Adjustment. We can combine feature-wise and class-wise adjustment to make the stratification in backdoor adjustment much more fine-grained. Our combination is simple: applying feature-wise adjustment after class-wise adjustment. Thus, we have:

$$P(Y|do(X = \mathbf{x})) \approx \frac{1}{n} \sum_{i=1}^n P(Y|[\mathbf{x}]_c \oplus \frac{1}{m} \sum_{j=1}^m [P(a_j|\mathbf{x})\bar{\mathbf{x}}_j]_c), \text{ where } c = \{k | k \in \mathcal{F}_i \cap \mathcal{I}_t\}. \quad (3.10)$$

We show the IFSL algorithm for fine-tuning and meta-learning in Algorithm 1 and 2, respectively.

3.2.4 Experiments

Datasets and Settings

Datasets. We conducted experiments on benchmark datasets in FSL literature: 1) *miniImageNet* [71] containing 600 images per class over 100 classes. We followed the split proposed in [217]: 64/16/20 classes for train/val/test. 2) *tieredImageNet* [203] is much larger compared to *miniImageNet* with 608 classes and each class around 1,300 samples. These classes were grouped into 34 higher-level concepts and then partitioned into 20/6/8 disjoint sets for train/val/test to achieve larger domain difference between training and testing. 3) Caltech-UCSD Birds-200-2011 (**CUB**) [16] for cross-domain evaluation. It contains 200 classes and each class has around 60 samples. The models used for CUB test were trained on the *miniImageNet*.

Implementation Details. We pre-trained the 10-layer ResNet (ResNet-10) [22] and the WideResNet (WRN-28-10) [218] as our backbones. Our proposed IFSL supports both fine-tuning and meta-learning. For fine-tuning, we applied average pooling on the last residual block and used the pooled features to train classifiers. For meta-learning, we deployed 5 representative methods that cover a large spectrum of meta-learning based FSL: 1) model initialization: MAML [67], 2) weight generator: LEO [69], transductive learning: SIB [75], 4) metric learning: MatchingNet (MN) [71], and 5) feature transfer: MTL [73]. For both fine-tuning and meta-learning, our IFSL aims to learn classifier $P(Y|do(X))$ instead of the conventional $P(Y|X)$. Detailed implementations are given in Appendix 5.

Evaluation Metrics. Our evaluation is based on the following metrics: 1) Conventional accuracy (**Acc**) is the average classification accuracy commonly used in FSL [67, 71, 73]. 2) **Hardness-specific Acc.** For each query, we define a hardness that measures its semantic dissimilarity to the support set, and accuracy is then computed at different levels of query hardness. Specifically, query hardness is computed by $h = \log((1 - s)/s)$ and $s = \exp\langle \mathbf{r}^+, \mathbf{p}_{c=gt}^+ \rangle / \sum_c \exp\langle \mathbf{r}^+, \mathbf{p}_c^+ \rangle$, where $\langle \cdot \rangle$ is the cosine similarity, $(\cdot)^+$ represents the ReLU activation function, $\mathbf{r}$ denotes

Table 3.1: Acc (%) averaged over 2000 5-way FSL tasks before and after applying IFSL. We obtained the results by using official code and our backbones for a fair comparison across methods. We also implemented SIB in both transductive and inductive setting to facilitate fair comparison. For IFSL, we reported results of combined adjustment as it almost always outperformed feature-wise and class-wise adjustment. See Section 3.2.5 for Acc and 95% confidence intervals on all 3 types of adjustment.

		ResNet-10				WRN-28-10				
Method		miniImageNet		tieredImageNet		miniImageNet		tieredImageNet		
		5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	
Fine-Tuning	Linear		76.38	56.26	81.01	61.39	79.79	60.69	85.37	67.27
		+IFSL+2.19	77.97+1.59	60.13+3.87	82.08+1.07	64.29+2.9	80.97+1.18	64.12+3.43	86.19+0.82	69.96+2.69
	Cosine		76.68	56.40	81.13	62.08	79.72	60.83	85.41	67.30
		+IFSL+1.77	77.63+0.95	59.84+3.44	81.75+0.62	64.47+2.39	80.74+1.02	63.76+2.93	86.13+0.72	69.36+2.06
	k -NN		76.63	55.92	80.85	61.16	79.60	60.34	84.67	67.25
		+IFSL+3.13	78.42+1.79	62.31+6.36	81.98+1.13	65.71+4.55	81.08+1.48	64.98+4.64	86.06+1.39	70.94+3.69
Meta-Learning	MAML [67]		70.85	56.59	74.02	59.17	73.92	58.02	77.20	61.40
		+IFSL+5.55	76.37+5.52	59.36+2.77	81.04+7.02	63.88+4.71	79.25+5.33	62.84+4.82	85.10+7.90	67.70+6.30
	LEO [69]		74.49	58.48	80.25	65.25	75.86	59.77	82.15	68.90
		+IFSL+1.94	76.91+2.42	61.09+2.61	81.43+1.18	66.03+0.78	77.72+1.86	62.19+2.42	85.04+2.89	70.28+1.38
	MTL [73]		75.65	58.49	81.14	64.29	77.30	62.99	83.23	70.08
		+IFSL+2.02	78.03+2.38	61.17+2.68	82.35+1.21	65.72+1.43	80.20+2.9	64.40+1.41	86.02+2.79	71.45+1.37
	MN [71]		75.21	61.05	79.92	66.01	77.15	63.45	82.43	70.38
		+IFSL+1.34	76.73+1.52	62.64+1.59	80.79+0.87	67.30+1.29	78.55+1.40	64.89+1.44	84.03+1.60	71.41+1.03
	SIB [75] (transductive)		78.88	67.10	85.09	77.64	81.73	71.31	88.19	81.97
		+IFSL+1.15	80.32+1.44	68.85+1.75	85.43+0.34	78.03+0.39	83.21+1.48	73.51+2.20	88.69+0.50	83.07+1.10
	SIB [75] (inductive)		75.64	57.20	81.69	65.51	78.17	60.12	84.96	69.20
		+IFSL+2.05	77.68+2.04	60.33+3.13	82.75+1.06	67.34+1.83	80.05+1.88	63.14+3.02	86.14+1.18	71.45+2.25

the Ω prediction logits of query, $\mathbf{p}_c$ denotes the average prediction logits of class c in the support set and gt is the ground-truth of query. Using **Hardness-specific Acc** is similar to evaluating the hardness of FSL tasks [35], while ours is query-sample-specific and hence is more fine-grained. Later, we will show its effectiveness to unveil the spurious effects in FSL. 3) Feature localization accuracy (**CAM-Acc**) quantifies if a model “pays attention” to the actual object when making prediction. It is defined as the percentage of pixels inside the object bounding box by using Grad-CAM [4] score larger than 0.9. Compared to **Acc** that shows if the prediction is correct, **CAM-Acc** reveals whether the prediction is based on the correct visual cues.

Results and Analysis

Conventional Acc. 1) From Table 3.1, we observe that IFSL consistently improves fine-tuning and meta-learning in all settings, which suggests that IFSL is agnostic to methods, datasets, and backbones. 2) In particular, the improvements are typically larger on 1-shot than 5-shot. For example, in fine-tuning, the average performance gain is 1.15% on 5-shot and 3.58% on 1-shot. The results support our analysis in Section 3.2.2 that FSL models are more prone to bias in lower-shot

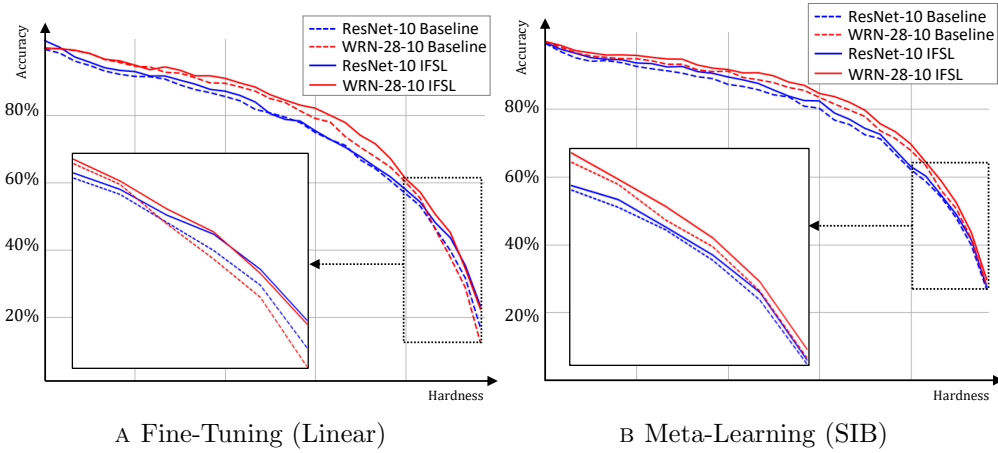


FIGURE 3.7: Accuracy across query hardness on 5-shot fine-tuning and meta-learning.

settings. 3) Regarding the average improvements on fine-tuning vs. meta-learning (*e.g.* k -NN and MN), we observe that IFSL improves more on fine-tuning in most cases. We conjecture that this is because meta-learning is an implicit form of intervention, where randomly sampled meta-training episodes effectively stratify the pre-trained knowledge. This suggests that meta-learning is fundamentally superior over fine-tuning due to increased robustness against confounders. We will investigate this potential theory in future work. 4) Additionally we see that the improvements on *miniImageNet* are usually larger than that on *tieredImageNet*. A possible reason is the much larger training set for *tieredImageNet*: it substantially increases the breadth of the pre-trained knowledge and the resulting models explain query samples much better. 5) According to Table 3.1 and Table 3.2, it is clear that our k -NN+IFSL outperforms IdeMe-Net [57] using the same pre-trained ResNet-10. This shows that using data augmentation — a method of physical data intervention as in IdeMe-Net [57] is inferior to our causal intervention in IFSL. 6) Overall, our IFSL achieves the new state-of-the-art on both datasets. Note that IFSL is flexible to be plugged into different baselines.

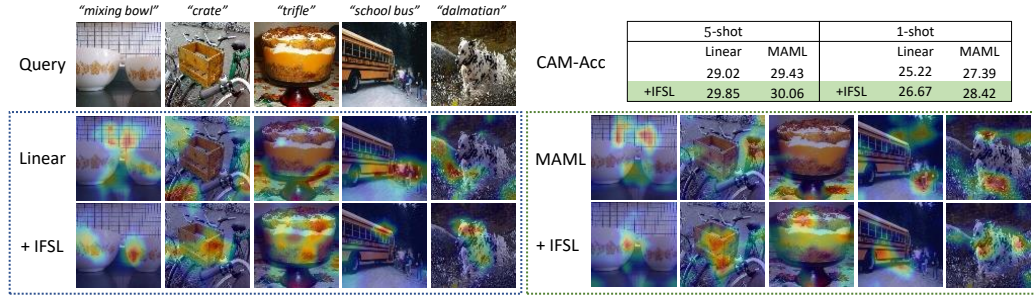
Hardness-specific Acc. 1) Figure 3.6a shows the plot of Hardness-specific Acc of fine-tuning. We notice that when query becomes harder, ResNet-10 (blue curves) becomes superior to WRN-28-10 (red curves). This tendency is consistent with Figure 3.3a illustrating the effect of the confounding bias caused by pre-training. 2) Intriguingly, in Figure 3.6b, we notice that this tendency is reversed for meta-learning, *i.e.*, deeper backbone always performs better. The improved performance of deeper backbone on hard queries suggests that meta-learning should have some

Table 3.2: Comparison with state-of-the-arts of 5-way 1-/5-shot Acc (%) on *miniImageNet* and *tieredImageNet*.

Method	Backbone	<i>miniImageNet</i>		<i>tieredImageNet</i>	
		5-shot	1-shot	5-shot	1-shot
Baseline++ [66]	ResNet-10	75.90	53.97	-	-
IdeMe-Net [†] [57]	ResNet-10	73.78	57.61	80.34	60.32
TRAML [219]	ResNet-12	79.54	67.10	-	-
DeepEMD [72]	ResNet-12	82.41	65.91	86.03	71.16
CTM [220]	ResNet-18	80.51	64.12	84.28	68.41
FEAT [77]	WRN-28-10	81.80	66.69	84.38	70.41
Tran. Baseline [35]	WRN-28-10	78.40	65.73	85.50	73.34
wDAE-GNN [202]	WRN-28-10	78.85	62.96	83.09	68.18
SIB [†] [75]	WRN-28-10	81.73	71.31	88.19	81.97
SIB+IFSL (ours)	WRN-28-10	83.21	73.51	88.69	83.07

[†]Using our pre-trained backbone.Table 3.3: Results of cross-domain evaluation: *miniImageNet* $\rightarrow$ CUB.

Backbone	Method	5-shot	1-shot
ResNet-10	Linear	58.84	42.25
		+IFSL	60.65
	SIB	60.60	45.87
		+IFSL	62.07
WRN-28-10	Linear	62.12	42.89
		+IFSL	64.15
	SIB	62.59	49.16
		+IFSL	64.43

Figure 3.8: Some *miniImageNet* visualizations of Grad-Cam [4] activation of query images and the CAM-Acc (%) table of using linear classifier and MAML. Categories with red text represent failed cases.

functions to remove the confounding bias. This evidence will inspire us to provide a causal view of meta-learning in future work. 3) Overall, Figure 3.7 shows that using IFSL further improves fine-tuning and meta-learning consistently across all hardness, validating the effectiveness of the proposed causal intervention.

CAM-Acc & Visualization. In Figure 3.8, we compare +IFSL to baseline linear classifier on the left and to baseline MAML [67] on the right, and summarize CAM-Acc results in the upper-right table. From the visualization, we see that using IFSL let the model pay more attention to the objects. However, notice that all models failed in the categories colored as red. A possible reason behind the failures is the extremely small size of the object — models have to resort to context for prediction. From the numbers, we can see our improvements for 1-shot are larger than that for

5-shot, consistent with our findings using other evaluation metrics. These results suggest that IFSL helps models use the correct visual semantics for prediction by removing the confounding bias.

Cross-Domain Generalization Ability. In Table 3.3, we show the testing results on CUB using the models trained on the *miniImageNet*. The setting is challenging due to the big domain gap between the two datasets. We chose linear classifier as it outperforms cosine and k -NN in cross-domain setting and compared with transductive method — SIB. The results clearly show that IFSL works well in this setting and brings consistent improvements, with the average 1.94% of Acc. In addition, we can see that applying IFSL brings larger improvements to the inductive linear classifier than to the transductive SIB. It is possibly because transductive methods involve unlabeled query data and performs better than inductive methods with the additional information. Nonetheless we observe that IFSL can further improve SIB in cross-domain (Table 3.3) and single-domain (Table 3.1) generalization.

3.2.5 Supplementary Information

Additional Results

We include additional results on **Conventional Acc** in Table 3.4 supplementary to Table 3.1.

Implementation Details

Pre-training. Prior to fine-tuning or meta-learning, we pre-trained a deep neural network (DNN) as feature extractor on the train split of a dataset. We use ResNet-10[22] or WRN-28-10[218] as feature extractor backbone. The networks are trained from scratch with stochastic gradient descent in a fully-supervised manner, *i.e.*, minimizing cross-entropy loss on the train split of a dataset. Specifically the training is conducted on 90 epochs with early stopping using validation accuracy. We used batch size of 256 and image size of 84×84 . For data augmentation, a random patch is sampled from an image, resized to 84×84 and randomly flipped along horizontal axis before used for training. The initial learning rate is set to 0.1 and it is scaled down by factor of 10 every 30 epochs.

TABLE 3.4: Supplementary to Table 3.1 Acc (%) and 95% confidence intervals averaged over 2000 5-way FSL tasks before and after applying three proposed implementations of adjustment. Specifically, “*ft*” refers to feature-wise adjustment, “*cl*” refers to class-wise adjustment and “*ft+cl*” refers to combined adjustment.

		ResNet-10				WRN-28-10			
Method		miniImageNet		tieredImageNet		miniImageNet		tieredImageNet	
		5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot
Fine-Tuning	Linear	<i>ft</i>	76.38 ± 0.36	56.26 ± 0.47	81.01 ± 0.38	61.39 ± 0.47	79.79 ± 0.33	60.69 ± 0.45	85.37 ± 0.34
		<i>cl</i>	76.84 ± 0.36	57.37 ± 0.43	81.45 ± 0.38	61.88 ± 0.47	80.22 ± 0.31	60.84 ± 0.45	85.70 ± 0.33
		<i>ft+cl</i>	77.23 ± 0.34	59.45 ± 0.45	81.33 ± 0.38	62.60 ± 0.48	80.27 ± 0.32	62.15 ± 0.44	85.54 ± 0.33
	Cosine	<i>ft</i>	77.97 ± 0.34	60.13 ± 0.45	82.08 ± 0.37	64.29 ± 0.48	80.97 ± 0.31	64.12 ± 0.44	86.19 ± 0.34
		<i>cl</i>	76.68 ± 0.36	56.40 ± 0.46	81.13 ± 0.39	62.08 ± 0.47	79.72 ± 0.33	60.83 ± 0.46	85.41 ± 0.34
		<i>ft+cl</i>	76.83 ± 0.35	56.86 ± 0.44	81.34 ± 0.37	62.45 ± 0.47	79.80 ± 0.32	61.25 ± 0.44	85.74 ± 0.33
	<i>k</i> -NN	<i>ft</i>	76.99 ± 0.35	57.65 ± 0.45	81.42 ± 0.38	63.37 ± 0.48	79.96 ± 0.32	62.04 ± 0.45	85.77 ± 0.33
		<i>cl</i>	77.63 ± 0.34	59.84 ± 0.46	81.75 ± 0.38	64.47 ± 0.48	80.74 ± 0.32	63.76 ± 0.45	86.13 ± 0.33
		<i>ft+cl</i>	76.63 ± 0.36	55.92 ± 0.46	80.85 ± 0.39	61.16 ± 0.48	79.60 ± 0.32	60.34 ± 0.45	84.67 ± 0.34
	MAML [67]	<i>ft</i>	77.98 ± 0.34	60.71 ± 0.44	81.95 ± 0.36	65.66 ± 0.48	81.17 ± 0.31	64.87 ± 0.44	85.76 ± 0.34
		<i>cl</i>	78.36 ± 0.35	61.32 ± 0.45	81.93 ± 0.37	65.71 ± 0.48	80.61 ± 0.31	64.43 ± 0.45	85.90 ± 0.33
		<i>ft+cl</i>	78.42 ± 0.34	62.31 ± 0.44	81.98 ± 0.38	65.71 ± 0.47	81.08 ± 0.32	64.98 ± 0.43	86.06 ± 0.32
Meta-Learning	LEO [69]	<i>ft</i>	70.85 ± 0.38	56.59 ± 0.48	74.02 ± 0.41	59.17 ± 0.52	73.92 ± 0.36	58.02 ± 0.47	77.20 ± 0.38
		<i>cl</i>	73.84 ± 0.37	57.63 ± 0.47	80.19 ± 0.40	60.03 ± 0.51	78.82 ± 0.36	58.55 ± 0.48	84.74 ± 0.37
		<i>ft+cl</i>	73.01 ± 0.36	56.69 ± 0.48	78.41 ± 0.40	61.16 ± 0.53	76.22 ± 0.35	58.32 ± 0.46	81.74 ± 0.38
	MTL [73]	<i>ft</i>	76.37 ± 0.37	59.36 ± 0.48	81.04 ± 0.39	63.88 ± 0.50	79.25 ± 0.34	62.84 ± 0.46	85.10 ± 0.39
		<i>cl</i>	74.49 ± 0.36	58.48 ± 0.48	80.25 ± 0.38	65.25 ± 0.51	75.86 ± 0.35	59.77 ± 0.47	82.15 ± 0.37
		<i>ft+cl</i>	76.77 ± 0.35	60.52 ± 0.47	80.97 ± 0.36	65.44 ± 0.49	77.81 ± 0.34	61.81 ± 0.46	84.95 ± 0.36
	MN [71]	<i>ft</i>	74.66 ± 0.36	58.62 ± 0.46	80.74 ± 0.37	65.37 ± 0.50	76.13 ± 0.35	60.22 ± 0.47	82.31 ± 0.37
		<i>cl</i>	71.91 ± 0.35	61.09 ± 0.47	81.43 ± 0.36	66.03 ± 0.48	77.72 ± 0.34	62.19 ± 0.45	85.04 ± 0.36
		<i>ft+cl</i>	75.65 ± 0.35	58.49 ± 0.46	81.14 ± 0.36	64.29 ± 0.50	77.30 ± 0.34	62.99 ± 0.46	83.23 ± 0.37
	SIB [75] (transductive)	<i>ft</i>	77.17 ± 0.35	58.85 ± 0.44	82.01 ± 0.36	64.67 ± 0.47	79.40 ± 0.34	63.65 ± 0.45	84.76 ± 0.36
		<i>cl</i>	77.10 ± 0.34	58.86 ± 0.45	82.34 ± 0.36	66.70 ± 0.51	79.29 ± 0.35	63.14 ± 0.46	86.21 ± 0.37
		<i>ft+cl</i>	78.03 ± 0.33	61.17 ± 0.45	82.35 ± 0.35	65.72 ± 0.48	80.20 ± 0.33	64.40 ± 0.45	86.02 ± 0.35
	SIB [75] (inductive)	<i>ft</i>	75.21 ± 0.35	61.05 ± 0.46	79.92 ± 0.37	66.01 ± 0.50	77.15 ± 0.36	63.45 ± 0.45	82.43 ± 0.37
		<i>cl</i>	75.52 ± 0.35	61.23 ± 0.45	80.18 ± 0.36	66.33 ± 0.49	77.80 ± 0.35	64.42 ± 0.46	83.82 ± 0.36
		<i>ft+cl</i>	75.40 ± 0.34	61.14 ± 0.44	80.04 ± 0.35	66.26 ± 0.50	77.23 ± 0.35	64.21 ± 0.47	82.77 ± 0.35
	SIB [75] (inductive)	<i>ft</i>	76.73 ± 0.34	62.64 ± 0.46	80.79 ± 0.35	67.30 ± 0.48	78.55 ± 0.36	64.89 ± 0.44	84.03 ± 0.36
		<i>cl</i>	78.88 ± 0.35	67.10 ± 0.56	85.09 ± 0.35	77.64 ± 0.58	81.73 ± 0.34	71.31 ± 0.56	88.19 ± 0.34
		<i>ft+cl</i>	79.58 ± 0.35	67.94 ± 0.55	85.12 ± 0.35	77.68 ± 0.57	82.00 ± 0.34	71.95 ± 0.56	88.20 ± 0.34
	SIB [75] (inductive)	<i>ft</i>	79.04 ± 0.33	67.77 ± 0.55	85.22 ± 0.35	77.72 ± 0.56	81.93 ± 0.35	71.66 ± 0.56	88.21 ± 0.33
		<i>cl</i>	80.32 ± 0.35	68.85 ± 0.56	85.43 ± 0.35	78.03 ± 0.57	83.21 ± 0.33	73.51 ± 0.56	88.69 ± 0.33
		<i>ft+cl</i>	75.64 ± 0.36	57.20 ± 0.57	81.69 ± 0.34	65.51 ± 0.56	78.17 ± 0.35	60.12 ± 0.56	84.96 ± 0.36
	SIB [75] (inductive)	<i>ft</i>	76.23 ± 0.35	58.67 ± 0.56	82.04 ± 0.35	66.69 ± 0.57	79.34 ± 0.35	61.77 ± 0.56	85.24 ± 0.36
		<i>cl</i>	76.61 ± 0.35	58.12 ± 0.55	82.21 ± 0.35	66.28 ± 0.56	79.11 ± 0.35	61.25 ± 0.55	85.63 ± 0.34
		<i>ft+cl</i>	77.68 ± 0.34	60.33 ± 0.54	82.75 ± 0.35	67.34 ± 0.55	80.05 ± 0.34	63.14 ± 0.54	86.14 ± 0.34

Linear Classifier. $f(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$, where $\mathbf{x}$ is the input feature, $\mathbf{W} \in \mathbb{R}^{K \times N}$ is the learnable weight parameter, N is the feature dimension and $\mathbf{b} \in \mathbb{R}^K$ is the learnable bias parameter.

Cosine Classifier. $f(\mathbf{x}) = \mathbf{W}\mathbf{x} / \|\mathbf{W}\| \|\mathbf{x}\|$, where $\mathbf{W} \in \mathbb{R}^{K \times N}$ is the learnable weight parameter. We implemented cosine classifier without using the bias term.

***k*-NN Classifier.** Our implementation of *k*-NN is similar to [34, 221]. For each of the K classes, we first calculated the average support set feature (centroid) denoted as $\mathbf{x}_i, i \in \{1, \dots, K\}$. The classifier output for class i is then given by $f_i(\mathbf{x}) = -\|\mathbf{x} - \mathbf{x}_i\|^2$. Notice that the prediction given by this classifier will be the nearest centroid.

Fine-Tuning. We froze the backbone and used the average pooling layer output of Ω to learn the classifier. The output logits from classifier functions are normalized using softmax to generate probability output $P(y|\mathbf{x})$. For linear and cosine classifier, we followed [66] and trained the classifier for 100 iteration with a batch

size of 4. For fine-tuning baseline, we set the learning rate as 1×10^{-2} and weight decay as 1×10^{-3} . For IFSL, we set the learning rate as 5×10^{-3} and weight decay as 1×10^{-3} . k -NN classifier is non-parametric and can be initialized directly from support set.

MAML. MAML [67] aims to learn an initialization of network parameters such that it can be fine-tuned within a few steps to solve a variety of few-shot classification tasks. When using pre-trained network with MAML, it has been shown that learning initialization of the backbone can lead to unsatisfactory performance [66, 73]. Therefore in our experiment, we froze the backbone and appended a 2-layer MLP with ReLU activation in between the hidden layers and a linear classifier after the average pooling layer of Ω . The hidden dimension of the layers in MLP is the same as output dimension of Ω (512 for ResNet-10 and 640 for WRN-28-10). The initialization of MLP and the linear classifier is meta-learnt using MAML. For hyper-parameters, we set the inner loop learning rate $\alpha = 0.01$, the outer loop learning rate $\beta = 0.01$ and the number of adaptation steps as 20. For IFSL, we adopted the same hyper-parameter setting and set $n=8$ for feature-wise and combined adjustment. Implementation-wise, we adopted the released code² from [66] and performed experiments on MAML without using first-order approximation. Following the implementation in [66], the model was trained on 10,000 randomly sampled tasks with model selection using validation accuracy. We used 2,000 randomly sampled tasks for validation and testing.

MTL. MTL [73] learns scaling and shifting parameters at each convolutional layer of the backbone. We used the MTL implementation released by the author³ which adopts linear classifier. We integrated our ResNet-10 and WRN-28-10 backbones into the released code. The learning rate for scaling and shifting weights ϕ_{ss} and initial classifier parameters was set to 1×10^{-4} uniformly. We set the inner loop learning rate for classifier as 1×10^{-2} and the inner loop update step as 100. For IFSL, we adopted the same hyper-parameter setting and set $n=8$ for feature-wise and combined adjustment. We trained the MTL model on 10,000 randomly sampled tasks with model selection using validation accuracy and used 2,000 randomly sampled tasks for validation and testing. We used 3 RTX 2080 Ti for MTL experiments on WRN-28-10 backbone.

²<https://github.com/wyharveychen/CloserLookFewShot>

³<https://github.com/yaoyao-liu/meta-transfer-learning>

LEO. LEO [69] learns to generate classifier parameters conditioned on support set and the generated parameters are further fine-tuned within each FSL task. Our experiments were conducted on the released code of LEO⁴ using linear classifier. Following author’s implementation, we saved the center cropped features from our pre-trained backbones and used the saved features to train LEO. For baseline, we used the hyper-parameter settings released by the author. For IFSL, we set $n=8$ for feature-wise and combined adjustment and halved the outer loop learning rate compared to baseline. The model was trained up to 100,000 randomly sampled tasks from training split with early stopping using validation accuracy. We used 2,000 randomly sampled tasks for validation and testing.

Matching Net. Matching Net (MN) [71] is a metric-based method that learns a distance kernel function for k -NN. We used the Matching Net implementation in [66]. The implementation follows the setup in [71] and uses LSTM-based fully conditional embedding. We set the learning rate as 0.01 uniformly. For IFSL, we used $n=16$ for feature-wise and combined adjustment. The model was trained using 10,000 randomly sampled tasks with model selection using validation accuracy. We used 2,000 randomly sampled tasks for validation and testing.

SIB. SIB [75] initializes classifier from support set and generates gradients conditioned on unlabelled query set features to update classifier parameters. We followed the SIB implementation released by the author⁵ which uses cosine classifier. In the transductive setting, the query set size is set to 15. In the inductive setting, we used only 1 query sample randomly selected from the K classes in each episode. In terms of hyper-parameter settings, we took 3 synthetic gradient steps ($K = 3$) for all our experiments. For baseline, the learning rate for SIB network and classifier was set to 1×10^{-3} following author’s implementation. For IFSL, we set the learning rate to 5×10^{-4} and used $n=4$ for feature-wise and combined adjustment. In both transductive and inductive settings, we meta-trained SIB using 50,000 randomly sampled tasks with model selection using validation accuracy. We used 2,000 randomly sampled tasks for validation and testing.

⁴<https://github.com/deepmind/leo>

⁵https://github.com/hushell/sib_meta_learn

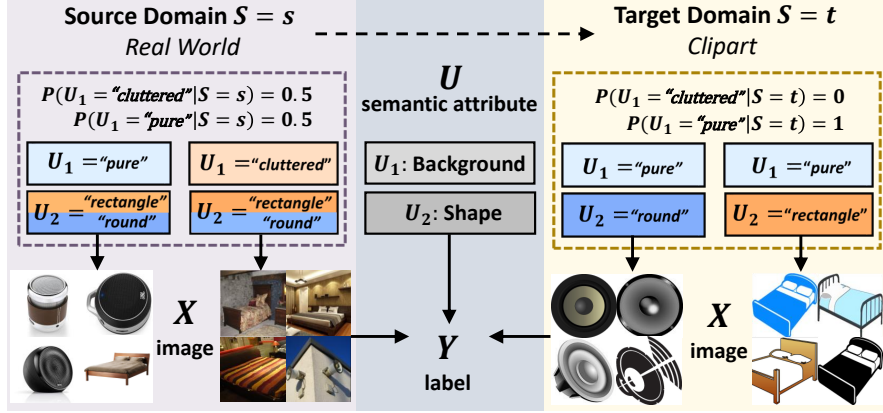


FIGURE 3.9: A DA example from source “Real World” to target “Clipart” domain in the Office-Home benchmark [7].

3.3 Case Study: Unsupervised Domain Adaptation

3.3.1 Introduction

Machine learning is always challenged when transporting training knowledge to testing deployment, which is generally known as Domain Adaptation (DA) [36]. As shown in Figure 3.9, when the *target* domain ($S = t$) is drastically different (*e.g.*, image style) from the *source* domain ($S = s$), deploying a classifier trained in $S = s$ results in poor performance due to the large *domain gap* [222]. To narrow the gap, conventional supervised DA requires a small set of labeled data in $S = t$ [223, 224], which is expensive and sometimes impractical. Therefore, we are interested in a more practical setting: Unsupervised DA (UDA), where we can leverage the abundant unlabelled data in $S = t$ [36, 96]. For example, when adapting an autopilot system trained in one country to another, where the street views and road signs are different, one can easily collect unlabelled street images from a camera-equipped vehicle cruise.

Existing UDA literature widely adopts the following two assumptions on the domain gap⁶: 1) *Covariate Shift* [89, 226]: $P(X|S = s) \neq P(X|S = t)$, where X denotes the samples, *e.g.*, real-world vs. clip-art images; and 2) *Conditional Shift* [100, 227]: $P(Y|X, S = s) \neq P(Y|X, S = t)$, where Y denotes the labels, *e.g.*, in clip-art domain, the “pure background” feature extracted from X is no

⁶A few early works [225] also consider target shift of $P(Y)$, which is now mainly discussed in other settings like long-tailed classification [41].

longer a strong visual cue for Y = “speaker” as in real-world domain. To turn “ $\neq$ ” into “ $=$ ”, almost all existing UDA solutions rely on learning invariant (or common) features in both source and target domains [89, 91, 96]. Unfortunately, due to the lack of supervision in target domain, it is challenging to capture such domain invariance.

For example, in Figure 3.9, when $S = s$, the distribution of “Background = pure” and “Background = cluttered” is already informative for “speaker” and “bed”, a classifier may recklessly tend to downplay the “Shape” features (or attributes) as they are not as “invariant” as “Background”; but, it does not generalize to $S = t$, where all clip-art images have “Background = pure” that is no longer discriminative. A popular approach to make up for the above loss of “Shape” is by imposing unsupervised reconstruction [99]—any feature loss hurts reconstruction. However, it is well-known that discrimination and reconstruction are adversarial objectives, leaving the finding of their ever-elusive trade-off parameter *per se* an open problem [33, 112, 228].

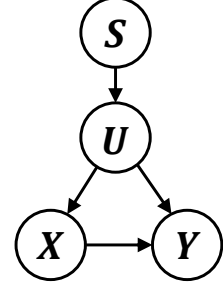


FIGURE 3.10: Causal graph of domain adaptation.

To systematically understand how domain gap causes the feature loss, we propose to use the *transportability theory* [49] to replace the above “shift” assumptions that overlook the explicit role of semantic attributes. Before we introduce the theory, we first abstract the DA problem in Figure 3.9 into the causal graph in Figure 3.10. We assume that the classification task $X \rightarrow Y$ is also affected by the unobserved semantic feature U (e.g., shape and background), where $U \rightarrow X$ denotes the generation of pixel-level image samples X and $U \rightarrow Y$ denotes the definition process of semantic class. Note that these causalities have been already shown valid in their respective areas, e.g., $U \rightarrow X$ and $U \rightarrow Y$ can be image [229] and language generation [230], respectively. In particular, the introduction of the domain selection variable S reveals that the fundamental reasons of the covariate shift and conditional shift are both due to $P(U|S = s) \neq P(U|S = t)$.

Note that the domain-aware U is the *confounder* that prevents the model from learning the domain-invariant causality $X \rightarrow Y$. It has been theoretically proven that the confounding effect cannot be eliminated by statistical learning without causal intervention [40]. Fortunately, the transportability theory offers a principled

DA solution based on the causal intervention in Figure 3.10:

$$P(Y|do(X), S) = \sum_u P(Y|X, U = u)P(U = u|S). \quad (3.11)$$

Note that the goal of DA is achieved by causal intervention $P(Y|do(X), S)$ using the *do*-operator [115]. To appreciate the merits of the calculus on the right-hand side of Eq. (3.11), we need to understand the following two points. First, $P(Y|X, U)$ is domain-agnostic as S is separated from X and Y given $U = u$ [40], thus $P(Y|X, U)$ generalizes to $S = t$ in testing even if it is trained on $S = s$. Second, every stratum of U is fairly adjusted subject to the domain prior $P(U|S)$, *i.e.*, forcing the model to respect “shape” as it is the only “invariance” that distinguishes between “speaker” and “bed” in controlled “cluttered” or “pure” backgrounds. Therefore, the semantic loss is eliminated.

However, Eq. (3.11) is purely declarative and far from practical, because it is still unknown how to implement the stratification and representation of the *unobserved* U , not mentioning to transport it across domains. To this end, we propose a novel approach to make Eq. (3.11) practical in UDA:

Identify the number of U by disentangled causal mechanisms. In practice, the number of U can be large due to the combinations of multiple attributes, making Eq. (3.11) computationally prohibitive (*e.g.*, many network passes). Fortunately, if the attributes are disentangled, we can stratify U according to the much smaller number of disentangled attributes, as the effect of each feature is independent with each other [38], *e.g.*, $P(Y|X, U = (u_1, u_2)) \propto P(Y|X, u_1) + P(Y|X, u_2)$ if u_1 and u_2 are disentangled. As detailed in Section 3.3.2, this motivates us to discover a small number of Disentangled Causal Mechanisms (DCMs) in unsupervised fashion [231], each of which corresponds to a feature-specific intervention between $X|S = s$ and $X|S = t$, *e.g.*, mapping a real-world “speaker” image to its clip-part counterpart by changing “Background” from “cluttered” to “pure”.

Represent U by proxy variables. Yet, DCMs do not provide vector representations of U , leaving difficulties in implementing Eq. (3.11) as neural networks. In Section 3.3.3, we show that the transformed output $\hat{X}$ of each DCM taking X as input can be viewed as a proxy of U [232], who provides a theoretical guarantee that we can replace the *unobserved* U with the *observed* $\hat{X}$ to make Eq. (3.11) computable.

Overall, instead of transporting the abstract and unobserved U in the original transportability theory of Eq. (3.11), we transport the concrete and learnable disentangled causal mechanisms, which generate the observed proxy of U . Therefore, we term the approach as **Transporting Casual Mechanisms (TCM)** for UDA. Through extensive experiments, our approach achieves state-of-the-art performance on UDA benchmarks: 70.7% on Office-Home [7], 90.5% on ImageCLEF-DA [9] and 75.8% on VisDA-2017 [3]. Specifically, we show that learning disentangled mechanisms and capturing the causal effect through proxy variables are the keys to improve the performance, which validates the effectiveness of our approach.

3.3.2 Disentangled Causal Mechanisms Discovery

Though Eq. (3.11) provides a principled solution for DA, it is impractical due to the unobserved U . To this end, the proposed TCM is a two-stage approach to make it practical. The first stage identifies the U stratification by discovering Disentangled Causal Mechanisms (DCMs) (Section 3.3.2) and the second stage represents each stratification of U with the proxy variable generated from the discovered DCMs (Section 3.3.3). The training and inference procedures are summarized in Algorithm 1 and 2, respectively.

Physicists believe that real-world observations are the outcome of the combination of independent physical laws. In causal inference [38, 167], we denote these laws as disentangled generative factors, such as shape, color, and position. For example, in our causal graph of Figure 3.10, if the semantic attribute U can be disentangled into generative causal factors $(U_1, \dots, U_k)$, each one will independently contribute to the observation: $P(Y|X) \propto \sum_{i=1}^k P(Y|X, U_i)$. Even though $|U|$ is large, *i.e.*, the number of sum in Eq. (3.11) is expensive, the disentanglement can still significantly reduce the number to a small k .

However, learning a disentangled representation of $U = (U_1, \dots, U_k)$ without supervision is challenging [47]. Fortunately, we can observe the outcomes of U : the source domain samples X_s and the target domain samples X_t generated from $P(X|U)$. This motivate us to discover k pairs of end-to-end functions $\{(M_i, M_i^{-1})\}_{i=1}^k$ in unsupervised fashion, where $M_i : X_s \rightarrow X_t$ and $M_i^{-1} : X_t \rightarrow X_s$, as shown in Figure 3.11 (a). Each (M_i, M_i^{-1}) corresponds to a counterfactual mapping that transforms a sample to the counterpart domain by intervening a disentangled factor

Algorithm 1 Two-stage Training of TCM

-
- 1: **Input:** Labelled domain $S = s$, unlabelled domain $S = t$, pre-trained backbone parameters β
 - 2: **Output:** DCMs $\{(M_i, M_i^{-1})\}_{i=1}^k$, fine-tuned backbone parameters β and linear functions parameters ω
 - 3: Randomly initialize $\{(M_i, M_i^{-1})\}_{i=1}^k, \omega$, VAE parameters θ , discriminator parameters γ (see Eq. (3.22))
 - 4: **repeat**
 - 5: // See Section 3.1 for details
 - 6: Sample $\mathbf{x}$ randomly from $S = s$ or $S = t$
 - 7: Calculate $\mathcal{L}_{CycleGAN}^i$ for $i \in \{1, \dots, k\}$
 - 8: Update $\{(M_i, M_i^{-1})\}_{i=1}^k$ with Eq. (3.12)
 - 9: **until** convergence
 - 10: **repeat**
 - 11: // See Section 3.2 for details
 - 12: Sample $\mathbf{x}_s, y_s$ from $S = s$, $\mathbf{x}_t$ from $S = t$
 - 13: Obtain $\hat{\mathcal{X}}_s = \{M_i(\mathbf{x}_s)\}_{i=1}^k, \hat{\mathcal{X}}_t = \{M_i^{-1}(\mathbf{x}_t)\}_{i=1}^k$
 - 14: Calculate $h_y(\mathbf{x}_s, \hat{\mathbf{x}}), \forall \hat{\mathbf{x}} \in \hat{\mathcal{X}}_s$ with Eq. (3.21)
 - 15: Obtain $P(Y|do(X = \mathbf{x}_s), S = s)$ with Eq. (3.19)
 - 16: Update $\omega, \beta, \theta, \gamma$ with Eq. (3.22)
 - 17: **until** convergence
-

Algorithm 2 Inference with TCM

-
- 1: **Input:** $\mathbf{x}_t, P(\hat{X}|S = t)$, DCMs $\{M_i^{-1}\}_{i=1}^k$, backbone parameters β and linear functions parameters ω
 - 2: **Output:** Predicted label y_t
 - 3: Obtain $\hat{\mathcal{X}}_t = \{M_i^{-1}(\mathbf{x}_t)\}_{i=1}^k$
 - 4: Calculate $h_y(\mathbf{x}_t, \hat{\mathbf{x}}), \forall \hat{\mathbf{x}} \in \hat{\mathcal{X}}_t$ with Eq. (3.21)
 - 5: $y_t \leftarrow \arg\max P(Y|do(X = \mathbf{x}_t), S = t)$ with Eq. (3.19)
-

U_i , i.e., modifying U_i w.r.t. the domain shift while fixing the values of other factors $U_{j \neq i}$ —this is essentially the definition of independent causal mechanisms [231], each of which corresponds to a disentangled causal factor. Hence we refer to these mapping functions as DCMs.

We first make a strong assumption that the one-to-one correspondence between (M_i, M_i^{-1}) and U_i has been already established, then we relax this assumption by removing the correspondence for the proposed practical solution. Without loss of generality, we consider a source domain sample $\mathbf{x}_s$. For each $M_i : X_s \rightarrow X_t$, we obtain a transformed sample $M_i(\mathbf{x}_s)$, corresponding to the interventional outcome of U_i drawn from $P(U_i|S = s)$ to $P(U_i|S = t)$. To ensure that the interventions are disentangled, we use the Counterfactual Faithfulness theorem [233, 234], which

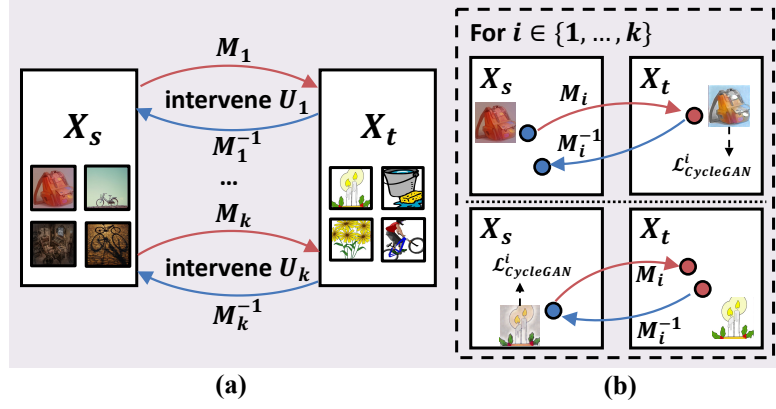


FIGURE 3.11: (a) Our DCMs $\{(M_i, M_i^{-1})\}_{i=1}^k$, where each $M_i : X_s \rightarrow X_t$ and $M_i^{-1} : X_t \rightarrow X_s$ correspond to intervention on the disentangled attribute U_i . (b) CycleGAN [8] loss $\mathcal{L}_{CycleGAN}^i$ on each (M_i, M_i^{-1}) .

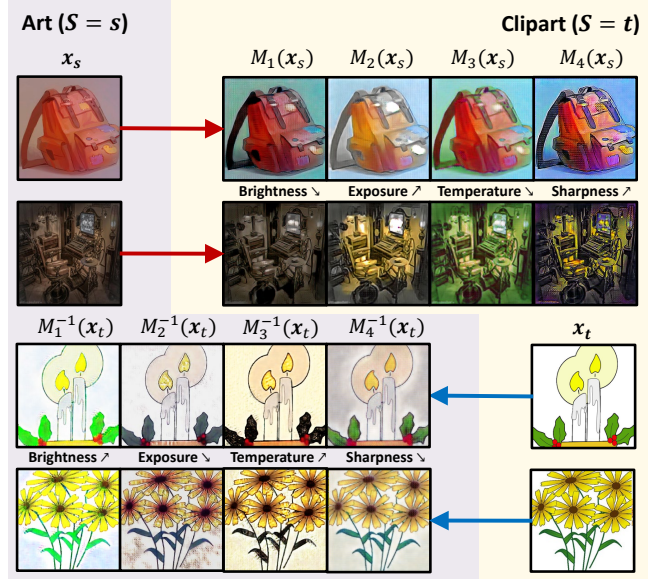


FIGURE 3.12: Transformation between “Art” and “Clipart” domain with 4 trained DCMs $\{(M_i, M_i^{-1})\}_{i=1}^4$, where each pair of mapping functions specializes in brightness, exposure, color temperature and sharpness, respectively.

guarantees that M_i is a disentangled intervention if and only if $M_i(\mathbf{x}_s) \sim P(X_t)$ (proof in Section 3.3.5). Hence, as shown in Figure 3.11 (b), we use the CycleGAN loss [8] denoted as $\mathcal{L}_{CycleGAN}^i$ to make the transformed samples from *each* $M_i : X_s \rightarrow X_t$ indistinguishable from the real samples in $S = t$. Likewise, we apply the CycleGAN loss to each $M_i^{-1} : X_t \rightarrow X_s$ such that the transformed samples are close to real samples in $S = s$.

Now we relax the above one-to-one correspondence assumption. We begin with a sufficient condition (proof in Section 3.3.5): *if (M_i, M_i^{-1}) intervenes U_i , the i -th mapping function outputs the counterfactual faithful generation,*

i.e., $i = \operatorname{argmin}_{j \in \{1, \dots, k\}} \mathcal{L}_{CycleGAN}^j$. To “guess” the one-to-one correspondence, we adopt a practical method by using the necessary conclusion: if $i = \operatorname{argmin}_{j \in \{1, \dots, k\}} \mathcal{L}_{CycleGAN}^j$, then (M_i, M_i^{-1}) corresponds to U_i intervention. Specifically, training samples are fed into all (M_i, M_i^{-1}) in parallel to compute $\mathcal{L}_{CycleGAN}^i$ for each pair. Only the winning pair with the smallest loss is updated. The objective is given by:

$$\begin{aligned} & \min_{(M_i, M_i^{-1})} \mathcal{L}_{CycleGAN}^i, \\ & \text{where } i = \operatorname{argmin}_{j \in \{1, \dots, k\}} \mathcal{L}_{CycleGAN}^j, \end{aligned} \quad (3.12)$$

The functions (*e.g.*, M_i) in the optimization objective denote their parameters for simplicity. Note that this is not sufficient, hence our approach has limitations. Still, this practice has been empirically justified in [231], and we leave a more theoretically complete approach as future work.

CycleGAN Loss. Specifically, $\mathcal{L}_{CycleGAN}^i$ is implemented by:

$$\mathcal{L}_{CycleGAN}^i = \mathcal{L}_{adv}^i + \alpha_1 \mathcal{L}_{cyc}^i + \alpha_2 \mathcal{L}_{idt}^i \quad (3.13)$$

where α_1, α_2 are trade-off parameters. The adversarial loss $\mathcal{L}_{adv}^i$ requires the transformed images to look like real images in the opposing domain. For a sample $\mathbf{x}$, $\mathcal{L}_{real}^i$ is given by:

$$\mathcal{L}_{adv}^i = \begin{cases} \log(1 - D'_t(M_i(\mathbf{x}))), & \text{if } \mathbf{x} \text{ from } S = s, \\ \log(1 - D'_s(M_i^{-1}(\mathbf{x}))), & \text{otherwise,} \end{cases} \quad (3.14)$$

where D'_s, D'_t are discriminators that return a large value when its input looks like images in $S = s, S = t$, respectively. $\mathcal{L}_{cyc}^i$ is the cycle consistency loss [8] given by

$$\mathcal{L}_{cyc}^i = \begin{cases} \|M_i^{-1}(M_i(\mathbf{x})) - \mathbf{x}\|, & \text{if } \mathbf{x} \text{ from } S = s, \\ \|M_i(M_i^{-1}(\mathbf{x})) - \mathbf{x}\|, & \text{otherwise,} \end{cases} \quad (3.15)$$

where $\|\cdot\|$ is implemented using L1 norm. We follow CycleGAN [8] to use an identity loss $\mathcal{L}_{idt}^i$ to improve generation quality:

$$\mathcal{L}_{idt}^i = \begin{cases} \|M_i^{-1}(\mathbf{x}) - \mathbf{x}\|, & \text{if } \mathbf{x} \text{ from } S = s, \\ \|M_i(\mathbf{x}) - \mathbf{x}\|, & \text{otherwise.} \end{cases} \quad (3.16)$$

After training, we obtain k DCMs $\{(M_i, M_i^{-1})\}_{i=1}^k$, where the i -th pair corresponds to U_i . Hence we identify the number of U as k . Figure 3.12 shows an example with $k = 4$, where the 4 DCMs correspond to “Brightness”, “Exposure”, “Temperature” and “Sharpness”, respectively⁷.

3.3.3 Representing U by Proxy Variables

While the learned DCMs identify the number of U , they do not provide a vector representation of U . Hence $P(U|S)$ and $P(Y|X, U)$ in Eq. (3.11) are still unobserved. Fortunately, U generates two *observed* variables $\hat{X}$ and Z as shown in Figure 3.13, which are called *proxy variables* [232] of U and make $P(U|S)$ and $P(Y|X, U)$ estimable. We will first explain $\hat{X}$ and Z as well as their related causal links.

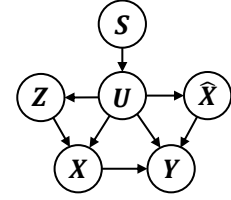


FIGURE 3.13: Causal graph with the proxy $\hat{X}$ and Z .

$\hat{X}, U \rightarrow \hat{X} \rightarrow Y$. $\hat{X}$ represents the DCMs outputs, *i.e.*, for a sample $\mathbf{x}_s$ in $S = s$, $\hat{X}|S = s$ takes value from $\hat{\mathcal{X}}_s = \{M_i(\mathbf{x}_s)\}_{i=1}^k$; and for a sample $\mathbf{x}_t$ in $S = t$, $\hat{X}|S = t$ takes value from $\hat{\mathcal{X}}_t = \{M_i^{-1}(\mathbf{x}_t)\}_{i=1}^k$. As detailed in Section 3.3.5, M_i, M_i^{-1} generates counterfactuals in the other domain by fixing $U_{j \neq i}$ and intervening U_i , *i.e.*, $\hat{X}$ is generated from U . Hence $U \rightarrow \hat{X}$ is justified. Moreover, as the other domain is also labelled, $\hat{X} \rightarrow Y$ denotes the predictive effects from the generated counterfactuals.

$Z, U \rightarrow Z \rightarrow X$. Z is a latent variable encoded from X by a VAE [235]. The link $Z \rightarrow X$ is because the latent variable is trained to reconstruct X . Furthermore, the latent variable of VAE is shown to capture some information of the underlying U [167], justifying $U \rightarrow Z$.

Note that the networks to obtain $\hat{X}$ and Z are trained in an unsupervised fashion. Hence $\hat{X}, Z$ are observed in both $S = s$ and $S = t$. Under this causal graph, we have a theoretically grounded solution to Eq. (3.11) using the theorem below (proof in Section 3.3.5 as corollary to [232]).

⁷Note that our DCM learning is orthogonal to the GAN models used. One can use more advanced GAN models to discover complex mechanisms such as shape or viewpoint changes.

Theorem (Proxy Function). *Under the causal graph in Figure 3.13, any solution $h_y(X, \hat{X})$ to Eq. (3.17) satisfies Eq. (3.18).*

$$P(Y|Z, X, S) = \sum_{\hat{\mathbf{x}}} h_y(X, \hat{\mathbf{x}}) P(\hat{X} = \hat{\mathbf{x}}|Z, X, S) \quad (3.17)$$

$$P(Y|X, U) = \sum_{\hat{\mathbf{x}}} h_y(X, \hat{\mathbf{x}}) P(\hat{X} = \hat{\mathbf{x}}|U). \quad (3.18)$$

Notice that we can set $S = s$ in Eq. (3.17) and use the labelled data in $S = s$ to learn h_y (details given below). We prove in the Section 3.3.5 that h_y is invariant across domains. This leads to the following inference strategy.

Inference

Taking expectation over $U|S = t$ on both sides of Eq. (3.18) leads to a practical solution to Eq. (3.11):

$$P(Y|do(X), S = t) = \sum_{\hat{\mathbf{x}} \in \hat{\mathcal{X}}_t} h_y(X, \hat{\mathbf{x}}) P(\hat{X} = \hat{\mathbf{x}}|S = t), \quad (3.19)$$

where $P(\hat{X}|S = t)$ is estimable as $\hat{X}$ is observed in $S = t$, and h_y is trained in $S = s$ as given below.

Learning h_y We detail the training of h_y , including the representation of its inputs X and $\hat{X}$, the function forms of the terms in Eq. (3.17) and the training objective.

Representation of $X, \hat{X}$. When X and $\hat{X}$ are images, directly evaluating Eq. (3.19) in image space can be computationally expensive, not to mention the tendency for severe over-fitting. Therefore, we follow the common practice in UDA [94, 101] to introduce a pre-trained feature extractor backbone (*e.g.*, ResNet [22]). Hereinafter, $X, \hat{X}$ denote their respective n -d features. Note that Z of l dimensions is encoded from X 's feature form ($l < n$).

Function Forms. We adopt an isotropic Gaussian distribution for $P(\hat{X}|S)$. We implement $P(Y|Z, X, S = s), P(\hat{X}|Z, X, S = s)$ in Eq. (3.17) with the following linear model $f_y(Z, X)$ and $f_{\hat{x}}(Z, X)$, respectively:

$$\begin{aligned} f_y(Z, X) &= \mathbf{W}_1 Z + \mathbf{W}_2 X + \mathbf{b}_1 \\ f_{\hat{x}}(Z, X) &= \mathbf{W}_3 Z + \mathbf{W}_4 X + \mathbf{b}_2, \end{aligned} \quad (3.20)$$

where $f_y : (Z, X) \rightarrow \mathbb{R}^c$ produces logits for c classes, $f_{\hat{x}} : (Z, X) \rightarrow \mathbb{R}^n$ predicts $\hat{X}$, $\mathbf{W}_1 \in \mathbb{R}^{c \times l}$, $\mathbf{W}_2 \in \mathbb{R}^{c \times n}$, $\mathbf{W}_3 \in \mathbb{R}^{n \times l}$, $\mathbf{W}_4 \in \mathbb{R}^{n \times n}$, $\mathbf{b}_1 \in \mathbb{R}^c$ and $\mathbf{b}_2 \in \mathbb{R}^n$. With this function form, we can solve $h_y : (X, \hat{X}) \rightarrow \mathbb{R}^c$ from Eq. (3.17) as (derivation in Section 3.3.5):

$$\begin{aligned} h_y(X, \hat{X}) &= \mathbf{b}_1 - \mathbf{W}_1 \mathbf{W}_3^+ \mathbf{b}_2 + \mathbf{W}_1 \mathbf{W}_3^+ \hat{X} \\ &\quad + (\mathbf{W}_2 - \mathbf{W}_1 \mathbf{W}_3^+ \mathbf{W}_4) X, \end{aligned} \quad (3.21)$$

where $(\cdot)^+$ denotes pseudo-inverse.

Overall Objective. The training objective is given by:

$$\min_{\omega, \beta, \theta} (\mathcal{L}_c + \mathcal{L}_v) + \min_{\beta} \max_{\gamma} \alpha \mathcal{L}_p, \quad (3.22)$$

where ω denotes the parameters of the linear functions f_y and $f_{\hat{x}}$, β denotes the parameters of the backbone, θ denotes the parameters of the VAE, $\mathcal{L}_c$ denotes the sum of Cross-Entropy loss to train f_y and the mean squared error loss to train $f_{\hat{x}}$ (see Eq. (3.20)). $\mathcal{L}_v$ is the VAE loss, $\mathcal{L}_p$ is the proxy loss, γ denotes the parameters of the discriminators used in $\mathcal{L}_p$, and α is a trade-off parameter.

VAE Loss. $\mathcal{L}_v$ is used to train the VAE, which contains an encoder $Q_{\theta}(Z|X)$ and a decoder $P_{\theta}(X|Z)$. Given a feature $\mathbf{x}_s$ in $S = s$, $\mathcal{L}_v$ is given by:

$$\begin{aligned} \mathcal{L}_v &= -\mathbb{E}_{Q_{\theta}(Z|X=\mathbf{x}_s)} [P_{\theta}(X = \mathbf{x}_s|Z)] \\ &\quad + D_{KL}(Q_{\theta}(Z|X = \mathbf{x}_s) \parallel P(Z)), \end{aligned} \quad (3.23)$$

where D_{KL} denotes KL-divergence and $P(Z)$ is set to the isotropic Gaussian distribution. Note that we only need to learn a VAE in $S = s$, as $h_y(X, \hat{X})$ is domain-agnostic.

Proxy Loss. In practice, the generated images from the DCMs $\{(M_i, M_i^{-1})\}_{i=1}^k$ may contain artifacts that contaminate the feature outputs from the backbone [236], making $\hat{X}$ as feature no longer resembles the features in the counterpart domain. Hence we regularize the backbone with a proxy loss to enforce feature-level resemblance between the samples transformed by the DCMs and the samples in the counterpart domain. Given feature $\mathbf{x}_s$ in $S = s$, $\mathbf{x}_t$ in $S = t$ and their corresponding

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet-50 [22]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [96]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [93]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
MCD [237]	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
MDD [238]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
CDAN [94]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
SymNets [104]	47.7	72.9	78.5	64.2	71.3	74.2	63.6	47.6	79.4	73.8	50.8	82.6	67.2
CADA [239]	56.9	76.4	80.7	61.3	75.2	75.2	63.2	54.5	80.7	73.9	61.5	84.1	70.2
ETD [240]	51.3	71.9	85.7	57.6	69.2	73.7	57.8	51.2	79.3	70.2	57.5	82.1	67.3
GVB-GD [10]	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
Baseline	54.3	70.1	75.2	60.4	71.5	72.3	60.1	52.1	74.2	71.5	56.2	81.3	66.6
TCM (Ours)	58.6	74.4	79.6	64.5	74.0	75.1	64.6	56.2	80.9	74.6	60.7	84.7	70.7

TABLE 3.5: Accuracy (%) on the Office-Home dataset [7] with 12 UDA tasks, where all methods are fine-tuned from ResNet-50 [22] pre-trained on ImageNet [23].

DCMs outputs sets $\hat{\mathcal{X}}_s, \hat{\mathcal{X}}_t$ (as n -d features), the loss is given by:

$$\begin{aligned}
\mathcal{L}_p = & \log D_s(\mathbf{x}_s) + \frac{1}{k} \sum_{\hat{\mathbf{x}}_s \in \hat{\mathcal{X}}_s} \log(1 - D_t(\hat{\mathbf{x}}_s)) \\
& + \log D_t(\mathbf{x}_t) + \frac{1}{k} \sum_{\hat{\mathbf{x}}_t \in \hat{\mathcal{X}}_t} \log(1 - D_s(\hat{\mathbf{x}}_t)),
\end{aligned} \tag{3.24}$$

where D_s, D_t are discriminators parameterized by γ that return a large value for features of real samples. Through min-max adversarial training, generated sample features $\hat{X}$ become similar to the sample features in the counterpart domain and fulfill the role as a proxy.

3.3.4 Experiment

Datasets

We validated TCM on three standard benchmarks for visual domain adaptation:

ImageCLEF-DA [9] is a benchmark dataset for ImageCLEF 2014 domain adaptation challenge, which contains three domains: 1) Caltech-256 (C), 2) ImageNet ILSVRC 2012 (I) and 3) Pascal VOC 2012 (P). For each domain, there are 12 categories and 50 images in each category. We permuted all the three domains and built six transfer tasks, *i.e.*, $I \rightarrow P$, $P \rightarrow I$, $I \rightarrow C$, $C \rightarrow I$, $C \rightarrow P$, $P \rightarrow C$.

Method	$I \rightarrow P$	$P \rightarrow I$	$I \rightarrow C$	$C \rightarrow I$	$C \rightarrow P$	$P \rightarrow C$	Avg
ResNet-50 [22]	74.8 ± 0.3	83.9 ± 0.1	91.5 ± 0.3	78.0 ± 0.2	65.5 ± 0.3	91.2 ± 0.3	80.7
DAN [96]	74.5 ± 0.4	82.2 ± 0.2	92.8 ± 0.2	86.3 ± 0.4	69.2 ± 0.4	89.8 ± 0.4	82.5
DANN [93]	75.0 ± 0.6	86.0 ± 0.3	96.2 ± 0.4	87.0 ± 0.5	74.3 ± 0.5	91.5 ± 0.6	85.0
RevGrad [241]	75.0 ± 0.6	86.0 ± 0.3	96.2 ± 0.4	87.0 ± 0.5	74.3 ± 0.5	91.5 ± 0.6	85.0
MADA [242]	75.0 ± 0.3	87.9 ± 0.2	96.0 ± 0.3	88.8 ± 0.3	75.2 ± 0.2	92.2 ± 0.3	85.8
CDAN [94]	77.7 ± 0.3	90.7 ± 0.2	97.7 ± 0.3	91.3 ± 0.3	74.2 ± 0.2	94.3 ± 0.3	87.7
A ² LP [102]	79.6 ± 0.3	92.7 ± 0.3	96.7 ± 0.1	92.5 ± 0.2	78.9 ± 0.2	96.0 ± 0.1	89.4
SymNets [104]	80.2 ± 0.3	93.6 ± 0.2	97.0 ± 0.3	93.4 ± 0.3	78.7 ± 0.3	96.4 ± 0.1	89.9
ETD [240]	81.0	91.7	97.9	93.3	79.5	95.0	89.7
Baseline	77.2 ± 0.5	89.7 ± 0.2	96.1 ± 0.3	92.1 ± 0.4	75.2 ± 0.3	93.5 ± 0.4	87.3
TCM (Ours)	79.9 ± 0.4	94.2 ± 0.2	97.8 ± 0.3	93.8 ± 0.4	79.9 ± 0.4	96.9 ± 0.4	90.5

TABLE 3.6: Accuracy (%) and the standard deviation on the ImageCLEF-DA dataset [9] with 6 UDA tasks, where all methods are fine-tuned from ResNet-50 [22] pre-trained on ImageNet [23].

Office-Home [7] is a very challenging dataset for UDA with 4 significantly different domains: Artistic images (A), Clipart (C), Product images (P) and Real-World images (R). It contains 15,500 images from 65 categories of everyday objects in the office and home scenes. We evaluated TCM in all 12 permutations of domain adaptation tasks.

VisDA-2017 [3] is a challenging simulation-to-real dataset that is significantly larger than the other two datasets with 280k images in 12 categories. It has two domains: Synthetic, with renderings of 3D models from different angles and with different lighting conditions; and Real with natural real-world images. We followed the common protocol [10, 94] to evaluate on Synthetic→Real task.

Setup

Evaluation Protocol. We followed the common evaluation protocol for UDA [10, 94, 104], where all labeled source samples and unlabeled target samples are used to train the model, and the average classification accuracy is compared in each dataset based on three random experiments. Following [102, 104], we reported the standard deviation on ImageCLEF-DA. For fair comparison, our TCM and all comparative methods used the backbone ResNet-50 [22] pre-trained on ImageNet [23].

Baseline. As existing domain-mapping methods either focus on the image segmentation task [243], or only evaluate on toy settings (*e.g.*, digit) [91], we implemented a domain-mapping baseline. Specifically, we trained a CycleGAN that transforms $X_s \rightarrow X_t$ and $X_t \rightarrow X_s$, whose network architecture is the same as each DCM in our proposed TCM. Then we learned a classifier on the transformed samples $X_s \rightarrow X_t$ with the standard cross-entropy loss, while using the loss in Eq. (3.24)

to align the features of the transformed samples with the target domain features. In inference, we directly used the learned classifier.

Results

Overall Results. As shown in Table 3.5, 3.6, 3.7, our method achieves the state-of-the-art average classification accuracy on Office-Home [7], ImageCLEF-DA [9], and VisDA-2017 [3]. Specifically, 1) ImageCLEF-DA has the smallest domain gap, where the 3 domains correspond to real-world images from 3 datasets. Our TCM outperforms existing methods on 4 out of 6 UDA tasks. 2) Office-Home has a much larger domain gap, such as the Artistic domain (A) and Clipart domain (C). Besides improvements in average accuracy, our method significantly outperforms existing methods on the most difficult tasks, *e.g.*, +1.6% on $A \rightarrow C$, +1.1% on $P \rightarrow C$. 3) VisDA-2017 is a large-scale dataset with 280k images, where TCM performs competitively. Note that the training complexity and convergence speed of TCM are comparable with existing methods (details in Section 3.3.5). Hence overall, our TCM is generally applicable to small to large-scale UDA tasks with various sizes of domain gaps.

Comparison with Baseline. We have three observations: 1) From Table 3.5, 3.6, 3.7, we notice that the Baseline does not perform competitively on 3 benchmarks. This shows that the improvements from TCM are not the results of superior image generation quality from CycleGAN. 2) Baseline accuracy in Table 3.6 is much lower compared with our TCM with a single DCM ($k = 1$) in Table 3.8 on ImageCLEF-DA. The only difference between them is that Baseline train a classifier directly with the transformed sample $X_s \rightarrow X_t$, while TCM($k = 1$) uses the proxy function $h_y(X, \hat{X})$ in Eq. (3.19). Hence this validates that the proxy theory in Section 3.3.2 has some effects in removing the confounding effect, even without stratifying U . As shown later, this is because $\hat{X}$ can make up for the lost semantic in X . 3) On all 3 benchmarks, when using multiple DCMs, TCM significantly outperforms Baseline, which validates our practical implementation of Eq. (3.19) by combining DCMs and proxy theory to identify the stratification and representation of U .

Number of DCMs. We performed an ablation on the number of DCMs k with

Method	Acc.
DAN [96]	61.6
DANN [93]	57.4
GTA [244]	69.5
MDD [238]	74.6
CDAN [94]	70.0
GVB-GD [10]	75.3
DMRL [245]	75.5
Baseline	72.8
TCM (Ours)	75.8

Table 3.7: Accuracy (%) of Synthetic→Real task on the VisDA-2017 dataset [3].

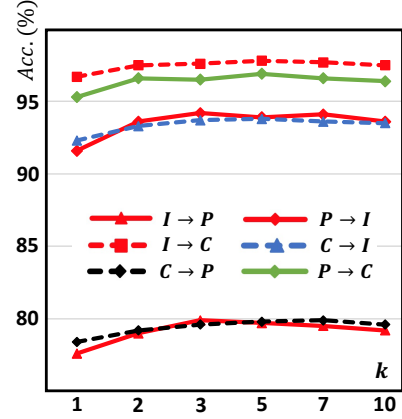


Figure 3.14: Plot of k against Accuracy (%) on the 6 UDA tasks from ImageCLEF-DA [9].

k	$I \rightarrow P$	$P \rightarrow I$	$I \rightarrow C$	$C \rightarrow I$	$C \rightarrow P$	$P \rightarrow C$	Avg
1	77.6	91.6	96.7	92.3	78.4	95.3	88.7
2	79	93.6	97.5	93.3	79.2	96.6	89.9
3	79.9	94.2	97.6	93.7	79.6	96.5	90.3
5	79.7	93.9	97.8	93.8	79.8	96.9	90.3
7	79.5	94.1	97.7	93.6	79.9	96.6	90.2
10	79.2	93.6	97.5	93.5	79.6	96.4	90.0

TABLE 3.8: Ablation on the number of DCMs k with accuracy (%) on the 6 UDA tasks from ImageCLEF-DA [9].

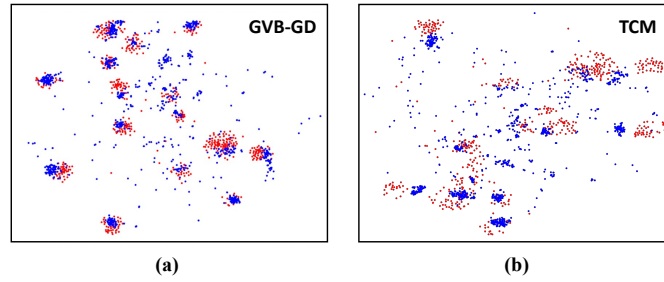


FIGURE 3.15: t-SNE [6] plot of features in $S = s$ (red) and in $S = t$ (blue) after training using method: (a) GVB-GD [10], (b) Our TCM. Only samples from the first 15 classes are plotted to avoid clutter.

ImageCLEF-DA dataset. The results are shown in Table 3.8 and plotted in Figure 3.14. We have three observations: 1) Among each step of k , the largest difference occurs between $k = 1$ and $k = 2$. This means that accounting for 2 disentangled factors can explain much of the domain shift due to $P(U|S = s) \rightarrow P(U|S = t)$ in ImageCLEF-DA. 2) Across all tasks, the best performance is usually achieved with $k = 5$. This supports the effectiveness of stratifying U according to a small

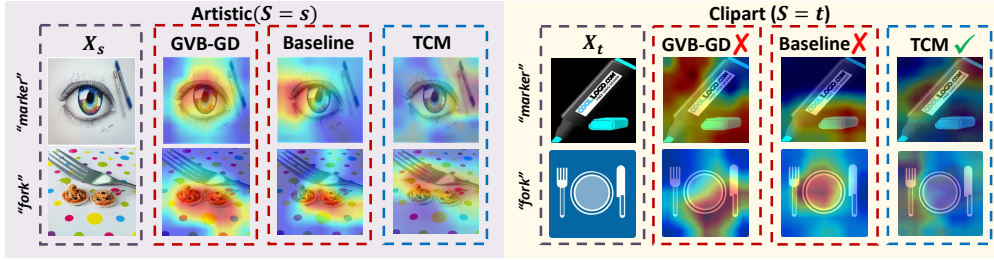


FIGURE 3.16: Class Activation Maps (CAMs) [5] of GVB-GD [10], Baseline and TCM on the Artistic($S = s$) $\rightarrow$ Clipart($S = t$) task in Office-Home dataset [7]. The left column shows two samples in each domain, whose class name is indicated on the left. In $S = t$, we show two samples predicted wrongly by GVB-GD and Baseline, but correctly with TCM.

number of disentangled attributes. 3) A large U (e.g., 10) leads to a slight performance drop. In practice, we observed that when k is large, a few DCMs seldom win any sample from Eq. (3.12), and hence are poorly trained. This can impact their image generation quality, leading to reduced performance. We argue that this is because the chosen k is larger than the number of factors that can be disentangled from the dataset. For example, if there exist $k' < k$ factors ($U_1, \dots, U_{k'}$), once k' DCMs establish correspondence with them, the rest $(k - k')$ DCMs will never win any sample, as they can never generate images that are more counterfactual faithful compared with intervention on each U_i (see Counterfactual Faithfulness theorem in Section 3.3.5).

t-SNE Plot. In Figure 3.15, we show the t-SNE [6] plots of the features in $S = s$ and $S = t$ in Office-Home A $\rightarrow$ C task after training, for both the current SOTA GVB-GD [10] and our TCM. GVB-GD is based on invariant-feature-learning and focuses on making X_s and X_t alike, while our TCM does not explicitly align them. We indeed observe a better alignment between the features in $S = s$ and $S = t$ in GVB-GD. However, as shown in Table 3.5, our method outperforms GVB-GD on A $\rightarrow$ C. This shows that the current common view on the solution to UDA based on [246], i.e., making domains alike while minimizing source risk, does not necessarily lead to the best performance. In fact, this is in line with the conclusion from recent theoretical works [247]. Our approach is orthogonal to existing approaches and we demonstrate the practicality of a more principled solution to UDA, i.e., establishing appropriate causal assumption on the domain shift and solving the corresponding transport formula (e.g., Eq. (3.11)).

Alleviating Semantic Loss. We use Figure 3.16 to reveal the semantic loss problem on existing methods, and how our method can alleviate it. The figure

shows the CAM [5] responses on images in $S = s$ and $S = t$ from the $A \rightarrow C$ task in Office-Home dataset. We compared with two lines of existing approaches: invariant-feature-learning method (GVB-GD [10]) and domain-mapping method (our Baseline), and we generated CAM for each of them as shown in the red dotted box. For our TCM, the prediction is a linear combination of the effects from the two images: X and the DCMs outputs $\hat{X}$ (see Eq. (3.21)). As the locations of objects tend to remain the same in $\hat{X}$ as in X (see Figure 3.12), we visualize the overall CAM of our TCM in the blue box by combining the CAM from X and $\hat{X}$ weighted by their contributions towards softmax prediction probability. On the left, we show the CAMs in $S = s$. By looking at the CAM of GVB-GD and the Baseline, we observe that they focus on the contextual object semantics (*e.g.*, food that commonly appears together with “fork”) to distinguish “marker” and “fork”, where the effect from the object shape is mostly lost. In contrast, our TCM focuses on the marker and fork, *i.e.*, the shape semantic is preserved in training. On the right in $S = t$, the contextual object semantic is no longer discriminative for the two classes. Hence GVB-GD and Baseline either focus on the wrong semantic (*e.g.*, plate) or become confused (focusing on a large area), leading to the wrong prediction. Thanks to the preserved shape semantic, our TCM focuses on the object and makes the correct prediction. This provides an intuitive explanation of how our TCM alleviates the semantic loss (more examples in Section 3.3.5).

3.3.5 Supplementary Information

Counterfactual Faithfulness Theorem

We will first provide a brief introduction to the concept of counterfactual and disentanglement. Causality allows to compute how an outcome would have changed, had some variables taken different values, referred to as a counterfactual. In Section 3.3.2, we refer to each DCM (M_i, M_i^{-1}) in $\{(M_i, M_i^{-1})\}_{i=1}^k$ as a counterfactual mapping, where each M_i (or M_i^{-1}) essentially follows the three steps of computing counterfactuals [248] (conceptually): given a sample $X = \mathbf{x}$, 1) In abduction, $(U_1 = u_1, \dots, U_i = u_i, \dots, U_k = u_k)$ is inferred from $\mathbf{x}$ through $P(U|X)$; 2) In action, the attribute U_i is intervened by setting it to u'_i drawn from $P(U_i|S = t)$ (or $P(U_i|S = s)$), while the values of other attributes are fixed; 3) In prediction,

the modified $(U_1 = u_1, \dots, U_i = u'_i, \dots, U_k = u_k)$ is fed to the generative process $P(X|U)$ to obtain the output of the DCM $M_i(\mathbf{x})$ (or $M_i^{-1}(\mathbf{x})$). More details regarding counterfactual can be found in [115].

Our definition of disentanglement is based on [38] of group theory. Let $\mathcal{U}$ be a set of (unknown) generative factors, *e.g.*, such as shape and background. There is a set of *independent causal mechanisms* $\varphi : \mathcal{U} \rightarrow \mathcal{X}$, generating images from $\mathcal{U}$. Let $\mathcal{G}$ be the group acting on $\mathcal{U}$, *i.e.*, $g \circ u$ transforms $u \in \mathcal{U}$ using $g \in \mathcal{G}$ (*e.g.*, changing background “cluttered” to “pure”). When there exists a direct product decomposition $\mathcal{G} = \prod_{i=1}^k \mathcal{G}_i$ and $\mathcal{U} = \prod_{i=1}^k \mathcal{U}_i$ such that $\mathcal{G}_i$ acts on $\mathcal{U}_i$, we say that each $\mathcal{U}_i$ is the space of a disentangled factor. The causal mechanism (M_i, M_i^{-1}) is disentangled when its transformation in $\mathcal{X}$ corresponds to the action of $\mathcal{G}_i$ on $\mathcal{U}_i$.

We use $\mathcal{U}$ and $\mathcal{X}$ to denote the vector space of U and X respectively. We denote the generative process $P(X|U)$ ($U \rightarrow X$) as a function $g : \mathcal{U} \rightarrow \mathcal{X}$. Note that we consider the function g as an embedded function [233], *i.e.*, a continuous injective function with continuous inversion, which generally holds for convolution-based networks as shown in [249]. Without loss of generality, we will consider the $S = s \rightarrow S = t$ mapping M_i for the analysis below, which can be easily extended to M_i^{-1} . Our definition of disentangled intervention follows the intrinsic disentanglement definition in [233], given by:

Definition (Disentangled Intervention). *A counterfactual mapping $M : \mathcal{X} \rightarrow \mathcal{X}$ is a disentangled intervention with respect to U_i , if there exists a transformation $M' : \mathcal{U} \rightarrow \mathcal{U}$ affecting only U_i , such that for any $u \in \mathcal{U}$,*

$$M(g(u)) = g(M'(u)). \quad (3.25)$$

Then we have the following theorem:

Theorem (Counterfactual Faithfulness Theorem). *The counterfactual mapping $M_i(X)$ is faithful if and only if M_i is a disentangled intervention with respect to U_i .*

Note that by definition, if $M_i(X)$ is faithful, $M_i(X) \in \mathcal{X}$. To prove the above theorem, one conditional is trivial: if M_i is a disentangled intervention, it is by definition an endomorphism of $\mathcal{X}$ so the counterfactual mapping must be faithful. For the second conditional, let us assume a faithful counterfactual mapping $M_i(X)$.

Given g is embedded, the counterfactual mapping can be decomposed as:

$$M_i(X) = g \circ M'_i \circ g^{-1}(X), \quad (3.26)$$

where $\circ$ denotes function composition and $M'_i : \mathcal{U} \rightarrow \mathcal{U}$ affecting only U_i . Now for any $u \in \mathcal{U}$, the quantity $M_i(g(u))$ can be similarly decomposed as:

$$M_i(g(u)) = g \circ M'_i \circ g^{-1} \circ g(u) = g \circ M'_i(u). \quad (3.27)$$

Since M'_i is a transformation in $\mathcal{U}$ that only affects U_i , we show that faithful counterfactual transformation $M_i(X)$ is a disentangled intervention with respect to U_i , hence completing the proof.

With this theory, faithfulness=disentangled intervention. In Section 3.1, we train M_i to ensure $M_i(\mathbf{x}_s) \sim P(X_t)$ (faithfulness) for every sample $\mathbf{x}_s$ in $S = s$, hence encouraging M_i to be a disentangled intervention. Note that the above analysis can easily generalize to M_i^{-1} .

Sufficient Condition

We will prove the following sufficient condition: *if (M_i, M_i^{-1}) intervenes U_i , the i -th mapping function outputs the counterfactual faithful generation, i.e., the smallest $\mathcal{L}_{CycleGAN}^i$.*

Without loss of generality, we will prove for the $S = s \rightarrow S = t$ mapping M_i , which can be extended to M_i^{-1} . For a sample $\mathbf{x}_s$ in $S = s$, let $g^{-1}(\mathbf{x}_s) = \mathbf{u} = (u_1, \dots, u_i, \dots, u_k)$. We modify U_i by changing u_i to a value $\hat{u}_i$ drawn from $P(U_i|S = t)$. Denote the modified attribute as $\hat{\mathbf{u}} = (u_1, \dots, \hat{u}_i, \dots, u_k)$. Denote the sample with attribute $\hat{\mathbf{u}}$ as $\hat{\mathbf{x}}$. Given M_i intervenes U_i , $M_i(\mathbf{x}_s)$ corresponds to a counterfactual outcome when U_i is set to $\hat{u}_i$ through intervention (or U set as $\hat{\mathbf{u}}$). Now as $g^{-1}(\hat{\mathbf{x}}) = \hat{\mathbf{u}}$, using the *counterfactual consistency rule* [40], we have $M_i(\mathbf{x}_s) = \hat{\mathbf{x}}$. As $\hat{\mathbf{x}}$ is faithful with the Counterfactual Faithfulness theorem, we prove that $M_i(\mathbf{x}_s)$, i.e., the output of the i -th mapping function, is also faithful, i.e., the smallest $\mathcal{L}_{CycleGAN}^i$.

Proxy Function Theorem

We will derive for the general case where $\tilde{X}$ is any continuous proxy. We will assume that the confounder U follows the completeness condition in [232], which

accommodates most commonly-used parametric and semi-parametric models such as exponential families.

Given $h_y(X, \hat{X})$ solves Eq. (3), we have:

$$\begin{aligned} P(Y|Z, X, S) &= \int_{-\infty}^{+\infty} h_y(X, \hat{X}) P(\hat{X}|Z, X, S) d\hat{X} \\ &= \int_{-\infty}^{+\infty} h_y(X, \hat{X}) \left\{ \int_{-\infty}^{+\infty} P(\hat{X}|U) P(U|Z, X, S) dU \right\} d\hat{X}. \end{aligned} \quad (3.28)$$

From the law of total probability, we have:

$$P(Y|Z, X, S) = \int_{-\infty}^{+\infty} P(Y|U, X) P(U|Z, X, S) dU. \quad (3.29)$$

With Eq. (3.28) and Eq. (3.29) and the completeness condition, we have:

$$P(Y|U, X) = \int_{-\infty}^{+\infty} h_y(X, \hat{X}) P(\hat{X}|U) d\hat{X}, \quad (3.30)$$

which proves the Proxy Function theorem.

From Eq. (3.30), we have:

$$\begin{aligned} &\int_{-\infty}^{+\infty} h_y(X, \hat{X}, S = s) P(\hat{X}|U) d\hat{X} \\ &= \int_{-\infty}^{+\infty} h_y(X, \hat{X}, S = t) P(\hat{X}|U) d\hat{X}. \end{aligned} \quad (3.31)$$

Hence from the completeness condition, we have $h_y(X, \hat{X}, S = s) = h_y(X, \hat{X}, S = t)$ [232]. Hence we prove that $h_y(X, \hat{X})$ is domain-agnostic.

Note that in Section 3.3.3, our proxy $\hat{X}$ is a continuous random variable takes values from $\{M_i(\mathbf{x}_s)\}_{i=1}^k$ for sample $\mathbf{x}_s$ in $S = s$ or $\{M_i^{-1}(\mathbf{x}_t)\}_{i=1}^k$ for sample $\mathbf{x}_t$ in $S = t$. This is a special case of the analysis above with the probability mass of $\hat{X}$ centers around the set of its possible values.

Derivation of Eq. 3.21

We derive Eq. 3.21 as a corollary to [232]. The goal is to solve for $h_y(X, \hat{X})$ under the function form in Eq. (6) from the formula below:

$$P(Y|Z, X, S = s) = \int_{-\infty}^{\infty} P(\hat{X}|Z, X) h_y(X, \hat{X}) d\hat{X}. \quad (3.32)$$

For simplicity, we define a standard multivariate Gaussian function $\phi(\cdot)$. If $A \sim \mathcal{N}(0, \mathbf{I})$ and $A \in \mathbb{R}^n$, we have:

$$P(A) = \phi(A) = \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2} A^T A\right). \quad (3.33)$$

Our function form for $P(Y|Z, X, S = s)$ is given by $\mathcal{N}(\mathbf{b}_1 + \mathbf{W}_1 Z + \mathbf{W}_2 X, \mathbf{\Sigma}_1)$ and for $P(\hat{X}|Z, X, S = s)$ is given by $\mathcal{N}(\mathbf{b}_2 + \mathbf{W}_3 Z + \mathbf{W}_4 X, \mathbf{\Sigma}_2)$, where the variance terms are omitted in the main text for brevity, as the final results only depend on the means. Specifically, $\mathbf{\Sigma}_2 \in \mathbb{R}^{n \times n}$ is a symmetrical matrix with Eigen-decomposition given by $\mathbf{\Sigma}_2 = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$, where $\mathbf{U} \in \mathbb{R}^{n \times n}$ is a full-rank matrix containing the eigen-vectors and $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with eigen-values. We define $\mathbf{B} = \mathbf{U} \mathbf{\Lambda}^{\frac{1}{2}}$, and hence $\mathbf{\Sigma}_2 = \mathbf{B} \mathbf{B}^T$. We can rewrite $P(\hat{X}|Z, X, S = s)$ as:

$$P(\hat{X}|Z, X, S = s) = |\mathbf{B} \mathbf{B}^T|^{-\frac{1}{2}} \phi(\mathbf{B}^{-1}(\hat{X} - \hat{\boldsymbol{\mu}})), \quad (3.34)$$

where $\hat{\boldsymbol{\mu}} = \mathbf{b}_2 + \mathbf{W}_3 Z + \mathbf{W}_4 X$. Define $Z' = \mathbf{B}^{-1} \hat{\boldsymbol{\mu}}$ and $\hat{X}' = \mathbf{B}^{-1} \hat{X}$. We define

$$t_y(Z', X) = P(Y|Z = \mathbf{W}_3^+ (\mathbf{B} Z' - \mathbf{W}_4 X - \mathbf{b}_1), X, S = s). \quad (3.35)$$

Now we can solve h_y from

$$t_y(Z', X) = \int_{-\infty}^{\infty} \phi(\hat{X}' - Z') h_y(X, \mathbf{B} \hat{X}') d\hat{X}'. \quad (3.36)$$

Specifically, let h_1 and h_2 represent the Fourier transform of ϕ and t_y , respectively:

$$\begin{aligned} h_1(\boldsymbol{\nu}) &= \int_{-\infty}^{\infty} \exp(-i \boldsymbol{\nu} Z') \phi(Z') dZ' \\ &= \int_{-\infty}^{\infty} \exp(-i \boldsymbol{\nu} Z) \phi(Z) dZ \end{aligned} \quad (3.37)$$

$$h_2(\boldsymbol{\nu}, X, Y) = \int_{-\infty}^{\infty} \exp(-i \boldsymbol{\nu} Z') t_y(Z', X) dZ', \quad (3.38)$$

where $i = (-1)^{\frac{1}{2}}$ is the imaginary unity. Substituting Eq. (3.36) (3.37) into Eq. (3.38), we have:

$$h_2(\boldsymbol{\nu}, X, Y) = h_1(\boldsymbol{\nu}) \int_{-\infty}^{\infty} \exp(-i\boldsymbol{\nu}\hat{X}') h_y(X, \mathbf{B}\hat{X}') d\hat{X}'. \quad (3.39)$$

Hence

$$\int_{-\infty}^{\infty} \exp(-i\boldsymbol{\nu}\hat{X}') h_y(X, \mathbf{B}\hat{X}') d\hat{X}' = \frac{h_2(\boldsymbol{\nu}, X, Y)}{h_1(\boldsymbol{\nu})}. \quad (3.40)$$

By Fourier inversion, we have:

$$h_y(X, \mathbf{B}\hat{X}') = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp(-i\boldsymbol{\nu}\hat{X}') \frac{h_2(\boldsymbol{\nu}, X, Y)}{h_1(\boldsymbol{\nu})} d\boldsymbol{\nu} \quad (3.41)$$

By substituting $\hat{X} = \mathbf{B}\hat{X}'$, we have

$$h_y(X, \hat{X}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp(-i\boldsymbol{\nu}\mathbf{B}^{-1}\hat{X}) \frac{h_2(\boldsymbol{\nu}, X, Y)}{h_1(\boldsymbol{\nu})} d\boldsymbol{\nu} \quad (3.42)$$

Solving for Eq. (3.42) yields:

$$\begin{aligned} h_y(X, \hat{X}) &= \mathbf{b}_1 - \mathbf{W}_1 \mathbf{W}_3^+ \mathbf{b}_2 + \mathbf{W}_1 \mathbf{W}_3^+ \hat{X} \\ &\quad + (\mathbf{W}_2 - \mathbf{W}_1 \mathbf{W}_3^+ \mathbf{W}_4) X, \end{aligned} \quad (3.43)$$

where scaling terms not related with $X, \hat{X}$ are dropped as they do not impact inference. This completes the derivation.

Training Details. The networks are randomly initialized by sampling from normal distribution with standard deviation of 0.02. We use Adam optimizer [250] with initial learning rate 0.0002, beta1 as 0.5 and beta2 as 0.999. We train the network with the initial learning rate for 100 epochs and decay the learning rate linearly for another 100 epochs on ImageCLEF-DA [9] and OfficeHome [7]. We train the network with the initial learning rate for 20 epochs and decay the learning rate linearly for another 20 epochs on VisDA-2017 [3]. The CycleGAN loss and discriminator loss are updated iteratively. The adopted competitive training scheme to train DCMs can sometimes be sensitive to network initialization. Hence all DCMs are first trained for 8000 iterations, before using the competitive training scheme.

Detailed Discussion on t-SNE Plot. Note that the proxy loss in Section 4

aligns the proxy features with the sample features in the counterpart domain, which is different from existing methods that try to align X_s and X_t directly. Our inference uses the proxy function implemented with Eq. (7), where in training, X takes values of X_s and $\hat{X}$ takes values of $M_i(X_s)$, in testing, X takes values of X_t and $\hat{X}$ takes values of $M_i^{-1}(X_t)$. While X_s and X_t (or $M_i(X_s)$ and $M_i^{-1}(X_t)$) are not aligned in TCM, we can still achieve competitive performance by finding how to transport with a causality theoretic viewpoint of the domain shift.

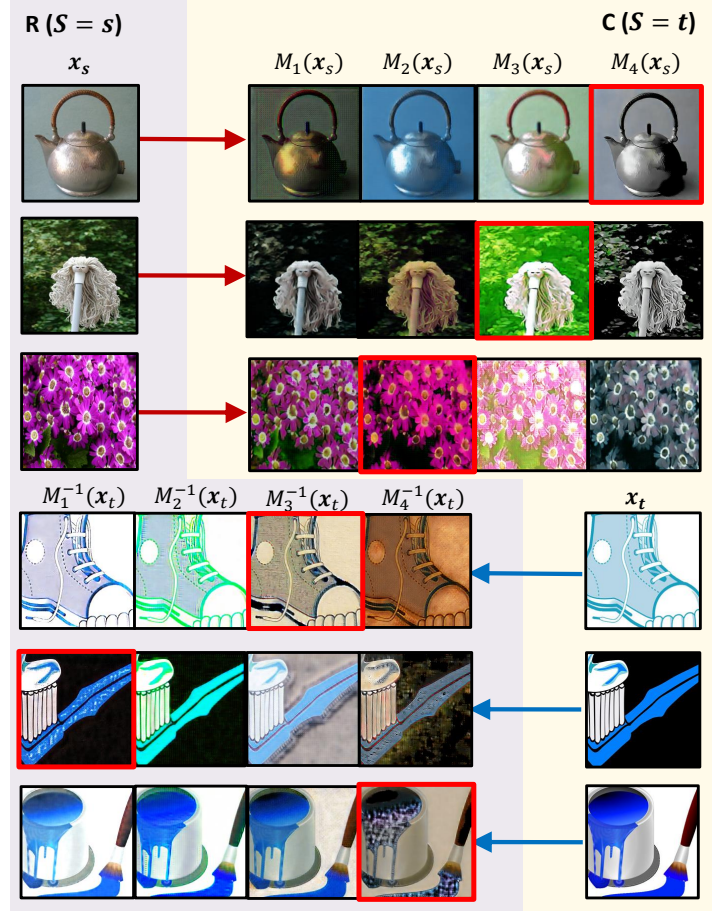


FIGURE 3.17: Supplementary to Figure 4. Transformation between “Real World” (R) and “Clipart” (C) domain with 4 trained DCMs $\{(M_i, M_i^{-1})\}_{i=1}^4$. The winning DCM is outlined in red.

Additional Visualizations of DCMs Outputs. In Figure 3.17, we show additional generated images from DCMs in the “Real World” (R) $\rightarrow$ “Clipart” (C) task of OfficeHome [7], which also has a large domain gap. $\{M_1, \dots, M_4\}$ (or $\{M_1^{-1}, \dots, M_4^{-1}\}$) roughly correspond to reducing (increasing) brightness, changing color, increasing (decreasing) saturation and removing (adding) color.

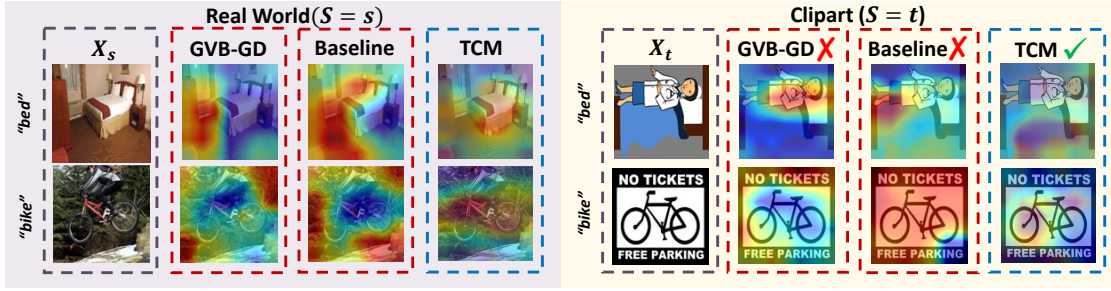


FIGURE 3.18: Class Activation Maps (CAMs) [5] of GVB-GD [10], Baseline and TCM on the Real World($S=s$) $\rightarrow$ Clipart($S=t$) task in Office-Home dataset [7]. The left column shows two samples in each domain, whose class name is indicated on the top. In $S=t$, we show two samples predicted wrongly by GVB-GD and Baseline, but correctly with TCM. Supplementary to Figure 8

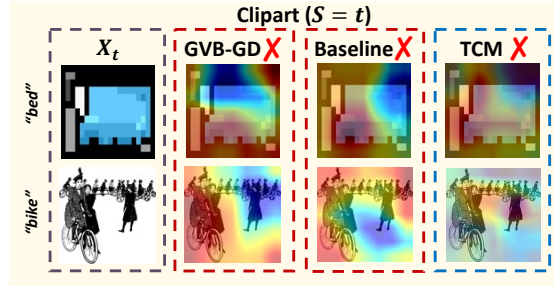


FIGURE 3.19: Class Activation Maps (CAMs) [5] of GVB-GD [10], Baseline and TCM on the Real World($S=s$) $\rightarrow$ Clipart($S=t$) task in Office-Home dataset [7]. The left column shows two samples in each domain, whose class name is indicated on the left. All methods fail on the two samples. Supplementary to Figure 8.

Additional CAM Responses. In Figure 3.18, we show the CAM responses in the “Real World” $\rightarrow$ “Clipart” task, where GVB-GD and baseline focuses on the background to distinguish “bed” and “bike” and the shape semantic is lost, leading to poor generalization in $S=t$. While TCM preserves the foreground object shape semantic. In Figure 3.19, all three methods fail on the two samples from $S=t$. On the “bed” sample, all methods indeed focus on the object. However, the object itself is not very discriminative, which may explain the failure. On the “bike” sample, the object is small and all methods fail to distinguish it from the context.

Convergence Speed, GFLOPS, #Parameters. The convergence speeds of TCM and related methods are shown in Figure 3.20. Our TCM (red) converges slightly slower in early iterations (*e.g.*, before 20 on ImageCLEF-DA and 250 on Office-Home) due to the training of VAE in Eq. 9 and the linear layers in Eq. 6. After that, TCM converges to the best performance. The GFLOPS and #Parameters are shown in Table 3.9. All methods use the same backbone and the backbone

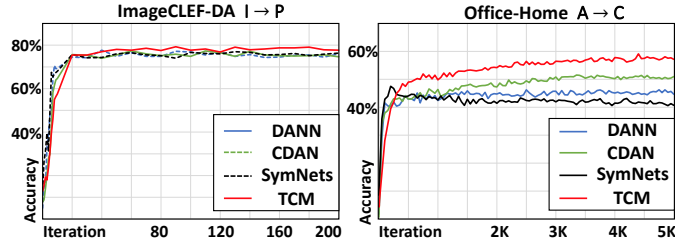


FIGURE 3.20: UDA accuracy (%) using DANN, CDAN, SymNets and our TCM, on different training iterations. Batch size is 32.

Method	Backbone GFLOPS	Additional GFLOPS	#Parameters (M)
DANN	133	0.084	1.3
CDAN	133	1.158	18.1
GVB-GD	133	0.092	1.4
Baseline	133	0.072	1.1
TCM	133	0.364	10.1

TABLE 3.9: GFLOPS of feature extractor backbone and additional modules (*e.g.*, discriminator networks) as well as #parameters of those additional modules on Office-Home.

fine-tuning takes the major time cost (133), compared with the GFLOPS from additional modules. Regarding the parameters, our TCM has more (but not the most) because of the additional VAE and linear layers.

3.4 Conclusion

In Section 3.2, we presented a novel causal framework: Interventional Few-Shot Learning (IFSL), to address an overlooked deficiency in recent FSL methods: the pre-training is a confounder hurting the performance. Specifically, we proposed a structural causal model of the causalities in the process of FSL and then developed three practical implementations based on the backdoor adjustment. To better illustrate the deficiency, we diagnosed the classification accuracy comprehensively across different similarities between the few-shot training set and test set. Particularly, a lower similarity means that the FSL task is harder. We showed that IFSL improves all the baselines across all hardness. It is worth highlighting that the contribution of IFSL is not only about improving the performance of FSL, but also offering a causal explanation for why IFSL works well: it is a causal approximation to many-shot learning. We believe that IFSL may shed light on exploring the new

boundary of FSL, even though FSL is well-known to be ill-posed due to insufficient data.

In Section 3.3, we presented a practical implementation of the transportability theory for UDA called Transporting Causal Mechanisms (TCM). It systematically eliminates the semantic loss from the confounding effect. In particular, we proposed to identify the confounder stratification by discovering disentangled causal mechanisms and represent the unknown confounder representation by proxy variables. Extensive results on UDA benchmarks showed that TCM is empirically effective. It is worth highlighting that TCM tackles UDA from a completely new perspective—instead of making the two domains similar (as in existing methods), we aim to find how to transport and what can be transported with a causality theoretic viewpoint of the domain shift.

In the next Chapter, we study how to mitigate spurious correlation without relying on information about the confounder, and we show it is possible when diverse unlabeled data is available.

Chapter 4

De-confound by Invariant Learning ¹

4.1 Preliminaries

Given a partition of the training dataset into subsets and a loss function, the goal of invariant learning is to learn a model that simultaneously minimizes the loss in all the subsets of the partition. The basic premise is that, when the spurious correlation *differs* in the subsets (*e.g.*, digit 0 in red only in one subset), a model that leverages the correlation cannot be optimal across *all* subsets. In particular, the model consists of a feature backbone $\Phi(\mathbf{x})$ that outputs a feature vector given a sample $\mathbf{x}$, and a classifier f that outputs the prediction logits given a feature.

Partition. We specify a partition of the N training samples into K subsets by the partition matrix $\mathbf{P} \in \{0, 1\}^{N \times K}$, where $P_{i,k} = 1$ if the i -th identity belongs to the k -th subset, and 0 otherwise.

Loss. The loss is typically classification loss, *e.g.*, cross-entropy or supervised contrastive [251]. Given a feature extractor Φ and classifier f , we denote the loss in the k -th subset of $\mathbf{P}$ as $\mathcal{L}_{cls}(\Phi, f, \mathbf{P}, k)$.

¹The work in this chapter is in the paper: **Zhongqi Yue**, Qianru Sun, Xian-Sheng Hua, Hanwang Zhang. Make the U in UDA Matter: Invariant Consistency Learning for Unsupervised Domain Adaptation. 2022. and **Zhongqi Yue**, Xinting Hu, Yew Soon Ong, Qianru Sun, Hanwang Zhang. Debaised Pseudo-Labeling for Semi-Supervised Semantic Segmentation. 2023.

Objective. Invariant learning uses the following objective, called the invariant risk minimization (IRM) [43]:

$$\begin{aligned} & \min_{\Phi, f} \sum_{k=1}^K \mathcal{L}_{cls}(\Phi, f, \mathbf{P}, k), \\ \text{s.t. } & f \in \bigcap_{k=1}^K \underset{f^*}{\operatorname{argmin}} \mathcal{L}_{cls}(\Phi, f^*, \mathbf{P}, k). \end{aligned} \quad (4.1)$$

The constrained optimization minimizes the combined classification loss (first line) while maintaining an invariant classifier f by keeping the losses in all subsets simultaneously optimal (second line). This objective learns the causal feature invariant across the subsets. For example, in Figure 1.9, while classifying dark digits as “0” and light digits as “1” can reduce $\mathcal{L}_{cls}$ in the 1st subset (*i.e.*, red samples), it hurts the loss in the 2nd subset (*i.e.*, blue samples). Therefore, the spurious digit color will be removed to satisfy the invariance constraint.

To avoid the challenging bi-level optimization in Eq. (4.1), we use two practical implementations—IRMv1 [43] and REx [252], which merge the classification loss and the constraint as a single invariant loss, denoted as $\mathcal{L}_{inv}(\Phi, f, \mathbf{P})$. The IRMv1 objective is given by

$$\mathcal{L}_{inv}(\Phi, f, \mathbf{P}) = \sum_{k=1}^K \mathcal{L}_{cls}(\Phi, f, \mathbf{P}, k) \quad (4.2)$$

$$+ \lambda \|\nabla_{\mathbf{w}=1.0} \mathcal{L}_{cls}(f \circ \Phi, \mathbf{w}, \mathbf{P}, k)\|_2^2, \quad (4.3)$$

and the REx objective is given by

$$\mathcal{L}_{inv}(\Phi, f, \mathbf{P}) = \sum_{k=1}^K \mathcal{L}_{cls}(\Phi, f, \mathbf{P}, k) \quad (4.4)$$

$$+ \lambda \operatorname{Var}(\{\mathcal{L}_{cls}(\Phi, f, \mathbf{P}, k)\}_{k=1}^K), \quad (4.5)$$

where λ is the invariant penalty strength, and $\operatorname{Var}(\cdot)$ computes the variance of a set of losses. Intuitively, the second term in IRMv1 [43] treats the logits produced by the composition $f \circ \Phi$ as a feature and computes the optimality of a baseline classifier $\mathbf{w} = \mathbf{1.0}$. Hence IRMv1 achieves invariance by encouraging a fixed baseline to be simultaneously optimal in all subsets. In contrast, REx minimizes the variance of all subset losses to achieve invariance.

In this chapter, we reveal that when unlabeled data is available in Unsupervised

Domain Adaptation (UDA) or Semi-Supervised Learning (SSL), we can use labeled and unlabeled data to form two environments, enabling invariant learning to mitigate spurious correlations.

4.2 Case Study: Unsupervised Domain Adaptation

4.2.1 Introduction

Domain Adaptation (DA) is all about training a model in a labelled source domain S (*e.g.*, an autopilot trained in daytime), and the model is expected to generalize in target domain T (*e.g.*, the autopilot deployed in nighttime), where T and S has a significant domain shift (*e.g.*, day *vs.* night) [36]. To illustrate the shift and generalization, we introduce the classic notion of causal representation learning [38, 39]: any sample is generated by $\mathbf{x} = \Phi(\mathbf{c}, \mathbf{e})$, where Φ is the generator, $\mathbf{c}$ is the causal feature determining the domain-invariant class identity (*e.g.*, road lane), and $\mathbf{e}$ is the environmental feature (*e.g.*, lighting conditions). As the environment is domain-specific, the domain shift $\mathbf{e}_S \neq \mathbf{e}_T$ results in $\mathbf{x}_S \neq \mathbf{x}_T$, challenging the pursuit of domain-invariant class model. However, if a model disentangles the causal feature by transforming each $\mathbf{x} = \mathbf{c}$, it can generalize under arbitrary shift in $\mathbf{e}$. In Figure 4.1, even though $\mathbf{e}_T = \text{“colorful”}$ is significantly different from $\mathbf{e}_S = \text{“red”}$, thanks to the non-discriminative “red” color in S , the model still easily disentangles $\mathbf{c}$ out of $\Phi(\mathbf{c}, \mathbf{e}_S)$ and thus generalizes to digits of any color.

Unfortunately, in general, disentangling $\mathbf{c}$ by using only S is impossible as $\mathbf{e}_S$ can be also discriminative, *i.e.*, spuriously correlated with $\mathbf{c}$. In Figure 4.2a, color $\mathbf{e}_S$ is correlated with shape $\mathbf{c}$, *i.e.*, $\mathbf{c}_1 = \text{“0”}$ and $\mathbf{c}_2 = \text{“1”}$ in S tend to have $\mathbf{e}_1 = \text{dark}$ and $\mathbf{e}_2 = \text{light}$ digit colors, respectively. As both $\mathbf{c}$ and $\mathbf{e}_S$ are discriminative, a classifier trained in S will inevitably capture both (see the red line in Figure 4.2a). This leads to poor generalization when the correlation between $\mathbf{e}_T$ and $\mathbf{c}$ is different, *i.e.*, the colors of $\mathbf{c}_1 = \text{“0”}$ and $\mathbf{c}_2 = \text{“1”}$ in T tend to be $\mathbf{e}_2 = \text{light}$ and $\mathbf{e}_1 = \text{dark}$ instead.

To avoid the spurious correlations between $\mathbf{e}_S$ and $\mathbf{c}$, a practical setting called Unsupervised DA (UDA) introduces additional information of T through a set

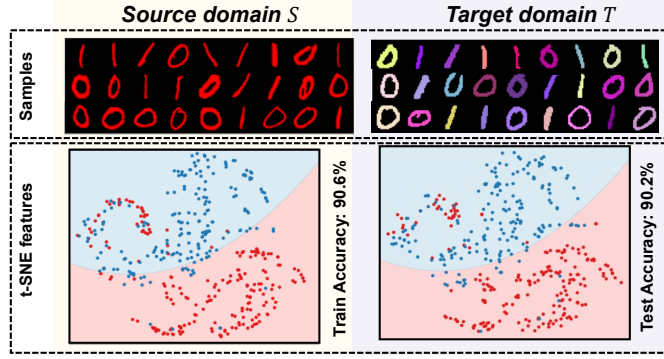


FIGURE 4.1: We color the “0”s and “1”s in MNIST [1] differently to form S and T , and train a linear classifier in S . Features are visualized by t-SNE [2], where $\bullet$ / $\bullet$ denotes “0” / “1”. The classification boundary is rendered blue and red.

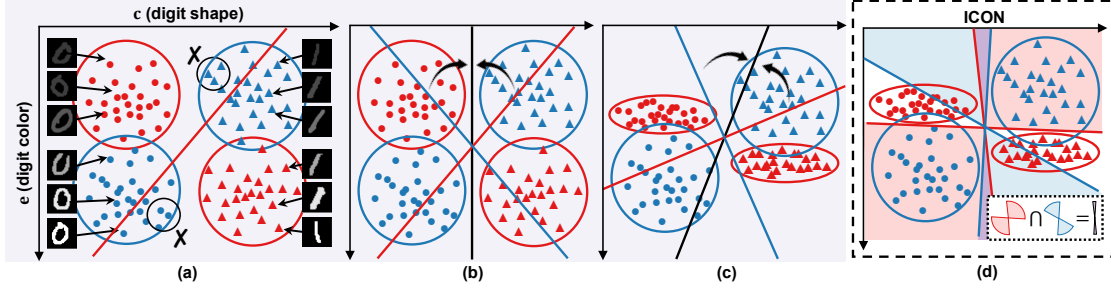


FIGURE 4.2: Red: S , Blue: T . $\bullet$: “0”, $\blacktriangle$: “1”. (a) The red line denotes the classifier trained in S . (b) Classifier (black line) can be corrected by respecting the BCE loss between S (red line) and T (blue line) (c) A failure case of minimizing the combined BCE losses in S and T (black line is wrongly tilted). Due to the low variance of $\mathbf{e}_S$ in each class, the optimal classifier *w.r.t.* the BCE loss in S (red line) looks “flat”, *i.e.*, it has large weight on $\mathbf{e}$ [11]. (d) Our ICON prevents such a failure case (the intersection area of the red and blue candidate classifier regions). Best viewed in color.

of unlabelled samples [100, 241, 253]. There are two main lineups: 1) Domain-alignment methods [94, 97, 254] learn the common feature space of S and T that minimizes the classification error in S . In Figure 4.2a, their goal is to unify the “0”s and “1”s across S and T . However, they suffer from the well-known misalignment issue [247, 255], *e.g.*, an alternative solution in Figure 4.2a also satisfies the training objective, where “0”s and “1”s in T are collapsed to “1”s and “0”s in S , respectively, but it generalizes poorly to T . 2) Self-training methods [256–258] use a classifier trained in S to pseudo-label samples in T , and jointly train the model with the labels in S and confident pseudo-labels in T . Yet, the pseudo-labels can be unreliable even with expensive threshold tweaking [12, 117], *e.g.*, the X areas in Figure 4.2a have confident but wrong pseudo-labels. In fact, the recent WILDS 2.0 benchmark [24] on real-world UDA tasks shows that both lineups even

under-perform an embarrassingly naive baseline: directly train in S and test in T .

Notably, although both methods respect the idiosyncrasies in S (by minimizing the classification error), they fail to account for the inherent distribution in T , *e.g.*, in Figure 4.2a, even though T is unlabelled, we can still identify the two sample clusters in T (enclosed by the blue circles), which provide additional supervision for the classifier trained in S . In particular, the classifier (red line) in Figure 4.2a breaks up the two clusters in T , showing that the correlation between color $\mathbf{e}_S$ and $\mathbf{c}$ in S is inconsistent with the clusters affected by $\mathbf{e}_T$ in T , and thus implying that color $\mathbf{e}$ is the environmental feature. Hence, to make the U in UDA matter, we aim to learn a classifier that is consistent with classification in S and clustering in T :

Consistency. We use the binary cross-entropy (BCE) loss to encourage the prediction similarity of each sample pair from the same class of S or cluster of in T , and penalize the pairs from different ones. In Figure 4.2b, the red and blue lines denote the classifiers that minimize the BCE loss in S and T , respectively. By minimizing the combined loss, we learn the classifier consistently with the sample distribution in both domains (the upright black line), which discards the environmental feature $\mathbf{e}$ by disentangling $\mathbf{c}$ from $\Phi(\mathbf{c}, \mathbf{e})$.

Invariance. However, enforcing consistency is a tug-of-war between S and T , and the BCE loss in one domain can dominate the other to cause failures. Without loss of generality, we consider the case where S dominates in Figure 4.2c. Due to the strong correlation between $\mathbf{e}_S$ and $\mathbf{c}$, the classifier minimizing the combined BCE losses (black line) deviates from the desired upright position as in Figure 4.2b. To tackle this, we aim to learn an invariant classifier [43] that is simultaneously optimal *w.r.t.* the BCE loss in S and T . In Figure 4.2d, for each of the two losses, we color the candidate regions where the classifier is optimal (*i.e.*, gradients ≈ 0) with red and blue, respectively. In particular, by learning an invariant classifier lying in the intersection (purple region), we successfully recover the desired one as in Figure 4.2b, even when one domain dominates.

Overall, we term our approach as **Invariant CONSistency learning (ICON)**. Our contributions are given below:

- ICON is a novel UDA method orthogonal to existing ones (hence can be combined with them). It is simple to implement (Section 4.2.4), and disentangles the

causal feature $\mathbf{c}$ from $\Phi(\mathbf{c}, \mathbf{e})$ that generalize to T with a theoretical guarantee (Section 4.2.5).

- ICON significantly improves the current state-of-the-art on classic UDA benchmarks [117, 259] with ResNet-50 [22] as the feature backbone: 75.8% (+2.6%) accuracy averaging over the 12 tasks in OFFICE-HOME [7], and 87.4% (+3.7%) mean-class accuracy on the Synthetic→Real task in VisDA-2017 [3] (Section 4.2.6).
- Notably, as of thesis submission, ICON dominates the leaderboard of the challenging WILDS 2.0 UDA benchmark [24] (<https://wilds.stanford.edu/leaderboard/>), which includes 8 large-scale classification, regression and detection tasks on image, text and graph data (Section 4.2.6).

4.2.2 Problem Formulations

UDA aims to learn a model with the labelled training data $\{\mathbf{x}_i, y_i\}_{i=1}^N$ in source S and unlabelled $\{\mathbf{x}_i\}_{i=1}^M$ in target T , where $\mathbf{x}_i$ denotes the feature of the i -th sample extracted by a backbone parameterized by θ (e.g., ResNet-50 [22]), and y_i is its ground-truth label. We drop the subscript i for simplicity when the context is clear. As shown in Figure 4.3, our model includes the backbone θ , a classification head f and a cluster head g , where f and g output the softmax-normalized probability for each class and cluster, respectively. Note that f and g have the same output dimension as the class numbers are the same in S and T . As discussed in Section 4.2.1, the generalization objective of ICON is to learn an invariant f that is simultaneously consistent with the classification in S , and the clustering in T , identified by g .

Consistency. We enforce the consistency of f with the binary cross-entropy (BCE) loss in S and T , denoted as $\text{BCE}(S, f)$ and $\text{BCE}(T, f)$, respectively. For example, in contrast to the cross-entropy (CE) loss that is sample-wise, $\text{BCE}(S, f)$ is a pair-wise loss given by:

$$\text{BCE}(S, f) = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} b \log(\hat{y}) + (1 - b) \log(1 - \hat{y}), \quad (4.6)$$

where $\hat{y} = f(\mathbf{x}_i)^T f(\mathbf{x}_j)$ is the predictive similarity, and $b = \mathbb{1}(y_i = y_j)$ is a binary label indicating if the pair is from the same class. Intuitively, BCE loss clusters

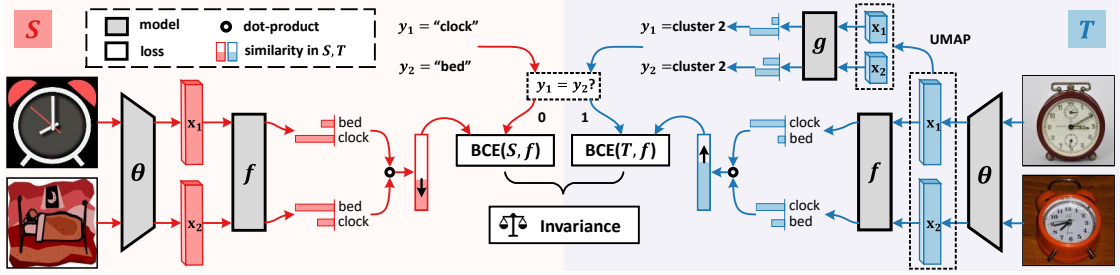


FIGURE 4.3: S : Clipart and T : Real-world from OFFICE-HOME [7] dataset. Without loss of generality, we give an example where the pair of samples is from different classes in S and the same cluster in T , respectively. The black arrows on the similarity bars denotes that minimizing the BCE losses in S and T will decrease the similarity in S (as $y_1 \neq y_2$) and increase that in T (as $y_1 = y_2$).

sample pairs with the same label by increasing the dot-product similarity of their predictions, and pushes away pairs with different labels by decreasing the similarity. In practice, BCE loss is computed within each mini-batch, hence there is no N^2 prohibitive calculation. However, $\text{BCE}(T, f)$ cannot be evaluated directly as T is unlabelled. So the first step is to cluster T and use the cluster label to compute $\text{BCE}(T, f)$.

4.2.3 Step 1: Identify Clusters in T

Dimensionality Reduction. To assist clustering, we pre-process the features in T by dimensionality reduction as a standard technique [2, 260] to highlight the causal feature $\mathbf{c}$ by suppressing any environmental feature $\mathbf{e}_T$ that is less discriminative (e.g., $\mathbf{e}_T$ = “colorful” in Figure 4.1). We use UMAP [261].

Generating Cluster Labels. The pre-processed features are used for clustering in T . We adopt the rank-statistics algorithm [262], as it can be easily implemented in a mini-batch sampling manner for online clustering. The algorithm trains g to cluster sample features whose top values have the same indices. After training, the predictions from g are used as the cluster labels to form $\{\mathbf{x}_i, y_i\}_{i=1}^M$ in T , where $y_i = \arg\max g(\mathbf{x}_i)$ is the cluster with the highest predicted probability, and $\mathbf{x}_i$ denotes the i -th pre-processed feature with a slight abuse of notation. This enables us to compute $\text{BCE}(T, f)$.

Important Highlights.

- In general, one can use any clustering algorithm that can specify the number of output clusters as the class numbers, *e.g.*, through the output dimensions of g in the rank-statistics algorithm or k in k -means. This is because the ground-truth class numbers in T is an important prior to the success of ICON (see Section 4.2.5).
- Note that the original feature $\mathbf{x}$ from the backbone is used to compute $\text{BCE}(T, f)$ instead of the pre-processed one, as f is incompatible with the reduced feature dimensions.
- The cluster label y in T only determines if the paired samples have the same label for $\text{BCE}(T, f)$. It does not necessarily align with the class index in S .

4.2.4 Step 2: Invariant Consistency (ICON)

ICON Objective enforces the supervision in S and the consistency in S and T with an invariant constraint [43]:

$$\begin{aligned}
 & \min_{\theta, f} \overbrace{\text{BCE}(S, f) + \text{BCE}(T, f)}^{\text{Consistency}} + \overbrace{\text{CE}(S, f)}^{S\text{-Supervision}} + \alpha \mathcal{L}_{st} \\
 & \text{s.t. } \underbrace{f \in \text{argmin}_{\bar{f}} \text{BCE}(S, \bar{f}) \cap \text{argmin}_{\bar{f}} \text{BCE}(T, \bar{f})}_{\text{Invariance}}. \tag{4.7}
 \end{aligned}$$

The computation of this objective is illustrated in Figure 4.3, where the standard S -supervision is omitted. We explain the two lines in Eq. (4.7) separately:

Line 1. The BCE losses train f to be consistent with the pair-wise label similarity in S and T . CE loss in S trains f to assign each sample its label (*e.g.*, predicting a clock feature as “clock”). $\mathcal{L}_{st}$ is the self-training loss with weight α , which leverages the learned invariant classifier to generate accurate pseudo-labels in T for additional supervision.

Line 2. The invariance below requires the backbone θ to elicit an invariant classifier f that simultaneously minimizes the BCE loss in S and T . This constraint prevents the case where one BCE loss is minimized at the cost of increasing the other to pursue a lower sum of the two (Figure 4.2c). We implement the constraint with REx [252].

Training and Testing. Overall, at the start of training, the backbone θ is initialized from a pre-trained weight (*e.g.*, on ImageNet [23]) and f, g are randomly

Algorithm 3 ICON Training (1 epoch)

-
- 1: **Input:** Labelled samples in S , unlabelled samples in T , feature backbone parameterized by θ , the classifier f and cluster head g , batch size b .
 - 2: **Output:** θ, f, g trained for 1 epoch.
 - 3: Compute $\{\mathbf{x}_i\}_{i=1}^M$ in T (features extracted by θ)
 - 4: Pre-process features in T with UMAP
 - 5: Train g using the pre-processed features
 - 6: **repeat**
 - 7: Sample $\{\mathbf{x}_i, y_i\}_{i=1}^b$ in S (features extracted by θ)
 - 8: Compute $\text{CE}(S, f), \text{BCE}(S, f)$
 - 9: Sample $\{\mathbf{x}_i\}_{i=1}^b$ in T (features extracted by θ)
 - 10: Assign $y_i \leftarrow \arg\max g(\mathbf{x}_i)$ for $i \in \{1, \dots, b\}$
 - 11: Compute $\text{BCE}(T, f)$
 - 12: Optimize θ, f with Eq. (4.7)
 - 13: **until** end of epoch
-

initialized. Then ICON is trained by iterating Algorithm 3 until convergence. Only the backbone θ and the classifier f are used in testing.

4.2.5 Invariant Learning in UDA

Preliminaries. We adopt the group-theoretic definition of disentanglement [38, 263] to justify ICON. We start by introducing some basic concepts in group theory. $\mathcal{G}$ is a group of semantics that generate data by group action, *e.g.*, a “turn darker” element $g \in \mathcal{G}$ transforms $\mathbf{x}$ from white “0” to $g \circ \mathbf{x}$ as grey “0” (bottom-left to top-left in Figure 4.4b). A sketch of theoretical analysis is given below and interested readers are encouraged to read the full proof in Section 4.2.7.

Definition (Generalization). *Feature space $\mathcal{X}$ generalizes under the direct product decomposition $\mathcal{G} = \mathcal{G}/\mathcal{H} \times \mathcal{H}$, if $\mathcal{X}$ has a non-trivial $\mathcal{G}/\mathcal{H}$ -representation, i.e., the action of each $h \in \mathcal{H}$ corresponds to a trivial linear map in $\mathcal{X}$ (i.e., identity map), and the action of each $g \in \mathcal{G}/\mathcal{H}$ corresponds to a non-trivial one.*

This definition formalizes the notion of transforming each sample $\Phi(\mathbf{c}, \mathbf{e})$ to the causal feature $\mathbf{c}$. Subgroup $\mathcal{H} \subset \mathcal{G}$ and quotient group $\mathcal{G}/\mathcal{H}$ transform the environmental feature $\mathbf{e}$ and the causal $\mathbf{c}$, respectively. If $\mathcal{X}$ has a non-trivial $\mathcal{G}/\mathcal{H}$ -representation, samples in a class are transformed to the same feature $\mathbf{c} \in \mathcal{X}$ regardless of $\mathbf{e}$ (i.e., representation of $\mathcal{G}/\mathcal{H}$), and different classes have different $\mathbf{c}$ (i.e., non-trivial). This definition is indeed in accordance with the common view of

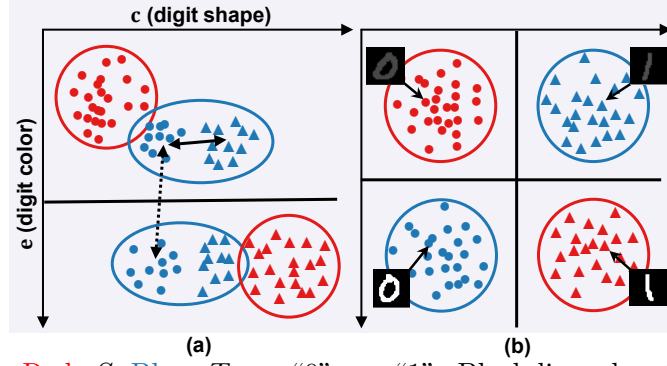


FIGURE 4.4: Red: S , Blue: T . $\bullet$: “0”, $\blacktriangle$: “1”. Black lines denote the invariant classifier satisfying the ICON objective. (a) Assumption 1 is violated. (b) Assumption 2 is violated.

good features [263], *i.e.*, they enable zero-shot generalization as $\mathbf{c}$ is class-unique, and they are robust to the domain shift by discarding the environmental features $\mathbf{e}$ from $\Phi(\mathbf{c}, \mathbf{e})$. To achieve generalization, we need the following assumptions in ICON.

Assumption (Identifiability of $\mathcal{G}/\mathcal{H}$).

- 1) $\forall \mathbf{x}_i, \mathbf{x}_j \in T, y_i = y_j \text{ iff } \mathbf{x}_i \in \{h \circ \mathbf{x}_j \mid h \in \mathcal{H}\}$;
- 2) There exists no linear map $l: \mathcal{X} \rightarrow \mathbb{R}$ such that $l(\mathbf{x}) > l(h \circ \mathbf{x}), \forall \mathbf{x} \in S, h \circ \mathbf{x} \in T$ and $l(g \circ \mathbf{x}) < l(gh \circ \mathbf{x}), \forall g \circ \mathbf{x} \in S, gh \circ \mathbf{x} \in T$, where $h \neq e \in \mathcal{H}, g \neq e \in \mathcal{G}/\mathcal{H}$.

Assumption 1 states that classes in T are separated by clusters. Figure 4.4a shows a failure case of Assumption 1, where the intra-class distance (*i.e.*, dotted double-arrow) is much larger than the inter-class one (*i.e.*, solid double-arrow) in T , *i.e.*, shape $\mathbf{c}$ is less discriminative than color $\mathbf{e}_T$. Hence the clusters in T are based on $\mathbf{e}$, causing ICON to capture the wrong invariance (Figure 4.2b). To prevent the failure case, recall that we use dimensionality reduction in Step 1 (Section 4.2.3) to highlight $\mathbf{c}$ and suppress $\mathbf{e}_T$. We also specify the number of clusters as the class numbers in T , which is a necessary condition to satisfy this assumption.

Assumption 2 states that $\mathbf{c}$ is the only invariance between S and T . Figure 4.4b illustrates a failure case, where there exist two invariant classifiers: both the vertical one based on shape $\mathbf{c}$ and the horizontal one based on color $\mathbf{e}$ separate the classes in S and clusters in T . Yet as T is unlabelled, there is no way for the model to determine which one is based on $\mathbf{c}$. In practice, this assumption can be met by collecting more diverse unlabelled samples in T (*e.g.*, darker “0”s and lighter “1”s in Figure 4.4b), or introducing additional priors on what features should correspond to the causal one.

Dataset	OFFICE-HOME	ViSDA-2017	WILDCAM	CAMELYON17	FMoW	POVERTYMAP	GLOBALWHEAT	OGB-MOLPCBA	CIVILCOMMENTS	AMAZON
Sample x	object image	object image	camera trap photo	tissue slide	satellite image	satellite image	wheat image	molecular graph	online comment	product review
Label y	65 categories	12 categories	182 species	tumor/not	62 land uses	asset wealth	wheat bbox	bioassays	toxic/not	5 review scores
Task	classification	classification	classification	classification	classification	regression	detection	classification	classification	classification
Source S	various*	synthetic images	photos from 243 traps	slides from hospital 1-3	images from 2002-2013	images in 14 countries	images in Europe	44,930 scaffold groups	online articles*	1,252 reviewers
Example S										
#Samples S	average 3,875	152,397	129,809	302,436	76,863	~10,000	2,943	350,343	269,038	245,502
Target T	various*	real photos	photos from 3215 traps	slides from hospital 5	images from 2016-2018	images in 5 countries	images across the world	43,793 scaffold groups	online articles*	1,334 reviewers
Example T										
#Samples T	average 3,875	55,388	819,120	600,030	173,208	261,396	42,445	517,048	1,551,515	268,761
Evaluation	average acc.	mean-class accuracy	macro-F1	acc.	worst-region acc.*	Pearson correlation* %	acc.	average precision	worst-group acc.*	10 th percentile acc.
Existing Methods	GVB [7]		Empirical Risk Minimization (ERM)							
	70.4	75.3	47.0 / 32.2	90.6 / 82.0	60.6 / 34.8	65 / 48	77.8 / 51.0	- / 28.3	89.8 / 66.6	72.0 / 54.2
	TCM [61]		CORAL [51]							
	70.7	75.8	40.5 / 27.9	90.4 / 77.9	58.9 / 34.1	54 / 36	- / -	- / 26.6	- / -	71.7 / 53.3
	SENTRY [41]		DANN [13]							
	72.0	76.7	48.5 / 31.9	86.9 / 68.4	57.9 / 34.6	50 / 33	- / -	- / 20.4	- / -	71.7 / 53.3
	CST [29]		Pseudo-Label [24]							
	72.2	80.6	47.3 / 30.3	91.3 / 67.7	60.9 / 33.7	- / -	73.3 / 42.9	- / 19.7	90.3 / 66.9	71.6 / 52.3
	ToAlign [59]		Noisy Student [60]							
	72.7	77.8	47.5 / 32.1	93.2 / 86.7	61.3 / 37.8	61 / 42	78.1 / 46.8	- / 27.5	- / -	- / -
	FixBI [37]		FixMatch [50]							
	73.0	82.8	46.3 / 31.0	91.3 / 71.0	58.6 / 32.1	54 / 30	- / -	- / -	89.4 / 65.7	71.9 / 53.9
	ATDOC [26]		SwAV [5]							
	73.2	83.7	47.3 / 29.0	92.3 / 91.4	61.8 / 36.3	60 / 45	- / -	- / -	89.9 / 69.4	73.6 / 56.4
ICON	75.8 +2.6	87.4 +3.7	50.6 / 34.5 +2.3	95.6 / 93.8 +2.4	62.2 / 39.9 +2.1	65 / 49 +1	78.6 / 52.3 +1.3	- / 28.3 +0.0	89.7 / 68.8 +1.9	71.9 / 54.7 +0.5

TABLE 4.1: Dataset details and the results of our proposed ICON compared with existing methods. Details on * in Section 4.2.6. WILDS 2.0 [24] datasets are highlighted in yellow, where models were evaluated on both the test data in S (first number) and in T (second number). As of thesis submission deadline, ICON dominates the WILDS 2.0 leaderboard (<https://wilds.stanford.edu/leaderboard/>), with the performance gain in T highlighted in red. “ERM (labeled T)” has full supervision in T . -/- means that the method is not applicable to the dataset.

Theorem. *When the above assumptions hold, ICON (optimizing Eq. (4.7)) learns a backbone θ mapping to a feature space $\mathcal{X}$ that generalizes under $\mathcal{G}/\mathcal{H} \times \mathcal{H}$. In particular, the learned f is the optimal classifier in S and T .*

By transforming each sample $\Phi(\mathbf{c}, \mathbf{e})$ in S and T to $\mathbf{c} \in \mathcal{X}$, the classifier easily associate each $\mathbf{c}$ to its corresponding class label using the labelled S to reach optimal.

4.2.6 Experiment

Datasets

As shown in Table 4.1, we extensively evaluated ICON on 10 datasets with standard protocols [24, 94], including 2 classic ones: OFFICE-HOME [7], ViSDA-2017 [3],

and 8 others from the recent WILDS 2.0 benchmark [24]. WILDS 2.0 offers large-scale UDA tasks in 3 modalities (image, text, graph) under real-world settings, *e.g.*, wildlife conservation, medical imaging and remote sensing. Notably, unlike the classic datasets where the same sample set in T is used in training (unlabelled) and testing, WILDS 2.0 provides unlabelled train set in T disjoint with the test set, which prevents explicit pattern mining of the test data. They also provide validation sets for model selection.

Details of * in Table 4.1: **Office-Home** has 4 domains: artistic images (Ar), clip art (Cl), product images (Pr), and real-world images (Rw), which forms 12 UDA tasks by permutation, and we only show Cl→Rw as an example in the table. **FMoW** considers the worst-case performance on different geographic regions, where the land use in Africa is significantly different. **PovertyMap** considers the lower of the Pearson correlations (%) on the urban or rural areas. **CivilComments** correspond to a semi-supervised setting with no domain gap between S and T , and is evaluated on the worst-case performance among demographic identities.

Implementation Details

Feature Backbone. We used the standard [12, 264] ResNet-50 [22] pre-trained on ImageNet [23] on the two classic datasets. For WILDS 2.0, we used the standard backbones specified in the benchmark.

Regression and Detection. For the regression task in POVERTYMAP, f is a regressor that outputs a real number. As the number of clusters in T is not well-defined, we directly used rank statistics to determine if two samples in T have the same label. For the BCE loss, we computed the similarity by replacing $f(\mathbf{x}_i)^\top f(\mathbf{x}_j)$ in Eq. (4.6) as $-(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2$. For detection in GLOBALWHEAT, we used the ICON objective in Eq. (4.7) to train the classification head without other modifications.

Other Highlights. We aim to provide a simple yet effective UDA method that works across data modalities and tasks, hence we did not use tricks tailored for a specific modality (*e.g.*, image) or even datasets, such as mixup [265], extensive augmentations (16 are used in [110]) or class-wise sampling [101]. We incorporated entropy minimization as a standard auxiliary objective [10, 12, 94] on classic UDA datasets. Thanks to our simple implementation, *all* our experiments can run on a single NVIDIA 2080Ti GPU.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
GVB [10]	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
TCM [266]	58.6	74.4	79.6	64.5	74.0	75.1	64.6	56.2	80.9	74.6	60.7	84.7	70.7
SENTRY [264]	61.8	77.4	80.1	66.3	71.6	74.7	66.8	63.0	80.9	74.0	66.3	84.1	72.2
CST [12]	59.0	79.6	83.4	68.4	77.1	76.7	68.9	56.4	83.0	75.3	62.2	85.1	73.0
ToAlign [267]	57.9	76.9	80.8	66.7	75.6	77.0	67.8	57.0	82.5	75.1	60.0	84.9	72.0
FixBi [268]	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
ATDOC [117]	60.2	77.8	82.2	68.5	78.6	77.9	68.4	58.4	83.1	74.8	61.5	87.2	73.2
ICON (Ours)	63.3	81.3	84.5	70.3	82.1	81.0	70.3	61.8	83.7	75.6	68.6	87.3	75.8

TABLE 4.2: Accuracy (%) on the OFFICE-HOME dataset [7] with 12 UDA tasks.

Main Results

Classic UDA Benchmarks. In Table 4.1 (first 2 columns), ICON significantly outperforms the existing state-of-the-arts on OFFICE-HOME [7] and VISDA-2017 [3] by 2.6% and 3.7%, respectively. In particular, Table 4.2 breaks down the performances in OFFICE-HOME, where ICON wins in 10 out of 12 tasks, and significantly improves the hard tasks (*e.g.*, +2.6% on Cl→Pr, +2.9% on Cl→Rw). On VISDA-2017, ICON beats the original competition winner (MT+16augs) [110] that averages predictions on 16 curated augmented views. ICON even outperforms the recently proposed MCC+NWD [259], which uses a much deeper ResNet-101 [22]. Notably, CST [12] and ATDOC [117] also aim to remove the spurious correlation, and both attain competitive performance, validating the importance of this research objective. Yet their training scheme allows the correlation to persist, hence ICON is more effective.

WILDS 2.0. In Table 4.1 (last 8 columns), ICON dominates the WILDS 2.0 leaderboard (link in caption) despite the challenging real-world data and stringent evaluation. For example, the animal images in iWILDCAM exhibit long-tail distribution that is domain-specific, because the rarity of species varies across locations. Hence under macro-F1 evaluation (sensitive to tail performance), the 2.3% improvement is noteworthy. On CIVILCOMMENTS with text data, ICON improves the worst performance across demographic groups by 2.2%, which promotes AI fairness and demonstrates the strength of ICON even under the SSL setting. On AMAZON, although ICON only improves 0.5% to reach 54.7%, the upper bound performance by training with labelled T (56.4%) is also low due to the strict evaluation metric. We also highlight that ICON increases the performance in T without sacrificing that in S (the first number). This empirically validates that the invariance objective of ICON resolves the tug-of-war scenario in Figure 4.2c.

Failure Cases. ICON performance is relatively weak on POVERTYMAP and OGB-MOLPCBA with two possible reasons: 1) no pre-training available, the

OFFICE-HOME					VISDA-2017				
$\beta \backslash \alpha$	0.25	0.5	1.0	2.0	$\beta \backslash \alpha$	0.25	0.5	1.0	2.0
0.05	73.8	75.0	75.5	73.9	0.05	85.2	86.2	86.3	81.5
0.1	73.9	75.4	75.8	74.4	0.1	85.6	86.4	86.7	82.6
0.25	70.2	72.8	75.2	74.5	0.25	76.8	81.2	87.4	86.3
1.0	73.9	75.4	75.8	74.4	1.0	74.6	77.6	70.0	72.2

FIGURE 4.5: Ablations of α, β on the classic UDA datasets.

Method	Office-Home	VisDA-2017
ICON	75.8	87.4
ICON-INV	74.2	84.2
ICON-INV-UMAP	74.1	82.0
Cluster with $2 \times \# \text{classes}$	69.7	78.6
Cluster with $0.5 \times \# \text{classes}$	67.5	76.2

TABLE 4.3: Ablations on each ICON component.

backbone is trained only on S and hence biased in the beginning, and 2) the ground-truth number of classes in T is not well-defined in both datasets, hence Assumption 1 in Section 4.2.5 may fail.

Ablations

Hyperparameters. We used two hyperparameters in our experiments: 1) the weight of the self-training loss denoted as α ; 2) the invariance penalty strength β in the REx [252] implementation of Eq. (4.7). Their ablations are in Figure 4.5 on the widely used classic UDA datasets. We observe that $\alpha \in [0.5, 1.0]$ and $\beta \in [0.1, 0.25]$ works well on both datasets. In particular, when α is increased, β should be increased accordingly to further regularize the supervision in S promoted by self-training. Hence overall, the hyperparameter search is easy.

Components. Recall that the main components of ICON include: 1) UMAP to reduce the feature dimensions in T before clustering; 2) clustering in T with $\# \text{clusters} = \# \text{classes}$; 3) the use of invariance in Eq. (4.7) (denoted as INV). Their ablations are included in Table 4.3. From the first three lines, we observe that removing each component leads to consistent performance drop on both datasets, which validates their effectiveness. However, by comparing the second and third line on OFFICE-HOME [7], we notice that the performance drop is small. We postulate that the backbone pre-trained on the vast ImageNet [23] effectively expresses the causal feature $\mathbf{c}$ of the common office/home objects, hence UMAP has limited

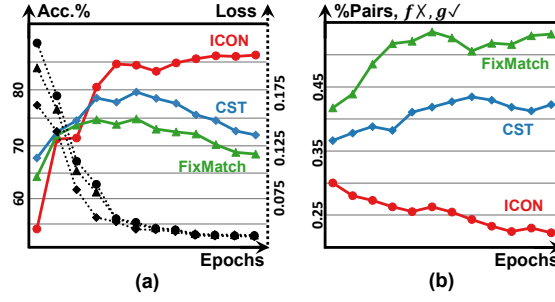


FIGURE 4.6: (a) Solid lines: test accuracy against training epochs for **FixMatch**, **CST** and **ICON**. Dotted black lines: average loss in each epoch with matching markers to denote the three methods. (b) Among the feature pairs where f fails, the percentage of pairs where g succeeds for the three methods in each epoch.

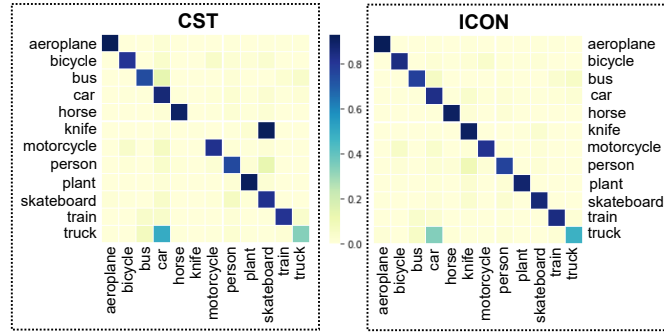


FIGURE 4.7: CST and ICON confusion matrix on VisDA-2017.

effect in further highlighting **c**. From the last two lines, when $\#clusters \neq \#classes$, the model performs poorly, *e.g.*, using one cluster in Figure 4.2b (*i.e.*, $0.5 \times \#classes$), the invariant classifier needs to predict the two red circles in S as dissimilar (two classes) and the two blue ones in T as similar (one cluster), which in fact corresponds to the erroneous red line (*i.e.*, predictions in T are similarly ambiguous). Overall, the empirical findings validate the importance of the three components, which are consistent with the theoretical analysis in Section 4.2.5.

Learning Curve. In Figure 4.6a, we plot the test accuracy *w.r.t.* training epochs on VisDA-2017 in colored solid lines and compare our ICON with two self-training baselines: FixMatch [107] and CST [12]. We also draw the training losses with the black dotted lines. We observe that the training losses for all methods converge quickly. However, while the two baselines achieve higher accuracy in the beginning, their performances soon start to drop. In contrast, ICON’s accuracy grows almost steadily.

Effect of Supervision in T . To find the cause of the performance drop, we used

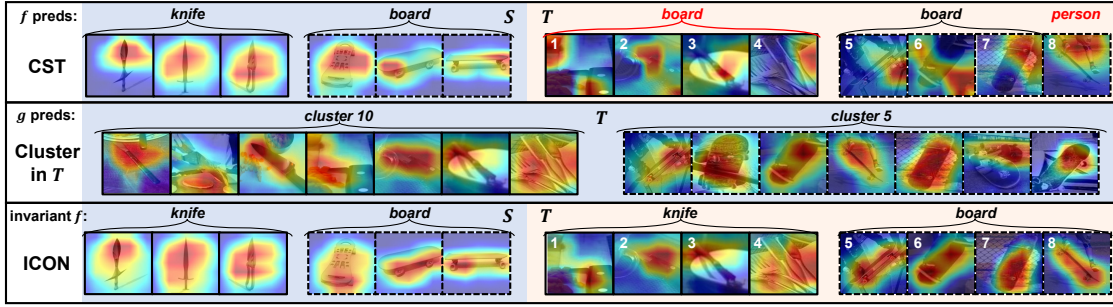


FIGURE 4.8: GradCAM on VisDA-2017 [3] using CST [12], the cluster head g and the proposed ICON. Ground-truth knives and boards are in solid and dotted boxes, respectively. Predictions from f, g are on top of each row. Red denotes wrong predictions. Pale blue and red background denotes training and testing, respectively. Note that g is trained without supervision in T .

the optimized g in Section 4.2.3 to probe the learned f in each epoch. Specifically, we sampled feature pairs $\mathbf{x}_1, \mathbf{x}_2$ in T and tested if the predictions of f, g are consistent with their labels, *e.g.*, f succeeds if $\text{argmax} f(\mathbf{x}_1) = \text{argmax} f(\mathbf{x}_2)$ when $y_1 = y_2$ (vice versa for $\neq$). Then among the pairs where f failed, we computed the percentage of them where g succeeded, which is shown in Figure 4.6b. A large percentage implies that f disrespects the inherent distribution in T correctly identified by g . We observe that the percentage has a rising trend for the two baselines, *i.e.*, they become more biased to the spurious correlations in S that do not hold in T . This also explains their dropping performance in Figure 4.6a. In particular, CST has a lower percentage compared to FixMatch, which suggests that it has some effects in removing the bias. In contrast, the percentage steadily drops for ICON, as we train f to respect the inherent distribution identified by g .

Confusion Matrix. To concretely show the bias, we plot the confusion matrix in Figure 4.7 for CST and ICON. The matrix on the left reveals that CST predicted most “knife”s as “skateboard”s (“board” for short), and some “truck”s as “car”s. On the right, we see that ICON predicts “knife” accurately and also improves the accuracy on “truck”.

Grad-CAM Visualization. Why does CST predict “knife” in T as “board”? To answer this, in Figure 4.8, we visualized the GradCAM [4] of CST and ICON trained on VisDA-2017. In the first row, we notice that the self-training based CST predicts the two categories perfectly by leveraging their apparent differences: the curved blade versus the flat board ending with semi-circles. However, this leads to poor generalization on “knife” in T , where the background (image 1, 4), the knife

component (2) or even the blade (3) can have board-like appearances. Moreover, biased to the standard “board” look in S , CST can leverage environmental feature for predicting “board” in T , *e.g.*, under side-view (5) or slight obstruction (6), which is prone to errors (*e.g.*, 8). As the board-like shape alone is non-discriminative for the two categories in T , in the second row, the cluster head g additionally leverages knife handles and wheels to distinguish them. In the third row, our ICON combines the supervision in S (first row) and T (second row) to learn an invariant classifier f . In S , ICON focuses on the overall knife shape instead of just blade, and incorporates wheels to predict “board”. In T , by comparing images with the same number in the first and third row, we see that ICON focuses more on the discriminative parts of the objects (*e.g.*, handles and wheels), hence generalizes better in T .

4.2.7 Supplementary Information

We include the proof of our theorem below.

Proof sketch. We will start by proving the binary classification case: we first show that the classifier satisfying the invariance objective in Eq. (2) (*i.e.*, line 2) separates the two classes among all samples in S and T . Then by minimizing line 1, samples are mapped to two class-specific features (*i.e.*, one feature for each class). Finally, by minimizing CE using labelled samples in S , we learn the optimal classifier that assigns the correct labels to all samples in S and T . We expand the binary classification case to C -class classification by formulating it as C 1-versus- $(C - 1)$ binary classification problems.

Proof. Consider a binary classification problem, where f outputs a single probability of belong to the first class, $\mathcal{G}/\mathcal{H} = \{g, e\}$, *i.e.*, $\mathbf{x}, g \circ \mathbf{x}$ are from different classes for all $\mathbf{x} \in \mathcal{X}$. Let f satisfies the invariance objective in Eq. (2), *i.e.*, $f \in \operatorname{argmin}_{\bar{f}} \text{BCE}(S, \bar{f}) \cap \operatorname{argmin}_{\bar{f}} \text{BCE}(T, \bar{f})$. As $\text{BCE}(S, f)$ is computed using the ground-truth labels, we must have $f(\mathbf{x}_1) \rightarrow 1, f(\mathbf{x}_2) \rightarrow 0$ or $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1$, $\forall \mathbf{x}_1 \in \mathcal{H}(\mathbf{x}) \cap S, \mathbf{x}_2 \in \mathcal{H}(g \circ \mathbf{x}) \cap S$, where $\mathbf{x} \in \mathcal{X}$. To see this, we shall expand $\text{BCE}(S, \bar{f})$ on a pair of samples $\mathbf{x}, \mathbf{x}' \in S$ with label y, y' :

$$-\mathbb{1}(y = y')\log(\hat{\mathbf{y}}^\top \hat{\mathbf{y}}') - \mathbb{1}(y \neq y')\log(1 - \hat{\mathbf{y}}^\top \hat{\mathbf{y}}'), \quad (4.8)$$

where $\mathbb{1}$ is the indicator function, and $\hat{\mathbf{y}}$ is given by $(f(\mathbf{x}), 1 - f(\mathbf{x}))$ (similarly for $\hat{\mathbf{y}}'$). Note that it is not difficult to show that $\hat{\mathbf{y}}^\top \hat{\mathbf{y}}' \in [0, 1]$, where the lower (or upper) bound is attained when they take different (or same) values from $(0, 1), (1, 0)$. As f minimizes $\text{BCE}(S, \bar{f})$, it minimizes the loss for each $\mathbf{x}, \mathbf{x}' \in S$: if $y = y'$, $\hat{\mathbf{y}}^\top \hat{\mathbf{y}}' \rightarrow 1$, and if $y \neq y'$, $\hat{\mathbf{y}}^\top \hat{\mathbf{y}}' \rightarrow 0$. This is only possible when the aforementioned condition holds. In the target domain T , although there is no ground-truth label, we prove that the evaluation of $\mathbb{1}(y = y')$ (or $\neq$) using the cluster labels is always the same as using ground-truth class labels, for all $\mathbf{x}, \mathbf{x}' \in T$ given Assumption 1: 1) suppose that $\mathbf{x} \in \{h \circ \mathbf{x}' \mid h \in \mathcal{H}\}$, then by the sufficient condition of Assumption 1, $\mathbb{1}(y = y') = 1$, reflecting that they are from the same class; 2) suppose that $\mathbf{x} \notin \{h \circ \mathbf{x}' \mid h \in \mathcal{H}\}$, then by the contra-position of the necessary condition, $\mathbb{1}(y \neq y') = 1$, reflecting that they are from the different classes. Hence in the same way as S , we can show that $f(\mathbf{x}_1) \rightarrow 1, f(\mathbf{x}_2) \rightarrow 0$ or $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1$, $\forall \mathbf{x}_1 \in \mathcal{H}(\mathbf{x}) \cap T, \mathbf{x}_2 \in \mathcal{H}(g \circ \mathbf{x}) \cap T$. Now considering the samples in S and T . Denote $D = S \cup T$. There are two possibilities: **1)** $f(\mathbf{x}_1) \rightarrow 1, f(\mathbf{x}_2) \rightarrow 0$ or $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1$, $\forall \mathbf{x}_1 \in \mathcal{H}(\mathbf{x}) \cap D, \mathbf{x}_2 \in \mathcal{H}(g \circ \mathbf{x}) \cap D$; **2)** $f(\mathbf{x}_1) \rightarrow 1, f(\mathbf{x}_2) \rightarrow 0$ (or $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1$), $\forall \mathbf{x}_1 \in \mathcal{H}(\mathbf{x}) \cap S, \mathbf{x}_2 \in \mathcal{H}(g \circ \mathbf{x}) \cap S$ and $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1$ (or $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1$), $\forall \mathbf{x}_1 \in \mathcal{H}(\mathbf{x}) \cap T, \mathbf{x}_2 \in \mathcal{H}(g \circ \mathbf{x}) \cap T$. Case 2 corresponds to the horizontal black line in Figure 4b. However, it contradicts with Assumption 2. Hence the only possibility is case 1, where f linearly separates the two classes in D , *i.e.*, all samples in S and T . In particular, as Assumption 2 prevents the failure case where f separates the samples not based on the class (or causal feature $\mathbf{c}$), it intuitively means that the only invariance between S and T is $\mathbf{c}$. Overall, this shows that a linear map $f \in \arg\min_{\bar{f}} \text{BCE}(S, \bar{f}) \cap \arg\min_{\bar{f}} \text{BCE}(T, \bar{f})$ separates the two classes in D .

We will now analyze the general case of c -class classification. Specifically, $\mathbf{w} \in \mathbb{R}^{c \times d}$ corresponds to a linear map that transforms each sample feature $\mathbf{x} \in \mathbb{R}^d$ to the c -dimensional logits. The classifier f parameterized by $\mathbf{w}$ outputs the softmax-normalized logits as the probability of belong to each of the c class. Without loss of generality, we consider the case where $\|\mathbf{w}_i\| = \|\mathbf{w}_j\|$ for $i, j \in \{1, \dots, c\}$ and $\|\mathbf{x}\| = M$ for all $\mathbf{x} \in \mathcal{X}$, *e.g.*, through the common l2-regularization and feature normalization. From the previous analysis, we prove that a linear classifier satisfying the invariance objective separates the two classes in D . We can generalize the results to multi-class classification by considering it as C 1-versus- $(C-1)$ binary classification problem (*e.g.*, separating class 1 samples and the rest of samples). We

can further show that minimizing line 1 of Eq. (2) leads to a feature space $\mathcal{X}$ that generalizes under $\mathcal{G}/\mathcal{H} \times \mathcal{H}$. In particular, to minimize $\text{BCE}(S, f) + \text{BCE}(T, f)$ (expanded in Eq. (4.8)), the prediction p of f on the correct class is maximized (*i.e.*, $p \rightarrow 1$). Specifically, the probability of belonging to class $i \in \{1, \dots, c\}$ for a feature $\mathbf{x}$ using softmax normalization is given by:

$$\frac{\exp(\mathbf{w}_i^\top \mathbf{x})}{\sum_{j=1}^c \exp(\mathbf{w}_j^\top \mathbf{x})}. \quad (4.9)$$

We show that to maximize p , a sample feature in class i is given by $M\bar{\mathbf{w}}_i$, where $M \rightarrow \infty$, $\bar{\mathbf{w}}_i$ is from normalizing $\mathbf{w}_i$, and for each $i \neq j \in \{1, \dots, c\}$, $\mathbf{w}_i^\top \mathbf{w}_j \leq 0$: the first condition is easily derived by maximizing $\exp(\mathbf{w}_i^\top \mathbf{x}) \rightarrow \infty$ under $\|\mathbf{x}\| = M$; the second condition is derived by limiting $p \rightarrow 1$, *i.e.*, $\exp(\mathbf{w}_j^\top \mathbf{x}) = 0$ or $\exp(\mathbf{w}_j^\top \mathbf{x}) \rightarrow -\infty$. Overall, this shows that minimizing Eq. (2) leads to $\mathcal{X}$ where the samples in class i are mapped to the same feature $M\bar{\mathbf{w}}_i$, and samples in different classes are mapped to different features as $\mathbf{w}_i \neq \mathbf{w}_j$ for $i \neq j$ (from $\mathbf{w}_i^\top \mathbf{w}_j \leq 0$). The classifier f is optimal by predicting the probability of each feature in S and T belonging to its ground-truth class as 1. Note that to further equip $\mathcal{X}$ with the equivariant property as introduced in Section 1, one can leverage a specially designed backbone θ (*e.g.*, equivariant neural networks [273]). Hence we complete the proof of our Theorem.

4.3 Case Study: Semi-Supervised Learning

4.3.1 Introduction

Semantic segmentation is a dense image classification task that predicts pixel-level semantic labels on the image [274, 275]. It widely applies to crucial areas, including autonomous driving [276, 277], medical imaging [278, 279] or remote sensing [280, 281]. However, as pixel-level annotations are extremely expensive, the labeled datasets are often limited in size and scope [128, 282], making fully-supervised learning impractical. Therefore, we focus on the practical semi-supervised semantic segmentation, where the training data consists of a labeled set $\mathcal{D}$ and a large unlabeled set $\mathcal{U}$, *e.g.*, for an autopilot system, $\mathcal{U}$ (*i.e.*, street images) can be collected in bulk by a camera-equipped vehicle cruise.

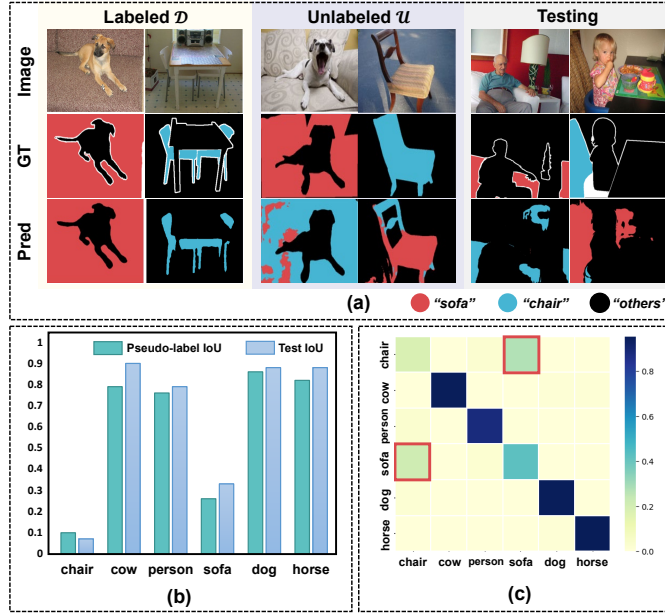


FIGURE 4.9: Results of UniMatch [13] (Section 4.3.5) trained on 92 images from PASCAL VOC 2012 [14]. (a) The ground-truth (GT) and predicted (Pred) semantic labels on “sofa” and “chair” images in $\mathcal{D}$, $\mathcal{U}$ and testing. (b) The class-wise IoU of pseudo-labels and in testing. (c) The confusion matrix in testing. We show a subset of the classes to avoid clutter. Full results in Section 4.3.6.

The mainstream method is pseudo-labeling [123, 125]: a model trained on $\mathcal{D}$ generates pseudo-labels on $\mathcal{U}$, which are then treated as the ground truth of $\mathcal{U}$ to update the model. Yet, training the model with noisy pseudo-labels leads to a well-known problem called confirmation bias [121, 137]. To alleviate it, existing works mainly employ two heuristics: 1) training using a subset of $\mathcal{U}$ with confident pseudo-labels [124, 125]; and 2) learning a robust model that outputs similar predictions on different views of the image in $\mathcal{U}$ [122, 126, 127]. For example, a top-performing method FixMatch [121] combines the two heuristics, *i.e.*, pseudo-labeling weakly-augmented $\mathcal{U}$ and updating the model using strongly-augmented $\mathcal{U}$ with confident pseudo-labels.

However, even with these heuristics, the model is still easily biased towards the context shortcut in scarce-labeled data $\mathcal{D}$. For example, in Figure 4.9, given limited labeled “chair” and “sofa” samples, color and texture feature becomes a valid contextual cue to distinguish them, *i.e.*, $\mathcal{D}$ has a context bias: “chair” $\approx$ (white, smooth) and “sofa” $\approx$ (brown, interleaved). The model is misled by the context bias and generates incorrect pseudo-labels on $\mathcal{U}$, *e.g.*, mis-classifying white “sofa” or interleaved “chair”, which reinforces the bias when updating the model with them. As a result, the model makes similar errors in testing, as shown in Figure 4.9 by (a)

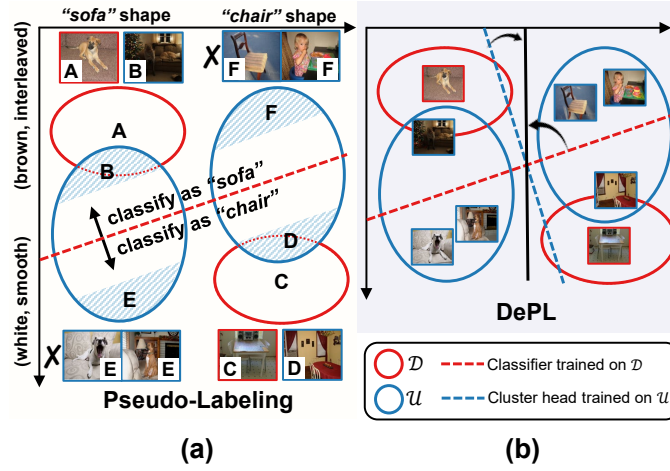


FIGURE 4.10: Circle denotes the pixel-level features in $\mathcal{D}$ and $\mathcal{U}$ in a 2D feature space. (a) An illustrated failure case of pseudo-labeling, where context bias in $\mathcal{D}$ leads to wrong pseudo-labels on $\mathcal{U}$. (b) Our DePL learns a debiased model (black line) to prevent such failure. Best viewed in color.

wrong predictions, (b) low accuracy on “chair” and “sofa”, and (c) the large error rate in the confusion matrix (highlighted by red boxes).

The cause of the persisting confirmation bias is two-fold:

1) The labeled $\mathcal{D}$ exhibits a different distribution from $\mathcal{U}$ due to labeling preference [283]. This is especially prominent in semi-supervised semantic segmentation, given the difficulty of annotating dense labels. Specifically, annotators tend to label more obvious objects—*e.g.*, located near the center or with a typical look—which naturally distributes differently from the less obvious ones. We illustrate the chair and sofa example in Figure 4.10a. The labeled $\mathcal{D}$ corresponds to the two red clusters A and C, which contain the features of obvious “sofa”—(brown, interleaved), and “chair”—(white, smooth), respectively. They differ both in context on the vertical axis (*e.g.*, color and texture) and class feature on the horizontal axis (*e.g.*, shape)². The unlabeled set $\mathcal{U}$ corresponds to the two blue clusters. It contains both obvious objects (*e.g.*, B, D) and less obvious ones (*e.g.*, white “sofa” in E or interleaved “chair” in F).

2) The model trained on $\mathcal{D}$ is inevitably biased to its contextual cues, *e.g.*, the optimal classification boundary for $\mathcal{D}$ (red line) is diagonal, which leverages both class and context feature. Consequently, the biased model mis-classifies the features in $\mathcal{U}$ from less obvious objects. Note that the two aforementioned heuristics are

²The pixel-wise feature contains both local (*e.g.*, color) and global feature (*e.g.*, texture, shape) with the large receptive field of the backbone.

unable to circumvent the bias. Confident pseudo-labels (shaded in blue) can still be wrong (*i.e.*, E, F), which invalidates the subsequent training with different views of $\mathcal{U}$, *i.e.*, making the model robustly *mis*-classifies the confident but wrong subset.

To address this, we point out that the inherent distribution in $\mathcal{U}$ contains the valuable debiasing cue, which is ever-overlooked in current pseudo-labeling methods. For example, while $\mathcal{U}$ is unlabeled, we can still identify the two unsupervised clusters in Figure 4.10b (blue circles). We observe that when the classifier captures the context bias in $\mathcal{D}$ (*e.g.*, red line), it breaks up the clusters in $\mathcal{U}$. Vice versa, when it is biased to $\mathcal{U}$ (*e.g.*, blue line), it breaks up the classes in $\mathcal{D}$ (red circles). Hence by contra-position, if a classifier separates the classes in $\mathcal{D}$ and the clusters in $\mathcal{U}$, it must be an unbiased one, *e.g.*, the vertical black line is the *only* classifier that meets the requirement in Figure 4.10b, which is indeed an unbiased one (*i.e.*, predicting with shape).

To this end, we propose **Debiased Pseudo-labeling (DePL)** as a simple yet effective regularization to alleviate the bias in semi-supervised semantic segmentation. Stemming from the fundamental clustering assumption [118, 119] in semi-supervised learning, we first learn a cluster head to group the pixel-level features in $\mathcal{U}$ as C unsupervised clusters, where C is the number of semantic classes. Then we use the Binary Cross-Entropy (BCE) loss as a regularization term, such that the classifier prediction is similar on pixel pairs from the same class of $\mathcal{D}$ or cluster of $\mathcal{U}$, and dissimilar on pairs from different ones. Hence to satisfy our regularization, the classifier must simultaneously separate the classes and clusters to become an unbiased one. In Section 4.3.2, we start with a preliminary on pseudo-labeling. Then we present the details of DePL in Section 4.3.3. Finally in Section 4.3.5, we show that our DePL can be plugged into different pseudo-labeling baselines, which leads to consistent performance increase on PASCAL VOC 2012 [14] and Cityscapes [128] under a wide range of settings.

4.3.2 Problem Formulations

The goal is to train a segmentation model: given an image of size $H \times W$, the model outputs a prediction in $\mathbb{R}^{H \times W \times C}$, representing the probabilities of belonging to the C semantic classes on each of the $H \times W$ pixel locations. The training data has two parts: 1) a set of labeled images $\mathcal{D}$, where each pixel is annotated with

its semantic label $y \in \{1, \dots, C\}$, and 2) a large unlabeled set $\mathcal{U}$. The model is trained by minimizing a combination of supervised cross-entropy loss on $\mathcal{D}$ and unsupervised loss on $\mathcal{U}$ (by pseudo-labeling). Details are given below.

Model. The prevailing segmentation model is Deeplab V3 [284]. We denote its high-level structure as **backbone** $\rightarrow$ **decoder** $\rightarrow f$, where $\rightarrow$ is the data forward direction and f is a classifier. Specifically, the **backbone** is typically ResNet-101 [285] pre-trained on ImageNet [286], whose last fully-connected (fc) layer is removed. Hence its output is the feature map from the last ResNet block. The **decoder** leverages Atrous Convolution [287] to enlarge the field of view, which helps capture global information (*e.g.*, shape) that benefits prediction. It also upsamples the input feature map through a trainable network for subsequent pixel-level prediction. f is implemented by an fc layer (*i.e.*, linear classifier). Given the pixel-level features from the decoder, f is *independently* applied to the feature at each pixel location. Note that while f takes the feature of a single pixel as input, its prediction still accounts for global information (*e.g.*, overall “sofa” shape) captured in the feature.

Data. Notice that the semantic label, decoder output, and classifier prediction are all pixel-level. Hence for simplicity, we formulate the semantic segmentation task under the conventional classification setting, where each pixel-level feature becomes a sample. Specifically, we write the labeled set $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i$ is the decoder output feature for the i -th *pixel*, $y_i \in \{1, \dots, C\}$ is its semantic label, and N denotes the total number of pixels for *all* labeled images. Similarly, we denote the unlabeled set $\mathcal{U} = \{\mathbf{x}_i\}_{i=1}^M$. We drop the subscript i when the context is clear. Given a feature $\mathbf{x}$, we denote its softmax-normalized prediction as $\hat{\mathbf{y}} = f(\mathbf{x}) \in \mathbb{R}^C$. In particular, it is common [122, 126, 127] to leverage different views of $\mathbf{x} \in \mathcal{U}$ for consistency training (*i.e.*, similar prediction from f on different views of $\mathbf{x}$). We denote the alternative view of $\mathbf{x}$ with $\mathbf{x}^*$, *e.g.*, the feature of a strongly-augmented image in FixMatch [121].

Objective. In pseudo-labeling methods, the model parameters are trained by minimizing the loss $\mathcal{L}_{pl}$ given below:

$$\mathcal{L}_{pl} = \sum_{i=1}^N \text{CE}(f(\mathbf{x}_i), y_i) + \lambda \sum_{j=1}^M \mathcal{L}_u(\mathbf{x}_j), \quad (4.10)$$

where CE is the cross-entropy loss on $\mathcal{D}$, λ is a balancing weight, and $\mathcal{L}_u$ is the unsupervised loss on $\mathcal{U}$ computed by pseudo-labeling. Without loss of generality, we consider the following form of $\mathcal{L}_u$:

$$\mathcal{L}_u(\mathbf{x}) = \mathbb{1}(\max(\hat{\mathbf{y}}) \geq \tau) \text{CE}(f(\mathbf{x}^*), \text{argmax}(\hat{\mathbf{y}})), \quad (4.11)$$

where $\mathbb{1}(\cdot)$ is the indicator function that returns 1 if the condition $\cdot$ holds and 0 otherwise, $\hat{\mathbf{y}} = f(\mathbf{x})$ is the predicted probabilities of the C classes on the original view $\mathbf{x}$, τ is a confidence threshold, and $\text{argmax}(\hat{\mathbf{y}})$ is the pseudo-label. Note that $\mathcal{L}_u$ combines the selection of confident pseudo-labels (by the indicator function) and consistency training (by $\mathbf{x}^*$). However, pseudo-labeling methods rely on the confidence threshold to select correct pseudo-labels for subsequent consistent training, which is not reliable in the presence of context bias (*e.g.*, Figure 4.10a). We present our DePL below as a debiased regularization.

Our approach stems from the fundamental clustering assumption [118, 119] in semi-supervised learning, which states that classes must be separated by low-density regions. This means that while $\mathcal{U}$ is unlabeled, we can separate features from different classes by clustering. The identified clusters allow us to explicitly regularize the classifier such that it simultaneously separates the classes in $\mathcal{D}$ and clusters in $\mathcal{U}$ (*e.g.*, black line in Figure 4.10b). We discuss each part below.

4.3.3 Clustering Assumption in SSL

Choice of Feature. Figure 4.11 compares 3 intermediate feature maps on their suitability for clustering: (a) Output of `backbone`; (b) Output of `backbone`→`decoder`; (c) Output of `backbone`→`decoder`→`avgpool`, *i.e.*, average pooling (b). Following [288], we upsampled the feature map to the image size by interpolation, and computed the average Euclidean distance between the neighboring activations (8 neighbors for each feature map location). In the ideal case where class features are fully separated, we should observe large Euclidean distance (brighter in Figure 4.11) at locations near the class boundary (*e.g.*, between “dog” and “background”), instead of those within a class (*e.g.*, inside “dog”). We find that the decoder output (b-c) is more suitable than the backbone one (a) advocated in [288], and we adopt the best performing (c) in DePL (*i.e.*, after average pooling). Ablation on the kernel size for average pooling is in Section 4.3.5. With slight abuse of notation, we

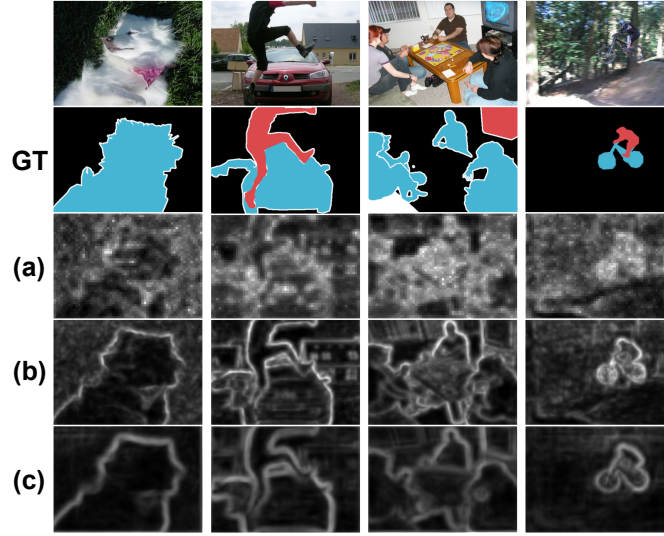


FIGURE 4.11: Clustering assumption on different intermediate outputs of the model. We used 4 images in $\mathcal{U}$ of PASCAL VOC 2012 dataset. GT denotes the ground-truth mask. $\bullet$: background; $\bullet$ and $\bullet$: the first and second non-background semantic classes in each image. (a) Local variations between each pixel and its 8 neighbors using the backbone feature. (b) Using decoder feature. (c) Using decoder feature after average pooling (7×7 kernel).

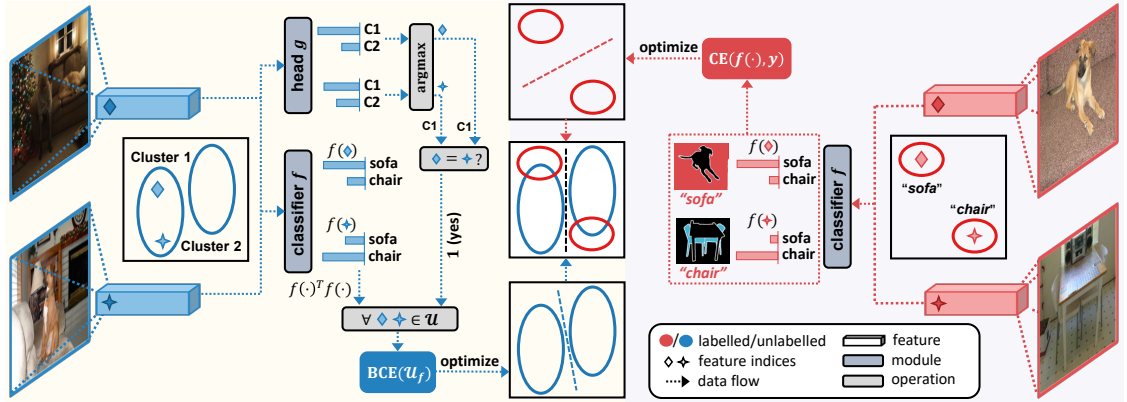


FIGURE 4.12: Pipeline of our proposed DePL. On the left, we show how $\mathcal{U}_f$ is constructed to compute $\text{BCE}(\mathcal{U}_f)$ using two features in $\mathcal{U}$ from the same Cluster 1 (C1) as an example. On the right, we show how CE is computed for a “sofa” feature and “chair” feature in $\mathcal{D}$. The optimal classification boundary for the CE loss and BCE loss are shown with the top and bottom diagram in the middle, respectively, which are combined to learn the unbiased one (black line in the center diagram). We omit 1) the learning of cluster head g as the process is similar with $\text{BCE}(\mathcal{U}_f)$; 2) pseudo-labeling loss as the process is similar with CE.

denote an average pooled feature in the unlabeled set as $\bar{\mathbf{x}} \in \mathcal{U}$, and one without pooling as $\mathbf{x} \in \mathcal{U}$.

Clustering Method. We learn a cluster head g that takes $\bar{\mathbf{x}} \in \mathcal{U}$ as input, and outputs the softmax-normalized probabilities for being in each of the C clusters. Note that C is always known in semi-supervised learning as classes are shared

between $\mathcal{D}$ and $\mathcal{U}$. We train g in a pair-wise manner such that it produces similar predictions on a pair of similar features and vice versa for a dissimilar pair, with two steps:

- 1) We first construct a pair-wise form of $\mathcal{U}$ based on the cluster prediction from g as:

$$\mathcal{U}_g = \{g(\bar{\mathbf{x}}_i)^\top g(\bar{\mathbf{x}}_j), \mathbb{1}(\bar{\mathbf{x}}_i \sim \bar{\mathbf{x}}_j) \mid (\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j) \in \mathcal{U}\}, \quad (4.12)$$

where the dot-product is used to measure the prediction similarity. We adopted two implementations to compute the feature similarity $\bar{\mathbf{x}}_i \sim \bar{\mathbf{x}}_j$: by checking i) if their cosine similarity is larger than a threshold (*e.g.*, 0.95); ii) if their top values have the same indices (*i.e.*, same rank statistics [289]). Ablation in Section 4.3.5 shows that they are both valid choices. Note that we freeze the features by detaching their gradients when constructing $\mathcal{U}_g$, as our goal here is to cluster $\mathcal{U}$ instead of learn features.

- 2) This allows us to train g by minimizing the Binary Cross Entropy loss given below.

$$\text{BCE}(\mathcal{U}_g) = - \sum_{(\hat{s}, s) \in \mathcal{U}_g} [s \log(\hat{s}) + (1 - s) \log(1 - \hat{s})], \quad (4.13)$$

where $\hat{s}$ denotes the prediction similarity $g(\bar{\mathbf{x}}_i)^\top g(\bar{\mathbf{x}}_j)$ for a pair $(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j) \in \mathcal{U}$, and $s = \mathbb{1}(\bar{\mathbf{x}}_i \sim \bar{\mathbf{x}}_j)$. Note that our cluster head g is trained online, which provides more up-to-date cluster assignment than the conventional offline methods, *e.g.*, k -means. After training, each $\bar{\mathbf{x}} \in \mathcal{U}$ is assigned a *cluster label* $y = \arg\max g(\bar{\mathbf{x}})$. Next, we regularize f to separate the clusters in $\mathcal{U}$ to form our DePL objective.

4.3.4 DePL Objective

We aim to regularize f to separate the clusters discovered in Section 4.3.3. However, we are unable to directly optimize f by minimizing the cross-entropy loss on each $\mathbf{x} \in \mathcal{U}$ with the cluster label y , because it does not necessarily align with the class index in $\mathcal{D}$. Hence we again adopt a pair-wise approach: training f to produce a similar prediction on feature pairs with the same cluster label, and push away those in different clusters. Correspondingly, we construct $\mathcal{U}_f$ based on the pair-wise prediction similarity of f :

$$\mathcal{U}_f = \{f(\mathbf{x}_i)^\top f(\mathbf{x}_j), \mathbb{1}(y_i = y_j) \mid \mathbf{x}_i, \mathbf{x}_j \in \mathcal{U}\}, \quad (4.14)$$

Algorithm 4 DePL Training (1 iteration)

-
- 1: **Input:** Labeled set $\mathcal{D}$, unlabeled set $\mathcal{U}$, backbone and decoder parameterized by θ , classifier f and cluster head g .
 - 2: **Output:** θ, f, g trained for 1 iteration.
 - 3: Extract features by θ : $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N, \mathcal{U} = \{\mathbf{x}_i\}_{i=1}^M$
 - 4: Extract average pooled feature by θ as $\{\bar{\mathbf{x}}_i\}_{i=1}^M$
 - 5: Compute pseudo-labeling loss $\mathcal{L}_{pl}$ by Eq. (4.10)
 - 6: Construct $\mathcal{U}_g$ with $\mathcal{U}$ and g via Eq. (4.12).
 - 7: Compute $\text{BCE}(\mathcal{U}_g)$ with Eq. (4.13)
 - 8: Construct $\mathcal{U}_f$ with $\mathcal{U}$ and f via Eq. (4.14).
 - 9: Compute $\text{BCE}(\mathcal{U}_f)$
 - 10: Optimize θ, f, g with Eq. (4.15)
-

where y_i, y_j are cluster labels. Note that we use $\mathbf{x} \in \mathcal{U}$ here (*i.e.*, without average pooling) to maximize feature fidelity when supervising f . Figure 4.12 left demonstrates the construction of $\mathcal{U}_f$ with two features in $\mathcal{U}$ from the same cluster. Our overall objective is given by:

$$\min_{\theta, f, g} \mathcal{L}_{pl} + \alpha \text{BCE}(\mathcal{U}_f) + \beta \text{BCE}(\mathcal{U}_g), \quad (4.15)$$

where θ denotes the backbone and decoder parameters, $\mathcal{L}_{pl}$ is the conventional training loss in Eq. (4.10), α is the weight of DePL regularization and β is the weight of clustering loss to train g . In practice, the loss weights are easy to choose, as shown by their ablations in Section 4.3.5. In Figure 4.12, we show how our regularization term $\text{BCE}(\mathcal{U}_f)$ works with the $\mathcal{L}_{pl}$ loss to learn an unbiased model (black line in the center diagram). We summarize DePL training in Algorithm 4.

4.3.5 Experiments

Settings

Datasets. We extensively evaluated our method on two semantic segmentation datasets, *i.e.*, Pascal VOC 2012 (VOC) [14] and Cityscapes [128]. Pascal VOC 2012 is an object-centric dataset with 20 semantic classes of objects and 1 background class. It has 1,464 training images and 1,449 images for validation. As per standard [25, 135, 139], we augmented the training set with 9,118 samples from the Semantic Boundaries Dataset (SBD) dataset [290], which have relatively lower

Methods	1/16 (92)	1/8(183)	1/4 (366)	1/2 (732)	Full (1464)	Crop
Sup-Only	50.70	59.10	65.00	70.60	74.10	513
CPS [291] [CVPR'21]	64.07	67.42	71.71	75.88	–	512
ReCo [292] [ICLR'22]	64.8	72.0	73.1	74.7	–	321
U ² PL [25] [CVPR'22]	67.98	69.15	73.66	76.16	79.49	513
ST++ [134] [CVPR'22]	65.20	71.00	74.60	77.30	79.10	321
PS-MT [142] [CVPR'22]	65.80	69.58	76.57	78.42	–	512
GTA-Seg [293] [NeurIPS'22]	70.02	73.16	75.57	78.37	80.47	513
FixMatch [121]	72.25	74.60	77.10	77.89	78.92	321
+ DePL (Ours)	74.91 +2.66%	77.19 +2.59%	78.06 +0.96%	78.84 +0.95%	80.05 +1.13%	321
UniMatch [13]	74.35	76.53	78.84	79.92	80.62	321
+ DePL (Ours)	75.43 +1.08%	77.63 +1.10 %	78.91 +0.07%	80.45 +0.53%	81.44 +0.82%	321

TABLE 4.4: Comparison with *state-of-the-art* methods on the Pascal VOC 2012 dataset in the *classic* setting. We reproduced UniMatch [13] with its official code on the same machine and code environment with where we conducted our DePL experiments. We indicate the improvements from our DePL with red numbers. The best result in each column is in **bold**.

annotation quality. Cityscapes [128] is an urban driving scene dataset with 19 classes. It contains 2,975 images for training and 500 for validation.

Evaluation Protocols. On VOC, there are two settings: 1) The *classic* setting takes only the standard 1,464 images from the original VOC train split as candidate labeled images, and all images from SBD are in the unlabeled set; 2) The *blended* setting takes all 10,582 images from VOC and SBD as the candidate ones. On Cityscapes, all images in the train split are candidate labeled images. In semi-supervised settings, a portion of the candidate labeled images (*e.g.*, 1/16) are sampled as the labeled set, and the remaining are used as the unlabeled set. On both datasets, We use the same random seed as in [291] to sample the labeled set for fair comparisons. In evaluation, we center cropped the validation images in Pascal VOC 2012, and adopted sliding window evaluation on Cityscapes following [25, 134]. The evaluation metric is the mean of Intersection over Union (mIoU) measured on the validation set.

Baselines. We have two pseudo-labeling baselines: 1) We implemented a strong FixMatch baseline, whose training objective follows Eq. (4.10). For the confidence threshold, we used $\tau = 0.95$ following common practice [121, 294] on all our experiments. For the image-level strong augmentations to generate the alternative view $\mathbf{x}^*$, we modified from the augmentation strategy in [134] by replacing Cutout [295] with CutMix [296], which is a standard augmentation technique in semantic segmentation. 2) We adopted another strong baseline UniMatch [13], which leverages

Methods	1/16	1/8	1/4	Crop
Sup-Only	69.55	73.63	76.67	513
CAC [139]	72.40	74.60	76.30	320
CPS [291]	74.48	76.44	77.68	512
AEL [135]	77.20	77.57	78.06	513
U ² PL [25]	74.43	77.60	78.70	513
ST++ [134]	74.50	76.30	76.60	321
PS-MT [142]	75.50	78.20	78.72	512
FixMatch [121]	74.15	75.18	76.42	513
+ DePL (Ours)	76.26	77.05	77.97	513
UniMatch [13]	78.10	78.40	79.20	513
+ DePL (Ours)	78.15	78.82	79.33	513

TABLE 4.5: Comparison with *state-of-the-art* methods on the Pascal VOC 2012 dataset in the *blended* setting.

two augmented views with image-level perturbation and feature-level perturbation. We used its best-performing dual-stream variant.

Implementation Details. We adopted the prevailing DeepLabv3+ [284] with ResNet-101 [285] as our segmentation model. We set the initial learning rate as 0.001 and 0.005 on VOC and Cityscapes with polynomial decay scheduling. We optimized the models with SGD [297] for 80 and 240 epochs on VOC and Cityscapes, respectively. We used standard image augmentations: random cropping, flipping, and resizing. We used a crop size of 321, 513, and 769 on *classic* VOC, *blended* VOC and Cityscapes, respectively. We followed [13] to set λ in Eq. (4.10) as 1 in all experiments.

Main Results

Results on VOC *classic*. In Table 4.4, we added our proposed DePL to the two strong baselines, and compared them with the state-of-the-art (SOTA) methods³. Overall, despite using a relatively small crop size, DePL improves the two baselines in all settings, and adding DePL to UniMatch [13] achieves new SOTA results. We have the following results break-down: 1) In general, the improvements are larger with smaller training images, *e.g.*, on the 1/16 (92) setting, we improve FixMatch and UniMatch by 2.66% and 1.08%, respectively, and on the Full (1464) setting, we improve them by 1.13% and 0.82%. This is because as the diversity of the labeled images increases, the spurious context bias specific to the labeled dataset

³All the results of U²PL in our paper are based on the revised performance in their official project page: <https://github.com/Haochen-Wang409/U2PL/issues/3>.

is reduced; hence debiasing is expected to bring fewer improvements. In fact, when the labeled data is sufficient enough such that evaluation $\approx$ in-distribution, semi-supervised learning is not expected to bring additional improvements [298]. Nevertheless, the effectiveness of DePL—especially in scenarios with limited labeled data—is a strong validation of its debiasing capability. 2) Our improvements on UniMatch are less than those on FixMatch, *e.g.*, +0.53% on UniMatch compared to +0.95% on FixMatch in the 1/2 (732) setting. Besides the obvious reason—it is harder to improve on the much stronger baseline results of UniMatch, we conjecture that random perturbations on the feature level introduced by UniMatch simulate a causal intervention [299, 300] to alleviate the context bias, *e.g.*, if we can destroy the spurious context feature (*e.g.*, texture in Figure 4.10) by perturbation, the model can only capture the unbiased class feature. Still, our DePL achieves further debiasing by improving UniMatch, *e.g.*, +1.08% in the 1/16 (92) setting.

Results on VOC *blended*. Overall, the improvements from DePL are consistent across the baselines and settings, with similar trends as observed in the *classic* setting. Note that our method has similar performance with UniMatch in the 1/16 setting. Besides the reason explained above, we postulate that on VOC *blended*, the supervision from $\mathcal{D}$ is not as robust as that in *classic* due to the coarse annotations on SBD dataset. Hence additionally regularizing the training by $\mathcal{U}$ may lead to poor optimization, especially when the labeled data is scarce (*e.g.*, in 1/16).

Results on Cityscapes. In contrast to the semantic classes in PASCAL VOC (*e.g.*, common objects, various vehicles, animals), Cityscapes focus on the urban scene. On Cityscapes, combining our DePL with UniMatch achieves new SOTA results, *e.g.*, in the 1/16 setting, we improve UniMatch by 0.67% and previous SOTA U²PL by 1.47%, and in the 1/8 setting, we improve them by 0.63% and 1.45%, respectively. This validates that DePL is effective on images from different domains.

Ablation Experiments

We used PASCAL VOC 2012 [15] as its training is more scalable compared to Cityscapes [128] training. In particular, we used UniMatch+DePL for ablation with 1/16 (92) and Full (1464) in the *classic* setting to get ablation results under two extreme cases.

Methods	1/16	1/8	1/4	Crop
Sup-Only	65.74	72.53	74.43	768
CPS [291]	69.78	74.31	74.58	800
AEL [135]	74.45	75.55	77.48	769
U ² PL [25]	74.90	76.48	78.51	769
PS-MT [142]	–	76.89	77.60	712
GTA-Seg [293]	69.38	72.02	76.08	–
UniMatch [13]	75.70	77.30	78.70	769
+ DePL (Ours)	76.37	77.93	78.80	769

TABLE 4.6: Comparison on the Cityscapes dataset. We list the AEL results reproduced by U²PL [25].

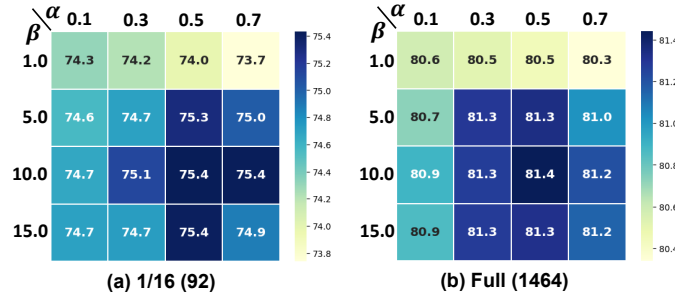


FIGURE 4.13: Ablation on α, β .

Loss Weight α, β . In Eq. (4.15), they correspond to the weight of our proposed regularization using the clusters of $\mathcal{U}$, and the weight of clustering loss, respectively. The search for their best combination is easy: We expect α to be less than the weight of the supervised cross-entropy loss, *i.e.* 1.0, as the cluster label of the unsupervised $\mathcal{U}$ is generally less accurate compared to ground-truth annotations. We also expect β to be larger than 1.0, such that the cluster head g can keep up with the feature learning to always return the up-to-date cluster prediction. This is consistent with the ablation results in Figure 4.13, *e.g.*, a small β leads to performance decreases, and for α , the performance saturates at around $\alpha = 0.5$. Overall, the results with a large $\beta > 1$ are stable, and the best performing $\alpha = 0.5, \beta = 10.0$ works well in both settings. Hence we choose this combination as default.

Kernel Size. We experimented with different kernel sizes for average pooling, which we use as a pre-processing before clustering in Section 4.3.3. The results are in Table 4.7. By comparing 1×1 with 3×3 , we see that the results are boosted in both settings by using average pooling, which is consistent with the qualitative results in Figure 4.12. Note that the performance drops from 7×7 to 9×9 . This is

Kernel Size	1/16 (92)	Full (1464)
1×1	74.61	80.82
3×3	75.17	81.22
5×5	75.21	81.35
7×7	75.43	81.44
9×9	75.37	81.30

TABLE 4.7: Ablation on the kernel size for average pooling. 1×1 corresponds to using no average pooling.

Similarity Measure	1/16 (92)	Full (1464)
cosine>0.95	75.25	81.28
cosine>0.9	75.30	81.36
cosine>0.85	75.04	80.94
top-5	75.21	81.18
top-3	75.43	81.44
top-2	74.89	81.06

TABLE 4.8: Ablation on the similarity measure. cosine> s denotes that $\mathbb{1}(\bar{\mathbf{x}}_i \sim \bar{\mathbf{x}}_j)$ in Eq. (4.12) returns 1 (*i.e.*, similar) if their cosine similarity is larger than s , and 0 otherwise. Correspondingly, top- k denotes that two features are similar if their top- k feature values have the same indices.

because a large kernel can dilute the large amount of features near class boundaries (*e.g.*, blurry edges in Figure 4.12), making it difficult to cluster them correctly.

Method	bg	pln	bike	bird	boat	botl	bus	car	cat	chair	cow	tabl	dog	hrs	mbk	prs	plant	shp	sofa	trn	tv	mIoU
Sup-only	89.9	73.6	33.8	75.1	42.0	54.4	80.0	75.8	78.9	24.7	50.2	43.1	72.6	50.2	68.2	77.2	34.9	64.8	30.6	67.6	55.1	59.2
CReST [301]	90.5	77.0	38.6	74.8	48.2	52.1	83.3	76.0	82.9	24.9	61.2	49.8	79.6	63.7	71.2	77.3	41.5	65.9	34.8	74.7	59.1	63.2
DARS [302]	91.3	82.6	37.4	81.9	50.5	58.6	88.5	82.9	82.8	25.5	56.3	49.1	75.3	64.6	73.6	79.7	42.2	64.0	37.1	73.4	57.9	64.5
USRN [303]	91.9	84.1	36.1	84.9	52.8	66.4	87.9	81.8	86.4	26.5	75.2	58.6	83.0	73.3	74.7	80.2	40.7	76.2	42.0	78.5	59.8	68.6
UniMatch [13]	93.6	79.4	43.4	89.9	69.4	80.6	94.2	86.3	92.8	31.4	87.9	62.1	89.8	86.8	87.3	84.2	56.9	87.9	42.2	85.2	69.3	76.2
+DePL	93.7	81.2	44.2	90.8	70.5	82.3	92.4	85.2	92.8	33.9	88.7	63.8	90.6	88.5	82.9	83.2	58.8	88.2	44.9	87.6	69.1	76.8

TABLE 4.9: Comparison of class-wise IoU on 1/32 VOC *blended*. We shorten the name of some semantic classes. bg: background, pln: aeroplane, botl: bottle, tabl: table, hrs: horse, mbk: motorbike, prs: person, shp: sheep, trn: train.

Similarity Measure. Its ablation results are in Table 4.8. Notice that the best performance achieved by cosine similarity and rank statistics are closed, *e.g.*, 75.30% and 75.43% in 1/16 (92), respectively. This suggests that both measures are valid choices. In particular, all results are better than the baseline UniMatch (74.35% and 80.62%, respectively as in Table 4.4), validating that our approach is not sensitive to the similarity measure. We adopted top-3 in our experiments.

Additional Results

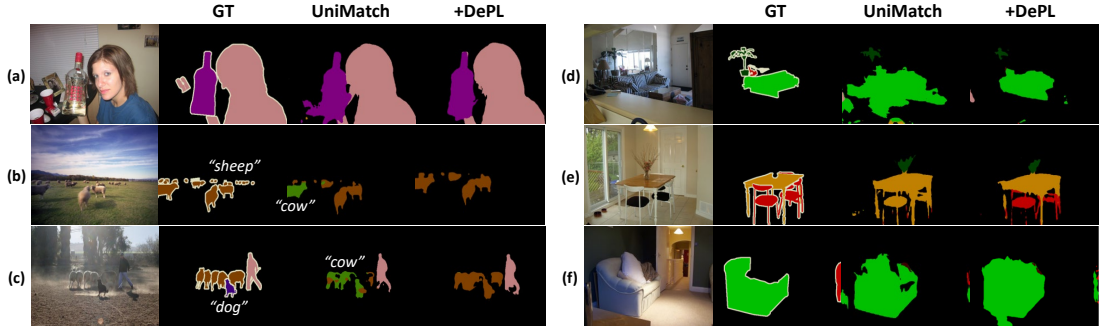


FIGURE 4.14: Predictions from UniMatch [13] and UniMatch+DePL (Ours). GT denotes the ground-truth mask. We used the models trained in 1/16 (92) on VOC *classic*. We highlight the label name when necessary.

Classwise-IoU. For a fair comparison with existing methods, we trained UniMatch [13] and our DePL on the 1/32 VOC *blended* setting. The results are in Table 4.9. Notice that we significantly outperform previous methods. Our DePL also has the highest IoU (bold) on most semantic classes. In particular, we significantly outperform UniMatch and other methods on hard classes, *e.g.*, +2.5% on “chair”, +2.7% on “sofa” and +1.7% on “table”, which corresponds to those exhibiting context bias (Figure 4.9). This strongly validates our debiasing capability.

Qualitative Examples. We show the predictions from UniMatch and our DePL in Figure 4.14. In (a), we observe that our DePL generates cleaner boundaries of the bottle, without mixing with its cluttered surroundings. In (b) and (c), UniMatch spuriously predicts dark “sheep” as “cow”, as “sheep” is mostly white in training. Our DePL is not misled by this bias. In (c), the “dog” is visually similar to “sheep”. Without prior knowledge of herding dogs, DePL predicts it as “sheep”. In (d), UniMatch predicts part of the brown surface as a “sofa”, while DePL generates a clean “sofa” prediction. In (e), the cushions of the chairs are black, resembling the table surface, and UniMatch generates wrong predictions on them. In contrast, DePL correctly classifies the indistinct chairs. In (f), DePL’s predicted mask on the “sofa” is arguably better than the ground-truth one. Note that both methods struggle to recognize the “chair” (labeled in red) near the edge, as most of it lies outside the image boundary.

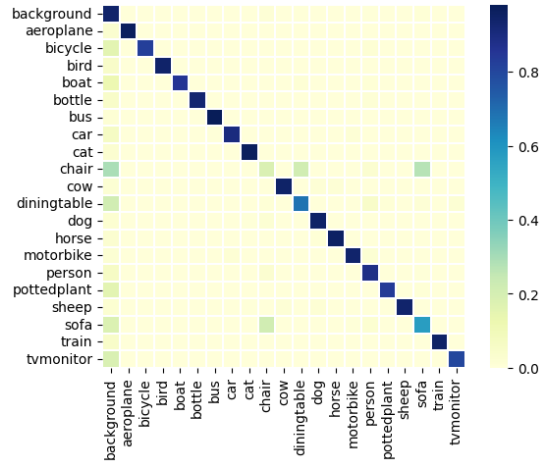


FIGURE 4.15: Confusion matrix for UniMatch [13] trained on 1/16 PASCAL VOC 2012 [15] *classic* setting. Supplementary to Figure 4.9

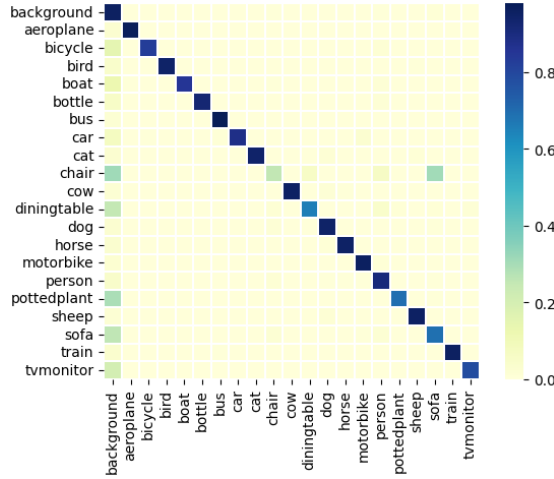


FIGURE 4.16: Confusion matrix for our DePL trained on 1/16 PASCAL VOC 2012 [15] *classic* setting. Supplementary to Figure 4.9

4.3.6 Supplementary Information

Full Class-wise Accuracy. In Figure 4.17, we show the full class-wise accuracy for UniMatch and our DePL, as well as the % improvements of DePL over UniMatch on each class. We first observe a strong correlation between the pseudo-labeling accuracy (*i.e.*, unlabeled) and test accuracy (*i.e.*, test) for both models. This validates that generating accurate pseudo-labels is the crux to improve semi-supervised semantic segmentation. From the last plot, we see that DePL improves the accuracy for most classes. In particular, the improvements are significant on a few under-performing classes. While the accuracy slightly drop on a few top-performing classes (*e.g.*, sheep), the errors made by DePL can be indeed plausible (Figure 4.14).

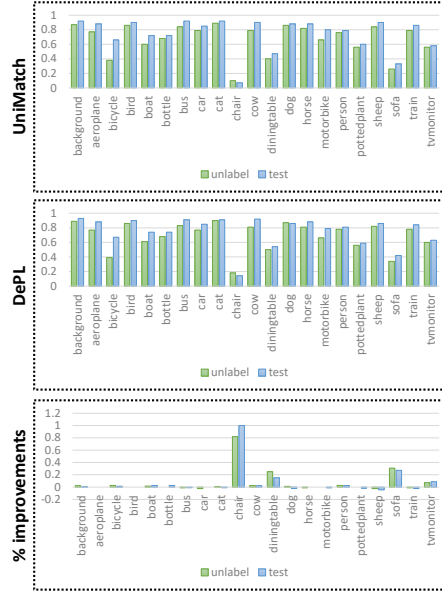


FIGURE 4.17: Class-wise IoU of UniMatch [13] and DePL on 1/16 PASCAL VOC 2012 [15] *classic* setting. Supplementary to Figure 1. The last plot shows the %improvements of DePL over UniMatch on each class.

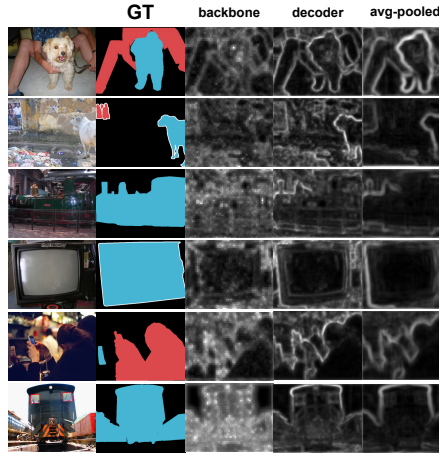


FIGURE 4.18: Clustering assumption on different intermediate outputs of the model. We used 6 images in $\mathcal{U}$ of PASCAL VOC 2012 dataset. Supplementary to Figure 4.12.

Clustering Assumption. In Figure 4.18, we show some additional examples to validate the clustering assumption on different intermediate model outputs. Overall, the average-pooled feature demonstrates the best clustering quality, *i.e.*, the bright areas correspond to the ground-truth class boundary.

Additional Qualitative Examples. In Figure 4.19, we show some additional examples on the predictions from UniMatch and Unimatch+DePL. For example. in the first image, DePL did not mistake the keyboard as the monitor, who share similar context feature “black”. In the third image, DePL managed to recognize

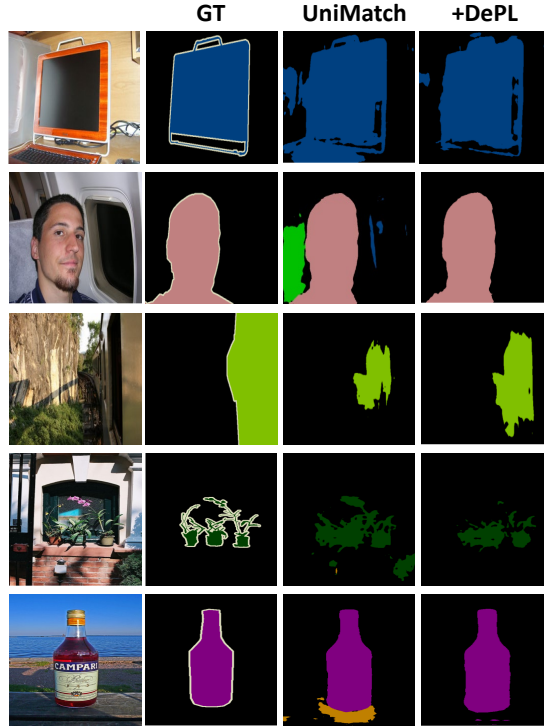


FIGURE 4.19: Predictions from UniMatch [13] and UniMatch+DePL (Ours). GT denotes the ground-truth mask. We used the models trained in 1/16 (92) on VOC *classic*. Supplementary to Figure 4.14

more areas of the train despite the heavy reflection on the train surface. In the fourth image, DePL did not recognize plants without pot as “potted plant”.

4.4 Conclusion

In Section 4.2, we presented a novel UDA method called Invariant CONsistency learning (ICON) that removes the spurious correlation specific to the source domain S to generalize in the target domain T . ICON pursues an invariant classifier that is simultaneously consistent with the labels in S and the clusters in T , hence removing the spurious correlation inconsistent in T . We show that ICON achieves generalization in T with a theoretical guarantee. Our theory is verified by extensive evaluations on both the classic UDA benchmarks and the challenging WILDS 2.0 benchmark, where ICON outperforms all the conventional methods. We will seek more principled pre-training paradigms that disentangle the causal features and relax the assumption of knowing the class numbers in T to improve regression tasks.

In Section 4.3, we presented a novel debiasing regularization for semi-supervised semantic segmentation called Debiased Pseudo-Labeling (DePL). Our approach stems from the fundamental clustering assumption in semi-supervised learning, enabling us to discover the inherent feature distribution in the unlabeled data by unsupervised clustering. Then, by regularizing the classifier to simultaneously separate the labeled classes and unlabeled clusters, the context bias that fails in the unlabeled data is removed. Our approach is strongly validated by the state-of-the-art performance and extensive ablations on standard benchmarks. As a future direction, we will pursue additional inductive bias beyond unsupervised clustering to discover the inherent distribution of unlabeled data, and seek other principled paradigms for debiasing.

In the next Chapter, we explore other inductive biases that can mitigate spurious correlations, including supervision about the class attributes, as well as prior information about granularity of attributes provided by diffusion time-steps.

Chapter 5

De-confound by Learning Disentangled Representation ¹

5.1 Preliminaries

A good feature can be formally defined by the classic notion of disentangled representation. We aim to provide a preliminary introduction below, and refer interesting readers to [38, 263] for more formal details. Each sample $\mathbf{x} \in \mathcal{X}$ in the image space $\mathcal{X}$ is generated from a hidden attribute vector $\mathbf{z} \in \mathcal{Z}$ in the vector space $\mathcal{Z}$ through an injective mapping $\Phi : \mathcal{Z} \rightarrow \mathcal{X}$ (*i.e.*, different samples are generated by different attributes). In particular, $\mathcal{Z}$ can be decomposed into the Cartesian product of N subspaces $\mathcal{Z} = \mathcal{Z}_1 \times \dots \times \mathcal{Z}_N$, where each $\mathcal{Z}_i$ is called a *modular* attribute, as its value can be locally intervened without affecting other modular attributes (*e.g.*, changing “expression” without affecting “gender” or “age”). A disentangled representation is an injective mapping $f : \mathcal{X} \rightarrow \mathcal{Z}$ that inverses Φ , *i.e.*, given each image $\mathbf{x}$, $f(\mathbf{x})$ recovers the modular attributes as a feature vector $\mathbf{z} = [\mathbf{z}_1, \dots, \mathbf{z}_N]$, which enables generalization in the following ways:

1) As shown in Figure 5.1 left, it is robust to the change of environmental feature. For example, if we disentangle foreground class and background, the trained

¹The work in this chapter has been published in the paper: **Zhongqi Yue***, Tan Wang*, Qianru Sun, Xian-Sheng Hua, Hanwang Zhang. Counterfactual Zero-Shot and Open-Set Visual Recognition. 2021. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* and **Zhongqi Yue**, Jiankun Wang, Qianru Sun, Lei Ji, Eric I-Chao Chang, Hanwang Zhang. Exploring Diffusion Time-steps for Unsupervised Representation Learning. 2023.

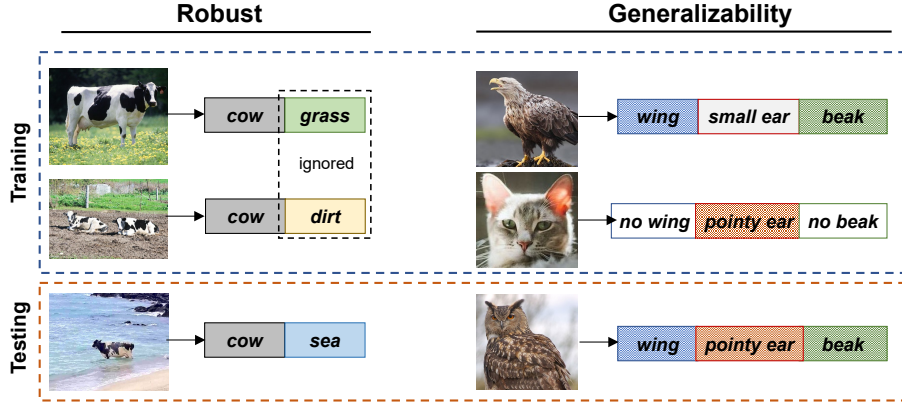


FIGURE 5.1: Disentangled representation enables generalization.

classifier can ignore the ever-changing background (*i.e.*, cows are on grass or dirt) and rely solely on the stable class feature (*i.e.*, cow feature), hence removing the spurious correlation between background and class (*i.e.*, most cows are on grass). In this way, even if we observe an atypical image in testing, *e.g.*, a cow in the sea, the classifier will not fail.

2) As shown in Figure 5.1 right, in the ZSL example, semantics changes are preserved in the feature space, even for unseen images such as an owl. Hence the metrics in the feature space trained on the seen-classes remain valid on the unseen-classes.

However, learning a disentangled representation requires explicit inductive bias about the hidden attributes [47]. In this chapter, we study two types of inductive biases: In Section 5.2, we present a case study of supervised disentangled representation learning. In Section 5.3, we extend the study to unsupervised learning by revealing the inductive bias provided by time-steps of a diffusion model.

5.2 Case Study: Supervised Learning

5.2.1 Introduction

Generalizing visual recognition to novel classes unseen in training is perhaps the Holy Grail of machine vision [304]. For example, if machines could classify new

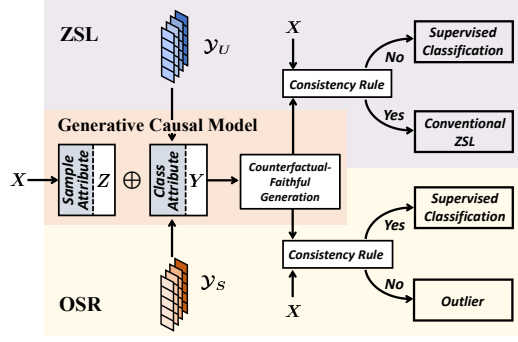


FIGURE 5.2: Our counterfactual framework for ZSL and OSR. Here $\oplus$ denotes vector concatenation, $\mathcal{Y}_U$ and $\mathcal{Y}_S$ denote the set of unseen and seen class attributes, respectively.

classes accurately by Zero-Shot Learning (ZSL¹) [29, 150], we could collect labelled data as many as possible for free; if machines could reject samples of unknown classes by Open-Set Recognition (OSR) [158, 306], any recognition system would be shielded against outliers. Unfortunately, this goal is far from achieved, as it is yet a great challenge for them to “imagine” the unseen world based on the seen one [307, 308].

Over the past decade, all the unseen-class recognition methods stem from the same grand assumption: *attributes (or features) learned from the training seen-classes are transferable to the testing unseen-classes*. Therefore, if we have the ground-truth *class attributes* describing both of the seen- and unseen-classes (or only those of the seen in OSR), ZSL (or OSR) can be accomplished by comparing the predicted class attributes of the test sample and the ground-truth ones [30, 151]; or by training a classifier on the samples generated from the ground-truth attributes [48, 309].

Not surprisingly, the above assumption is hardly valid in practice. As the model only sees the seen-classes in training, it will inevitably cater to the seen idiosyncrasies, and thus result in an unrealistic imagination of the unseen world. Figure 5.3a illustrates that the samples, generated from the class attribute of an unseen-class, do not lie in the sample domain between the ground-truth seen and unseen, *i.e.*, they resemble neither the seen nor the unseen. As a result, the seen/unseen boundary learned from the generated unseen and the true seen

¹Conventional ZSL [150] only evaluates the recognition on the unseen-classes. We refer to ZSL as a more challenging setting: Generalized ZSL [305], which is evaluated on the both seen- and unseen-classes.

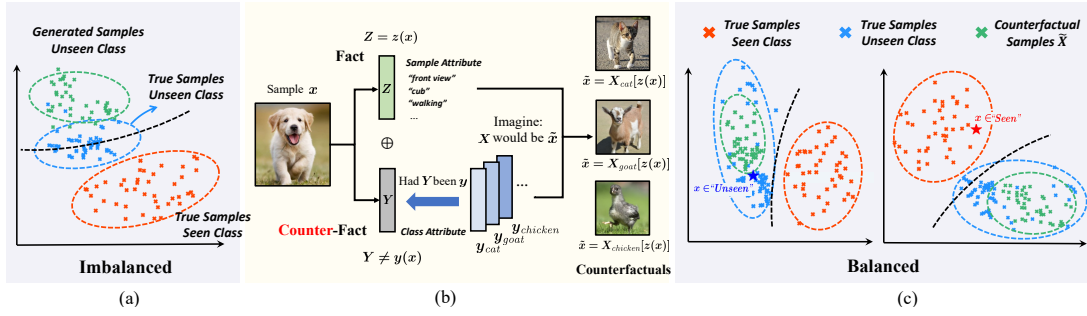


FIGURE 5.3: (a) t-SNE [6] plot of the CUB [16] samples in ZSL using a conventional unseen generation method [17], where a single pair of seen- and unseen-class is shown to avoid clutter. Due to the out-of-distribution generation (green), the decision boundary (black dashed lines) is imbalanced between the true seen (red) and unseen (blue) samples. (b) Illustration of our counterfactual generation. (c) t-SNE plot of the CUB using our counterfactual generation. The decision boundary is balanced. See Figure 5.5 for more examples of OSR.

samples is imbalanced. Such imbalanced classification increases the recall of the seen-class by sacrificing that of the unseen. Interestingly, we find that all the existing ZSL methods suffer from this imbalance: the seen accuracy is much higher than the unseen (*cf.* Table 5.1).

Astute readers may intuitively realize that it is all about *disentanglement*: if every attribute is disentangled, any unseen combination will be sensible. However, it is well-known that learning disentangled features is difficult, or even impossible without proper supervision [47]. In this paper, we propose another way around: **Counterfactual Inference** [115], which does not require the disentanglement for the class attributes. As illustrated in Figure 5.3b, denote X as the sample variable, which can be encoded into the *sample* attributes $Z = z(X = x)$ (*e.g.*, “front view”) and *class* attributes $Y = y(X = x)$. The counterfactual [40]: $\tilde{x} = X_y[z(x)]$ of sample x , is read as: X would be $\tilde{x}$, had Y been y , given the fact that $Z = z(X = x)$. Note that the given “fact” is $Z = z(x)$ and the “counter-fact” (what if) is $Y \neq y(x)$, indicating that the encoded $y(x)$ — also an observed fact — clashes with the counter-fact y . The key difference between the conventional unseen generation and the counterfactual is that: the former is purely based on the **sample-agnostic** class attributes y (or together with Guassian prior z [154]), while the latter is also grounded by the **sample-specific** attributes $z(x)$.

So, why sample-specific? The imagination of an unseen-class is indeed not necessary to start from scratch, *i.e.*, purely from the class definition Y , where the effect of some attributes may be lost due to entanglement [33]; instead, it should

also start from the observed fact $Z = z(x)$, which can make up those entangled attributes. In fact, such grounded generation simulates how we humans imagine an unseen sample [248]: we imagine a “dinosaur” (class attributes) based on its fossil (sample attributes). More formally, by disentangling the two groups: class attribute Y and sample Z , we can use the theorem of **Counterfactual Faithfulness** (Section 5.2.4 & 5.2.5) to guarantee that the counterfactual distribution is coherent with the ground-truth seen/unseen distribution. As shown in Figure 5.3c (left), the unseen-class generation grounded by a sample is in the true unseen domain, leading to a more balanced decision boundary. It is worth noting that the group disentanglement between Z and Y is much more relaxed and thus more approachable than the full disentanglement [233, 310], *i.e.*, the attributes within each group are not required to be disentangled. In Section 5.2.5, we design a training procedure to achieve this.

We propose a counterfactual framework for ZSL and OSR, because they are both underpinned by the generalization to unseen-classes. As summarized in Figure 5.2, the counterfactual is powered by a Generative Causal Model (Section 5.2.3). The counterfactual faithfulness allows us to use the **Consistency Rule**: if the counterfact y is indeed the underlying ground-truth, the counterfactual $\tilde{x}$ equals to the factual x . Therefore, we can use the rule as a seen/unseen binary classifier by varying y across the seen/unseen class attributes (Section 5.2.4): If a sample is seen, we can apply the conventional supervised learning classifier; otherwise, for ZSL, we apply the conventional ZSL classifier, and for OSR, we reject it.

To the best of our knowledge, the proposed counterfactual framework is the first to provide a theoretical ground for balancing and improving the seen/unseen classification. In particular, we show that the quality of disentangling Z and Y is the key bottleneck, so it is a potential future direction for ZSL/OSR [38, 167, 231, 311]. Our method can serve as an unseen/seen binary classifier, which can plug-and-play and boost existing ZSL/OSR methods to achieve new state-of-the-arts (Section 5.2.6).

5.2.2 Problem Formulations

We train a model using a labelled dataset $\mathcal{D} = \{\mathbf{x}_i, l_i\}_{i=1}^N$ on seen-classes $\mathcal{S}$, to recognize samples from both $\mathcal{S}$ and unseen-classes $\mathcal{U}$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ is the d -dimensional feature vector of the i -th sample and $l_i \in \mathcal{S}$ is its corresponding

label. We assume that the samples of $\mathcal{S}$ and $\mathcal{U}$ are embedded in the same feature space $\mathcal{X}$, whose ambient space is the RGB color space or the network feature space provided by the dataset curator [29].

Zero-Shot Learning (ZSL). It includes two settings: 1) *Conventional ZSL*, where the model is only evaluated on $\mathcal{U}$, and 2) *Generalized ZSL*, where the model is evaluated on $\mathcal{S} \cup \mathcal{U}$. A common practice [305] is to use an additional set of class attribute vectors $\mathcal{Y}_S$ and $\mathcal{Y}_U$ to describe the seen- and unseen-classes, respectively. Compared to the one-hot label embeddings, these attributes can be considered as dense label embeddings [312, 313]. When the context is clear, we refer to ZSL as Generalized ZSL.

Open-Set Recognition (OSR). It is evaluated on both $\mathcal{S}$ and $\mathcal{U}$. Compared to ZSL, $\mathcal{U}$ in OSR is marked as “unknown”. Instead of using dense labels, each seen-class is described by the one-hot embedding of K dimensions, where K is the number of seen-classes. The one-hot embeddings are considered as the seen-class attributes set $\mathcal{Y}_S$. Since OSR is evaluated in an open environment, there is no unseen-class attributes set $\mathcal{Y}_U$.

5.2.3 Generative Causal Model

Our assumption is that both ZSL and OSR follow a Generative Causal Model (GCM) [40] shown in Figure 5.4, where the *class attribute* Y and the *sample attribute* Z jointly determine the observed image feature X . In general, the generative causal process $Z \rightarrow X, Y \rightarrow X$ can be confounded, represented as the dashed links in Figure 5.4. One can remove the confounders through more elaborate data collection [29] and experimental design [314]. In this paper, we follow the conventions in

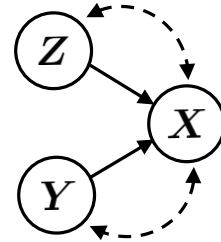


FIGURE 5.4: Our GCM for ZSL and OSR.

ZSL and OSR to ignore the confounders [48, 155]. Our GCM entails a generation and inference process. Specifically, given Z and Y , we can generate X by sampling from the conditional distribution $P_\theta(X|Z, Y)$. While given X , we can infer Z and Y through the posterior $Q_\phi(Z|X)$ and $Q_\psi(Y|X)$. Therefore, GCM can support both generative and inference-based methods. As the former prevails over the latter in literature, we focus on it in this paper.

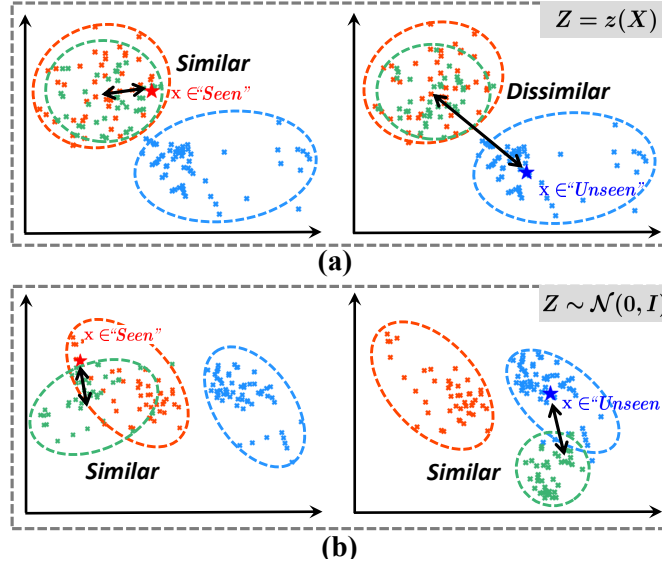


FIGURE 5.5: OSR using (a) our counterfactual framework ($Z=z(X)$); (b) existing generative approach ($Z \sim \mathcal{N}(0, I)$) when the test sample is from seen-classes (left) and unseen-classes (right). Red dashed line denotes the true samples from the seen-class and blue dash line as those from the unseen-class. Green dash line denotes generated samples using the seen-class attribute. $\star$ and $\star$ denote the samples from seen and unseen class, respectively.

In ZSL, Y takes values from the dense labels set $\mathcal{Y}_S \cup \mathcal{Y}_U$. Generative ZSL methods [154, 155] aim to learn $P_\theta(X|Z, Y)$. In testing, they generate unseen-class samples X by using a Gaussian prior as Z and the attributes Y in $\mathcal{Y}_U$. Then, the generated unseen samples and the seen $\mathcal{D}$ are used to train a classifier to recognize both $\mathcal{S}$ and $\mathcal{U}$. In OSR, the value of Y is from the one-hot embedding set $\mathcal{Y}_S$. Generative OSR methods [18, 48] first infer Y from a test sample $\mathbf{x}$ by sampling $\mathbf{y} \sim Q_\psi(Y|X = \mathbf{x})$. Then, the inferred $\mathbf{y}$ and Gaussian noise $\mathbf{z}$ are used to generate $\mathbf{x}'$ from $P_\theta(X|Z = \mathbf{z}, Y = \mathbf{y})$. Finally, the sample is marked as “unknown” if $\mathbf{x}$ is dissimilar to $\mathbf{x}'$ subject to a specified threshold.

Note that all the above generation methods adopt a prior Gaussian noise for Z , which is not sample-specific. As Y is inevitably entangled (examples in Section 5.2.9), there is no mechanism to make up for the missing sample attribute effect during generation, so, the generated unseen-class samples will be unrealistic and lie outside $\mathcal{X}$, rendering the decision rule trained on the seen samples inapplicable.

5.2.4 Counterfactual Generation and Inference

Given a sample $\mathbf{x}$, we use the GCM to generate counterfactual samples $\tilde{\mathbf{x}} = X_{\mathbf{y}}[z(\mathbf{x})]$ following the three steps of computing counterfactual [115]:

Abduction- “*given the fact that $Z = z(\mathbf{x})$* ”. We solve for the endogenous sample attribute $z(\mathbf{x})$ given the evidence $X = \mathbf{x}$. In our GCM, we can sample from the posterior $z(\mathbf{x}) \sim Q_{\phi}(Z|X = \mathbf{x})$.

Action- “*had Y been y* ”. Here $\mathbf{y} \in \mathcal{Y}_S \cup \mathcal{Y}_U$ is the intervention target of Y . Note that the class attribute Y of this sample can be inferred from $y(\mathbf{x}) \sim Q_{\psi}(Y|X = \mathbf{x})$, but in this step, we intervene on Y by discarding the inferred value and setting Y as $\mathbf{y}$, which may be different from $y(\mathbf{x})$.

Prediction- “ *X would be $\tilde{\mathbf{x}}$* ”. Conditioning on the inferred $Z = z(\mathbf{x})$ (fact) and the intervention target $Y = \mathbf{y}$ (counter-fact), we can generate the counterfactual sample $\tilde{\mathbf{x}}$ from $P_{\theta}(X|Z = z(\mathbf{x}), Y = \mathbf{y})$. Notice that the feature imagination from class attribute $Y = \mathbf{y}$ is now grounded by the sample attribute $Z = z(\mathbf{x})$.

We want that the above counterfactual sample lies in the true distribution of the seen or unseen samples. Formally, such property is defined as below:

Definition (Counterfactual Faithfulness). *Given $\mathbf{x} \in \mathcal{X}$, the counterfactual generation $\tilde{\mathbf{x}}$ using a GCM is faithful whenever $\tilde{\mathbf{x}} \in \mathcal{X}$.*

In fact, the failure of existing methods, as mentioned in Section 5.2.3, is because the counterfactual faithfulness does not hold; however, it holds for our generation. We delay the justification to Section 5.2.5. It assures that any distance metric in $\mathcal{X}$ is applicable for both $\mathbf{x}$ and its counterfactual generation $\tilde{\mathbf{x}}$. This allows us to build a binary classifier on seen/unseen by applying the **Consistency Rule**:

$$y^*(\mathbf{x}) = \mathbf{y} \implies X_{\mathbf{y}}[z(\mathbf{x})] = \mathbf{x}, \quad (5.1)$$

where $y^*(\mathbf{x})$ is the (unobserved) ground-truth class attribute of $\mathbf{x}$ and $X_{\mathbf{y}}[z(\mathbf{x})]$ is the generated counterfactual with this ground-truth. Our binary classification strategy is based on the equivalent *contraposition* of the rule, which states that *if $\mathbf{x}$ is dissimilar to $X_{\mathbf{y}}[z(\mathbf{x})]$, the ground-truth attribute of $\mathbf{x}$ cannot be $\mathbf{y}$* :

$$X_{\mathbf{y}}[z(\mathbf{x})] \neq \mathbf{x} \implies y^*(\mathbf{x}) \neq \mathbf{y}. \quad (5.2)$$

Thanks to the counterfactual faithfulness, the dissimilarity can be measured by any distance metric defined in $\mathcal{X}$ (e.g., Euclidean distance). Figure 5.5 uses the OSR task to show the benefit of our counterfactual approach grounded by $Z = z(X)$. The class-agnostic Z in the existing methods cannot make up for the attributes that entangled in Y . This leads to non-faithful generation on unseen-class samples, and the distance can hardly tell apart the seen and unseen-class samples (Figure 5.5b). In contrast, our approach achieves counterfactual faithfulness and thus the distance remains discriminative (Figure 5.5a). Next, we detail the binary inference rule for both ZSL and OSR.

Inference in ZSL. Since ZSL is evaluated in a closed environment, *i.e.*, the set of unseen-class attributes $\mathcal{Y}_U$ is known in testing, we use the contraposition—if feature $\mathbf{x}$ is *dissimilar* to counterfactual generations from the *unseen*-classes, $\mathbf{x}$ belongs to *seen*. For example in Figure 5.3c, conditioning on a seen-class sample $\mathbf{x}$, the counterfactual generations using the unseen-class attribute are indeed dissimilar to $\mathbf{x}$ as they lie in the opposite side of the classifier decision boundary. Specifically, we generate a set of counterfactual features $\tilde{X}$ of the *unseen*-classes by using the inferred Z from $Q_\phi(Z|X = \mathbf{x})$ and intervening Y with the ground-truth attributes from $\mathcal{Y}^U$. Using the counterfactual set $\tilde{X}$ of the unseen-classes and $\mathcal{D}$ of the seen-classes, we train a multi-label classifier whose vocabulary is $\mathcal{S} \cup \mathcal{U}$. Denote the mean-pooling of the top- K classifier probabilities among seen- and unseen-classes as S^K and U^K , respectively. The binary seen/unseen label $b(\mathbf{x})$ is given by:

$$b(\mathbf{x}) = \begin{cases} \text{seen}, & \text{if } U^K < S^K \\ \text{unseen}, & \text{otherwise} \end{cases} \quad (5.3)$$

Inference in OSR. Since OSR is in an open environment, *i.e.*, there could be infinite number of unseen-classes, it is impossible to generate unseen-class counterfactuals. Therefore, we use the contraposition the other way round—if $\mathbf{x}$ is *dissimilar* to the counterfactual generations of the *seen*-classes, $\mathbf{x}$ belongs to the *unseen*. This is shown in Figure 5.5a, where the unseen-class sample (right), compared to the seen-class sample (left), is much more dissimilar to the seen-class counterfactuals. Hence by thresholding the dissimilarity, we can classify both samples correctly. Specifically, we generate a set of counterfactual features $\tilde{X}$ on *seen*-classes by setting Y as the one-hot embeddings in $\mathcal{Y}_S$, and we calculate the minimum Euclidean

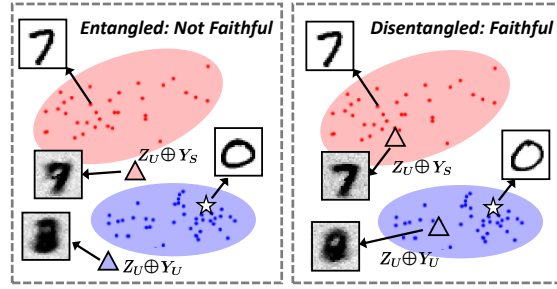


FIGURE 5.6: Counterfactual generation conditioned on an unseen sample (denoted as the star). $Z_U \oplus Y_S$ and $Z_U \oplus Y_U$ represent counterfactual images conditioned on the unseen sample, using seen and unseen attribute respectively.

distance between $\mathbf{x}$ and each $\tilde{\mathbf{x}} \in \tilde{X}$, denoted as d_{min} . If d_{min} is larger than a threshold τ , the sample is dissimilar to the seen-classes counterfactuals. The binary label $b(\mathbf{x})$ is given by:

$$b(\mathbf{x}) = \begin{cases} \text{unseen,} & \text{if } d_{min} > \tau \\ \text{seen,} & \text{otherwise} \end{cases} \quad (5.4)$$

Two-Stage Inference. As shown in Figure 5.2, after the stage-one binary classification, at the second stage, the predicted seen-classes samples can be classified by using the standard supervised classifier, and the predicted unseen-classes samples can be fed into any Conventional ZSL algorithms in ZSL, or rejected as outliers in OSR. The detailed implementation is in Section 5.2.6.

5.2.5 Counterfactual-Faithful Training

Our framework centers on the counterfactual-faithful generation, which can be guaranteed by the following theorem (proved as a corollary of [233], Section 5.2.9):

Theorem. *The counterfactual generation $X_{\mathbf{y}}[z(\mathbf{x})]$ is faithful if and only if the sample attribute Z and class attribute Y are group disentangled.*

Figure 5.6 shows the difference between the generations with and without the group disentanglement. Therefore, to achieve counterfactual faithful generation, the full disentanglement among each dimension of Y and Z , which may be impossible in general [47], is relaxed to group disentanglement between Y and Z , which can be more easily addressed by the following specially designed training objective:

$$\min_{\theta, \phi} \mathcal{L}_Z + \nu \mathcal{L}_Y + \max_{\omega} \rho \mathcal{L}_F, \quad (5.5)$$

where ν, ρ are a trade-off parameters. All three losses are designed to achieve the counterfactual faithfulness, which assures the distance metric of the consistency rule. Next, we detail each of them.

Disentangling Z from Y . We minimize the β -VAE [183] loss $\mathcal{L}_Z$ given by:

$$\begin{aligned} \mathcal{L}_Z = & -\mathbb{E}_{Q_\phi(Z|X)} [P_\theta(X | Z, Y)] \\ & + \beta D_{KL}(Q_\phi(Z | X) \parallel P(Z)), \end{aligned} \quad (5.6)$$

where D_{KL} denotes KL-divergence, $P_\theta(X|Z, Y)$ and $Q_\phi(Z|X)$ are implemented using the Deep Gaussian family [235], and the prior $P(Z)$ is set to the isotropic Gaussian distribution. Compared to the standard VAE objective, β -VAE re-weights the KL divergence term by a factor of β ($\beta > 1$). This is shown to be highly effective in learning a disentangled sample attribute Z [167, 183]. Intuitively, by placing a strict constraint for Z as a whole to follow the endogenous prior $P(Z)$, the value of Z becomes unaffected by the distribution of Y , *i.e.*, Z is disentangled from Y .

Disentangling Y from Z . However, the above regularization cannot guarantee that the GCM correctly uses the class attribute Y during generation. For example, recent GAN models [8, 315] can generate a large variety of photo-realistic images by only using Z . In fact, as shown in recent literature [311], it is possible for an over-parameterized model $P_\theta(X|Z, Y)$ to ignore Y and use purely Z to generate X , leading to non-faithful generations. This is because the information in Y might be *fully contained* in Z . Therefore, it is necessary to additionally disentangle Y from Z . Specifically, given a training sample $\mathbf{x}$, its ground-truth attribute $\mathbf{y}$ and its sample attribute $z(\mathbf{x})$ sampled from $Q_\phi(Z|X = \mathbf{x})$, we require $\mathbf{x}$ to be close to $\mathbf{x}_\mathbf{y} = X_\mathbf{y}[z(\mathbf{x})]$, but far away from the counterfactual generations in the set $\tilde{X} = \{X_{\mathbf{y}'}[z(\mathbf{x})] \mid \mathbf{y}' \in \mathcal{Y}^S \wedge \mathbf{y}' \neq \mathbf{y}\}$. We use a contrastive loss as:

$$\mathcal{L}_Y = -\log \frac{\exp(-\text{dist}(\mathbf{x}, \mathbf{x}_\mathbf{y}))}{\sum_{\mathbf{x}' \in \tilde{X} \cup \{\mathbf{x}_\mathbf{y}\}} \exp(-\text{dist}(\mathbf{x}, \mathbf{x}'))}, \quad (5.7)$$

where dist denotes Euclidean distance. Intuitively, this loss actively intervenes the class attribute Y , given fixed sample attributes Z , and therefore maximizes the sample difference (*e.g.*, in terms of contrastiveness) before and after the intervention, which complies with the recent definition of causal disentanglement [38]. Therefore, $\mathcal{L}_Y$ can disentangles Y from Z .

Further Disentangling by Faithfulness. Note that the faithfulness $\tilde{\mathbf{x}} \in \mathcal{X}$ is a necessary condition for disentanglement, so we can also use it as an objective. Note that the VAE objective optimizes a lower bound of the likelihood $P(X)$, where the bound looseness undermines the faithfulness. To address this problem, we additionally use the Wasserstein GAN (WGAN) [316] loss. Specifically, we train a discriminator $D(X, Y)$ parameterized by ω that outputs a real value, indicating if feature X is realistic conditioned on Y (larger is more realistic). Given a feature $\mathbf{x}$ with attribute $\mathbf{y}$, we can generate the counterfactual $\mathbf{x}'$ using $z(\mathbf{x})$ and $\mathbf{y}$. The discriminator is trained to mark $\mathbf{x}$ as real (large $D(\mathbf{x}, \mathbf{y})$) and $\mathbf{x}'$ as unreal (small $D(\mathbf{x}', \mathbf{y})$). Specifically, the WGAN loss $\mathcal{L}_F$ is given by:

$$\begin{aligned} \mathcal{L}_F = & \mathbb{E}[D(\mathbf{x}, \mathbf{y})] - \mathbb{E}[D(\mathbf{x}', \mathbf{y})] \\ & - \lambda \mathbb{E} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}, \mathbf{y})\|_2 - 1)^2], \end{aligned} \quad (5.8)$$

where λ is a penalty term, $\hat{\mathbf{x}} = \alpha \mathbf{x} + (1 - \alpha) \mathbf{x}'$ with α sampled from the uniform distribution defined on $[0, 1]$. By training the generation process $P_\theta(X|Z, Y)$ and the discriminator in an adversarial fashion, the GCM is regularized to generate $\mathbf{x}'$ that is similar to $\mathbf{x}$ in the same distribution, *i.e.*, counterfactual-faithful for seen-class samples.

5.2.6 Experiments

ZSL Datasets. We evaluated our method on standard benchmark datasets: Caltech-UCSD-Birds 200-2011 (*CUB*) [16], *SUN* [317], Animals with Attributes 2 (*AWA2*) [29] and attribute Pascal and Yahoo (*aPY*) [318]. In particular, we followed the unseen/seen split in the *Proposed Split (PS) V2.0* [29], recently released to fix a test-data-leaking bug in the original PS.

OSR Datasets. We used the standard evaluation datasets: *MNIST* [319], *SVHN* [320], *CIFAR10* [20] and *CIFAR100* [20]. Following the standard benchmark [321, 322] in OSR, we split MNIST, SVHN and CIFAR10 into 6 seen-classes and 4 unseen-classes, and construct two additional datasets CIFAR+10 (*C+10*) and CIFAR+50 (*C+50*), where 4 non-animal classes in CIFAR10 are used as seen-classes, while additional 10 and 50 animal classes from CIFAR100 are used as unseen-classes.

	Method	CUB			AWA2			SUN			aPY		
		<i>U</i>	<i>S</i>	<i>H</i>	<i>U</i>	<i>S</i>	<i>H</i>	<i>U</i>	<i>S</i>	<i>H</i>	<i>U</i>	<i>S</i>	<i>H</i>
Inf.	ALE [†] [152]	23.7	62.8	34.4	14.0	81.8	23.9	21.8	33.1	26.3	4.6	73.7	8.7
	DEWISE [†] [151]	23.8	53.0	32.8	17.1	74.7	27.8	16.9	27.4	20.9	3.5	78.4	6.7
	LATEM [†] [323]	15.2	57.3	24.0	11.5	77.3	20.0	14.7	28.8	19.5	1.3	71.4	2.6
	RelationNet [324]	36.3	63.8	46.3	22.1	91.4	35.5	15.8	25.5	19.6	11.5	80.7	20.2
Gen.	GDAN [26]	35.0	28.7	31.6	26.0	78.5	39.1	38.2	19.8	26.1	29.0	63.7	39.9
	CADA-VAE [325]	50.3	56.1	53.0	55.4	76.1	64.0	43.6	36.4	39.7	34.0	54.2	41.7
	LisGAN [155]	44.9	59.3	51.1	53.1	68.8	60.0	41.9	37.8	39.8	33.2	56.9	41.9
	TF-VAEGAN [17]	50.7	62.5	56.0	52.5	82.4	64.1	41.0	39.1	40.0	31.7	61.5	41.8
	GCM-CF (Ours)	61.0	59.7	60.3	60.4	75.1	67.0	47.9	37.8	42.2	37.1	56.8	44.9

Table 5.1: ZSL Accuracy ($U\%$, $S\%$, $H\%$) on the four datasets, where Inf. means inference-based methods and Gen. means generation-based methods. Note that PS V2.0 was released recently in <https://drive.google.com/file/d/1p9gtkuHCCCykyezSarCw-1siCSXUyKH/view> to fix a test-data leaking bug. This can have large impacts on the performance of existing methods, such as GDAN [26]. Therefore all our evaluations were conducted on PS V2.0. [†] indicates that the results are taken from the PS V2.0 report [27], and we reproduced the results on all other methods using the official code. For our method GCM-CF, we used AREN [28] for Conventional ZSL on CUB, and otherwise used TF-VAEGAN [17] for supervised classification and Conventional ZSL.

ZSL Evaluation. It was conducted in the Generalized ZSL setting. We used two metrics: 1) **ZSL Accuracy**. It consists of 3 numbers (U, S, H), where U/S is the per-class top-1 accuracy of unseen-/seen-classes test samples, and H is the harmonic mean of U, S , given by $H = 2 \times S \times U / (S + U)$. 2) **CVb**. To measure the balance between unseen/seen classification, we propose to use the Coefficient of Variation of the seen and unseen binary classification accuracy, denoted as CVb. Let S_b and U_b be the binary accuracy on seen- and unseen-classes, respectively. CVb is given by $\sqrt{0.5(S_b - \mu)^2 + 0.5(U_b - \mu)^2} / \mu$, where $\mu = (S_b + U_b) / 2$. Note that the variation between S and U of ZSL Accuracy is not a good measure of balance, as they are affected by the number of seen- and unseen-classes, which can be quite different (see SUN in Table A1). 3) **AUSUC**. We draw the Seen-Unseen accuracy Curve (SUC) by plotting a series of S against U of ZSL Accuracy, where the series is obtained by adjusting a calibration factor ω that is subtracted from the classifier logits on the seen-classes. Then we use the Area Under SUC (AUSUC) for evaluation. Compared to a single ZSL Accuracy, SUC and the area provide a more detailed view of the capability of an algorithm to balance the unseen-seen decision boundary [33, 305].

OSR Evaluation. We used the following metrics: 1) Macro-averaged **F1 scores** over seen-classes and “unknown” (for all unseen-classes), which shows how well a method can recognize seen classes while rejecting unseen-classes samples; 2) **Openness-F1 Plot**. We also studied the response of F1 scores under varying

openness given by $1 - \sqrt{2N/(N+M)}$, where N and M are number of seen- and unseen-classes, respectively. Compared to a single F1 score where the openness is fixed, this plot shows the robustness of an OSR classifier to the open environment with an unknown number of unseen-classes.

Implementation Details. For ZSL, we implemented our GCM based on the network architecture in TF-VAEGAN [17]. Following common protocol [154, 155], we used the ResNet-101 [22] features for X and attributes provided in [29] for $\mathcal{Y}_S, \mathcal{Y}_U$. For OSR, our GCM was implemented using the networks in CGDL [18, 326] and X represents actual images.

5.2.7 Results on ZSL

Method	CUB	AWA2	SUN	aPY
GDAN [16]	15.1	25.1	27.3	14.4
CADA-VAE [317]	6.7	6.0	7.3	11.2
LisGAN [29]	4.9	7.0	4.6	4.0
TF-VAEGAN [318]	8.0	9.4	10.2	5.8
GCM-CF (Ours)	1.5	2.1	1.0	2.3

Table 5.2: CVb (%) of generative models on all datasets.

Mitigate the Imbalance. As shown in Table 5.1, our counterfactual approach, denoted as GCM-CF, achieves a **more balanced ZSL Accuracy** and significantly improves the existing state-of-the-art (SOTA) by 2.2% to 4.3%, with a much higher score on U . For example, compared to LisGAN on aPY, our method gains 3.9% on U while sacrificing only 0.1% on S . To further show that GCM-CF mitigates the unseen/seen imbalance, we diagnosed the binary classification accuracy using CVb in Table 5.2. Note that existing methods have very large CVb, which means that there is a large difference between seen and unseen classification accuracy and reveals the imbalance problem. Our method has the **lowest CVb**. This shows that our approach indeed achieves a more balanced binary decision boundary between seen and unseen. However, one may argue that the imbalance problem can be solved by simply adjusting the calibration factor ω . Therefore, we plot the SUC using varying ω and measured the AUSUC on all datasets. The result is shown in Figure 5.7, where GCM-CF outperforms other methods in every inch and achieves **the best AUSUC**. This shows that GCM-CF fundamentally improves the unseen/seen classification beyond the reach of simple calibration. Overall, the balanced and much improved ZSL Accuracy, lower CVb for binary classification, and higher accuracy on varying calibrations demonstrate that our method

Stage 2 \ Stage 1	TF-VAEGAN [17]			GCM-CF (Ours)		
	<i>U</i>	<i>S</i>	<i>H</i>	<i>U</i>	<i>S</i>	<i>H</i>
RelationNet [324]	49.3	81.2	61.3	55.8	75.0	64.0
CADA-VAE [325]	49.5	81.1	61.5	57.6	75.0	65.2
LisGAN [155]	48.8	80.4	60.7	56.1	74.3	63.9
TF-VAEGAN [17]	52.8	83.2	64.6	60.4	75.1	67.0

Table 5.3: Comparison of the two-stage inference performance on AWA2 [29] using TF-VAEGAN [17] and our GCM-CF as the stage-one binary classifier. The results on other datasets are shown in Appendix, where using GCM-CF as stage-one binary classifier improves all the methods.

mitigates the unseen/seen imbalance in ZSL. This indicates that our GCM generates faithful counterfactuals (see Figure 5.3c) and supports the effectiveness of our counterfactual-faithful training in disentangling Z and Y .

Stage-One Binary Classifier. Our GCM-CF can serve as a stage-one binary unseen/seen classifier and plug into *all* ZSL methods for subsequent supervised learning on seen and conventional ZSL on unseen. We performed experiments on 4 representative methods—inference based: RelationNet [324], generation with VAE: CADA-VAE [325], with GAN: LisGAN [155] and with VAE-GAN [327]: TF-VAEGAN [17]. The ZSL Accuracy on AWA2 is shown in Table 5.3. By comparing with Table 5.1, we observe that GCM-CF can significantly improve all of them on H . For comparison, we used the current SOTA ZSL method TF-VAEGAN as a binary unseen/seen classifier and performed the same experiments. The results on the ZSL Accuracy show that our GCM-CF significantly outperforms TF-VAEGAN. This demonstrates that compared to the class-agnostic Z in existing methods, the use of sample attribute Z that is disentangled from class attribute Y in our GCM-CF is highly effective in improving ZSL performance. Overall, our method as a robust binary classifier can serve as a new strong baseline to evaluate ZSL methods using two-stage inference.

5.2.8 Results on OSR

Strong Open-Set Classifier. In Table 5.4, our GCM-CF achieves SOTA F1 scores in all datasets. This strongly supports that GCM-CF improves F1 scores in general, *i.e.*, not only on some particular splits. Furthermore, a strong open-set classifier should stay robust regardless of the number of unseen-classes during evaluation. Therefore, we evaluate OSR classifiers with the Openness-F1 Plot. As

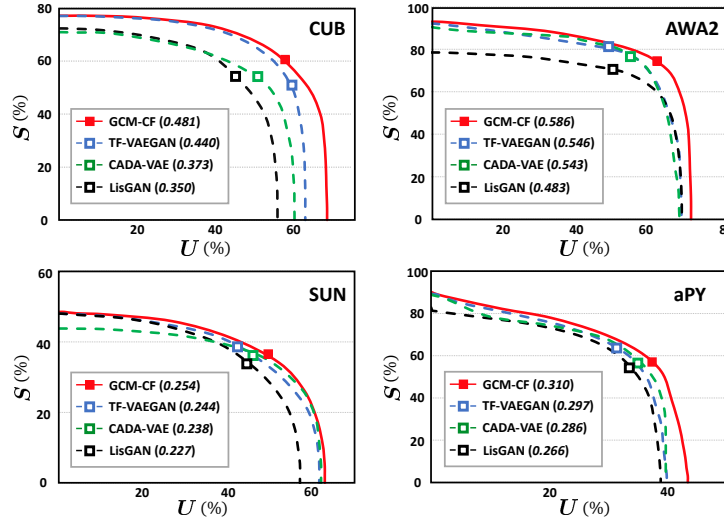


Figure 5.7: The Seen-Unseen accuracy Curve (SUC) together with the Area Under SUC (AUSUC) on the four datasets. The rectangle denotes the ZSL Accuracy when calibration factor $\omega=0$.

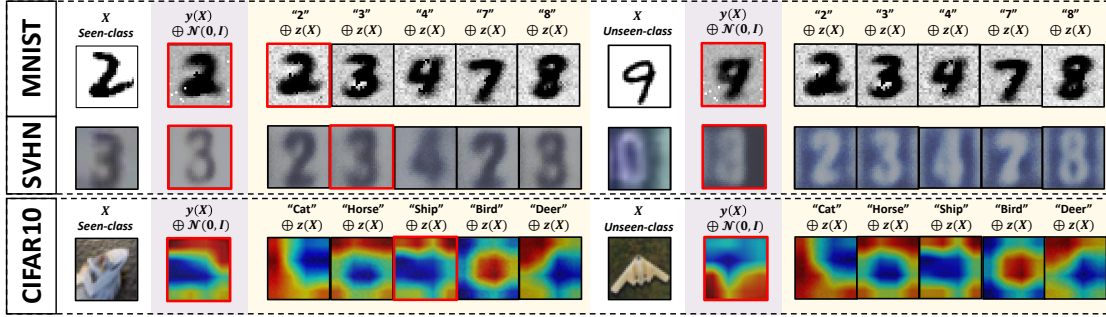


Figure 5.8: Comparison of the reconstructed images using CGDL [18] ($y(X) \oplus \mathcal{N}(0, I)$) and the counterfactual images generated from our GCM-CF ($y \oplus z(X)$) on the seen- and unseen-class samples. The red box on a generated image denotes that it is similar to the original. For CIFAR10, due to the conflict between visual perception and discriminative training [19], we first train a classifier on seen-classes images and then use CAM [5] to show the sensible yet discriminative features: although the pixel-level generation is not sensible, the pixel-level distance remains discriminative as shown in Table 5.4.

Method	MNIST	SVHN	CIFAR10	C+10	C+50
Softmax	76.82	76.16	70.39	77.82	65.96
OpenMax [30]	85.93	77.95	71.38	78.68	67.68
CGDL [18]	88.95	76.31	71.03	77.92	70.96
GCM-CF (Ours)	91.37	79.25	72.63	79.38	74.60

Table 5.4: Comparison of the F1 score averaged over 5 random splits in OSR. We used the official code on CGDL [18] and implemented Softmax and OpenMax [30] for evaluation. For GCM-CF, after binary classification, we used CGDL for supervised classification on the seen-classes.

shown in Figure 5.9, our GCM-CF achieves the highest F1 score on all openness settings, and our performance is especially competitive in the challenging environment with large openness. Overall, these results clearly demonstrate the effectiveness of our GCM-CF on seen-unseen classification in OSR.

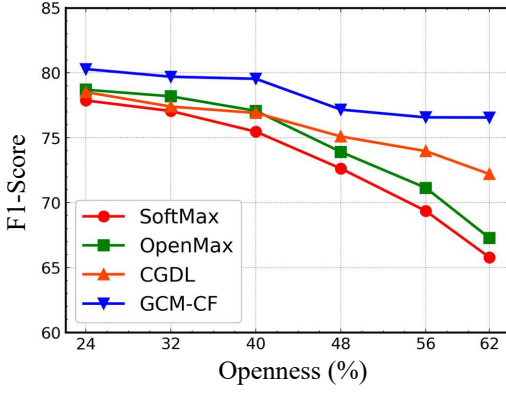


Figure 5.9: Openness-F1 Plot where 4 non-animal classes from CIFAR10 [20] were used as seen-classes and various classes were drawn from CIFAR100 [20] as unseen-classes.

Qualitative Results. Figure 5.8 shows OSR evaluation using existing reconstruction-based approach and our counterfactual approach. Notice the reconstructed image on the unseen-class sample (*e.g.*, “0, 9”) can be similar to the original image. This makes it difficult to tell apart unseen and seen by thresholding reconstruction error. By using our GCM-CF to generate counterfactuals on each seen-class, the generated images are counterfactual-faithful and indeed look like seen-classes. Note that this holds on CIFAR10, where given seen- or unseen-class samples, the CAMs of the generated counterfactuals using the same seen-class attribute look similar. Therefore, by thresholding the dissimilarity between the test sample and its counterfactuals, we can easily distinguish unseen from seen. These results show the effectiveness of our binary inference strategy. Interestingly, we also observe that the generated samples from GCM indeed capture sample attributes better (*e.g.*, text color, background color on SVHN), which validates that we can learn a disentangled Z using the proposed counterfactual-faithful training.

5.2.9 Supplementary Information

Proof of the Theorem in 3.4

Let $V = (Z, Y)$ and V takes values from the space $\mathcal{V}$. For class attribute Y that is a K -dimensional real vector, denote $\mathcal{E} = \{e_1, \dots, e_K\}$ as the subset of dimensions spanned by the class attribute Y , and $\bar{\mathcal{E}}$ as those by the sample attribute Z . Therefore, $\mathcal{V}^{\mathcal{E}}$ and $\mathcal{V}^{\bar{\mathcal{E}}}$ represent the space of class attribute Y and sample attribute Z , respectively. We use $g : \mathcal{V} \rightarrow \mathcal{X}$ to denote the endogenous mapping to the feature space $\mathcal{X}$. Note that this g corresponds to sampling from $P_{\theta}(X|Z, Y)$ in our GCM. We will introduce the concept of embedded GCM to facilitate the proof.

Definition (Embedded GCM). *We say that a GCM is embedded if $g : \mathcal{V} \rightarrow \mathcal{X}$ is a continuous injective function with continuous inversion g^{-1} .*

Using the results in [328], if V is compact (*i.e.*, bounded), the GCM is embedded if and only if g is injective. Though we implement our GCM using VAE, whose latent space is not compact, it is shown that restricting a VAE to a product of compact intervals that covers most of the probability mass (using KL-divergence in the objective) will result in an embedded GCM that approximates the original one for most samples [233]. Moreover, $P_\theta(X|Z, Y)$ are implemented using deterministic mappings in our model, which are indeed injective as shown in [249]. Therefore, without loss of generality, our GCM can be considered as embedded. We will give the formal definition of intrinsic disentanglement, which can be used to show that group disentanglement of Z and Y leads to faithfulness.

Definition (Intrinsic Disentanglement). *In a GCM, the endomorphism $T : \mathcal{X} \rightarrow \mathcal{X}$ is intrinsically disentangled with respect to the subset $\mathcal{E}$ of endogenous variables, if there exists a transformation T' affecting only variables indexed by $\mathcal{E}$, such that for any $\mathbf{v} \in \mathcal{V}$,*

$$T(g(\mathbf{v})) = g(T'(\mathbf{v})). \quad (5.9)$$

Now we will first establish the equivalence between intrinsic disentanglement and faithfulness using the theorem below.

Theorem (Intrinsic Disentanglement and Faithfulness). *The counterfactual mapping $X_{\mathbf{y}}[z(X)]$ is faithful if and only if it is intrinsically disentangled with respect to the subset $\mathcal{E}$.*

To prove the above theorem, one conditional is trivial: if a transformation is intrinsically disentangled, it is by definition an endomorphism of $\mathcal{X}$ so the counterfactual mapping must be faithful. For the second conditional, let us assume a faithful counterfactual mapping $X_{\mathbf{y}}[z(X)]$. Based on the three steps of computing counterfactuals and the embedding property discussed earlier, the counterfactual mapping can be decomposed as:

$$X_{\mathbf{y}}[z(X)] = g \circ T' \circ g^{-1}(X), \quad (5.10)$$

where $\circ$ denotes function composition, T' is the counterfactual transformation of V as defined in Section 3.2, where Z is kept as $Z = z(X)$ and Y is set as $\mathbf{y}$. Now

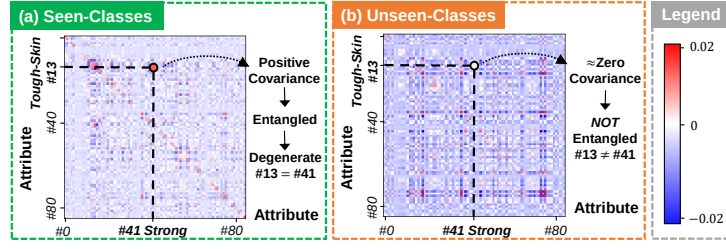


FIGURE 5.10: Visualization of the covariance matrix in AWA2 dataset of the (a) seen-classes and (b) unseen-classes attributes.

for any $\mathbf{v} \in \mathcal{V}$, the quantity $X_{\mathbf{y}}[z(g(\mathbf{v}))]$ can be similarly decomposed as:

$$X_{\mathbf{y}}[z(g(\mathbf{v}))] = g \circ T' \circ g^{-1} \circ g(\mathbf{v}) = g \circ T'(\mathbf{v}). \quad (5.11)$$

Since T' is a transformation that only affects variables in $\mathcal{E}$ (*i.e.*, Y), we show that faithful counterfactual transformation $X_{\mathbf{y}}[z(X)]$ is intrinsically disentangled with respect to $\mathcal{E}$.

In our work, we define group disentanglement of Z and Y as intrinsic disentanglement with respect to the set of variables in Y . We used the sufficient condition by learning a GCM where Y and Z are group disentangled, such that when we fix Z and only change Y , the resulting generation lies in $\mathcal{X}$ according to the theorem that we just proved. We also exploited the necessary condition by training for faithfulness through the WGAN loss, which in turn encourages the group disentanglement of Z and Y .

Entanglement of Y . Two class attributes are entangled when they always or never appear together, and they effectively degenerate to one feature. If two attributes are entangled among seen-classes but not unseen, the degenerated feature learnt from seen cannot tell unseen-classes apart. Entanglement can be identified by finding pairs of attributes with large absolute covariance. For example, Figure 5.10 shows a “strong” animal usually has “tough skin” among seen-classes. Yet the co-appearance no longer holds for unseen-classes as shown in Figure 5.10.

5.3 Case Study: Unsupervised Learning

5.3.1 Introduction

A good feature representation should faithfully capture the underlying generative attributes in a compact and modular vector space [38, 210, 263], enabling not only sample inference (*e.g.*, image classification), but also counterfactual generation [44] (*e.g.*, image synthesis of unseen attribute combinations). Over the past decade, discriminative training is the feature learning mainstream that has exceptional performance in inference tasks [22, 329]. However, it hardly achieves faithful generation due to the deliberate discard of certain attributes, *e.g.*, class-irrelevant attributes in supervised learning or augmentation-related attributes in contrastive learning.

On the other hand, generative Denoising Diffusion Probabilistic Model (DM) [330, 331] can retain all the underlying attributes for faithful generation [332, 333], or even extrapolate novel attribute combinations by textual control [334, 335] (*e.g.*, “Teddy bear skating in Time Square”), outperforming other generative models like VAE [235] and GAN [336]. This implies that DM effectively captures the modularity of hidden attributes [234]. However, as DM’s formulation has no explicit encoders that transform samples into feature vectors, the conventional encoder-decoder feature learning paradigm via reconstruction is no longer applicable [337–339].

Can we extract DM’s knowledge about the modular attributes as a compact feature vector? To answer the question, we start by revealing a natural connection between the diffusion time-steps and the hidden attributes. In the forward diffusion process, given an initial sample $\mathbf{x}_0$, a small amount of Gaussian noise $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is incrementally added to the sample across T time-steps, resulting in a sequence of noisy samples $\mathbf{x}_1, \dots, \mathbf{x}_T$. In particular, each $\mathbf{x}_t$ adheres to its noisy sample distribution $q(\mathbf{x}_t|\mathbf{x}_0)$, whose mean and covariance matrix $\sigma^2\mathbf{I}$ is denoted as a dot and a circle with radius σ in Figure 5.11a, respectively. As t increases, $q(\mathbf{x}_t|\mathbf{x}_0)$ progressively collapses towards a pure noise by reducing mean (*i.e.*, dots moving closer to the origin), and increasing σ (*i.e.*, larger radii), which enlarges the overlapping area (OVL) of the probability density function of noisy sample distributions. In particular, we theoretically show in Section 5.3.3 that OVL is correlated with

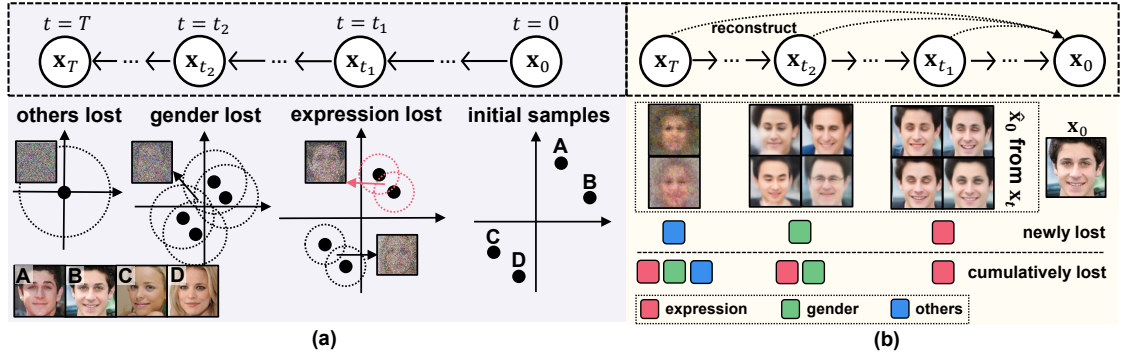


FIGURE 5.11: (a) Illustration of attribute loss as time-step t increases in the forward diffusion process. The two axes depict a two-dimensional sample space. (b) DM reconstructed $\mathbf{x}_0$, denoted as $\hat{\mathbf{x}}_0$, from randomly sampled $\mathbf{x}_t$ at various time-steps t . DM is pre-trained on CelebA, from where $\mathbf{x}_0$ is drawn.

DM’s ambiguous reconstruction of different samples, *e.g.*, due to the large overlap between the red circles at t_1 , noisy samples drawn from the two distributions are reconstructed ambiguously as either A or B, losing the attribute “expression”² that distinguishes them. As t increases, more noisy sample distributions exhibit significant overlaps, inducing additional attribute loss, *e.g.*, further losing “gender” at t_2 that confuses the identities of A/B and C/D. Eventually, with a large enough T , $\mathbf{x}_T$ becomes pure noise and all attributes are lost—the reconstruction can be any sample.

This intrinsic connection between diffusion time-steps and modular attributes can act as an effective inductive bias for unsupervised learning. Note that DM training can be understood as learning to reconstruct $\mathbf{x}_0$ given $\mathbf{x}_t$ at each time-step t (Section 5.3.2). However, due to the aforementioned attribute loss, perfect DM reconstruction is impossible, *e.g.*, in Figure 5.11b, reconstructions exhibit variations in the cumulatively lost attributes. Hence, by contraposition, if a feature enables perfect reconstruction, it must capture the cumulatively lost attributes. This motivates us to learn an *expanding* set of features to supplement the *expanding* lost attributes as t increases. Specifically, we first map $\mathbf{x}_0$ to a feature $\mathbf{z} = f(\mathbf{x}_0)$ using an encoder f . Then we partition $\mathbf{z}$ into T disjoint subsets $\{\mathbf{z}_i\}_{i=1}^T$. At each t , we optimize f so that $\{\mathbf{z}_i\}_{i=1}^t$ compensates for a pre-trained DM’s reconstruction error. Intuitively by induction, starting from $t = 0$ with no lost attribute; if $\{\mathbf{z}_i\}_{i=1}^t$ captures the cumulatively lost attributes till t , $\mathbf{z}_{t+1}$ must capture the newly lost

²Attribute name is illustrative. Our method uses no attribute supervision, nor explicitly names them.

attribute to enable perfect reconstruction at $t + 1$; until $t = T$, $\{\mathbf{z}_i\}_{i=1}^{T-1} \cup \mathbf{z}_T$ learns all hidden attributes in a compact and modular vector space.

We highlight that the analysis above also exposes the deficiency of previous works [21, 190], which learn a time-step-independent feature to supplement DM reconstruction error. Without explicit design to enforce modularity, the learned feature entangles all the attributes. At time-step t , it leaks information about attributes lost at a later time-step (*i.e.*, $t + 1, \dots, T$). Hence, instead of learning the lost attribute $\mathbf{z}_t$, the reconstruction may digress to use the spurious correlation between leaked information and lost attribute. In Section 5.3.5, we validate this deficiency by observing failed counterfactual generation and lower attribute prediction accuracy.

We summarize the paper structure and our contributions below:

- In Section 5.3.2, we provide a preliminary introduction to DM.
- In Section 5.3.3, we build a theoretical framework that connects diffusion time-steps and the hidden modular attributes, which motivates a simple and practical approach for unsupervised disentangled representation learning, discussed in Section 5.3.4.
- In Section 5.3.5, our approach outperforms baselines (PDAE [21], SimSiam [340] and SimCLR [341]) significantly in attribute classification and regression, and enables counterfactual generation.

5.3.2 Problem Formulations

Denoising Diffusion Probabilistic Model [342] (DM) is a latent variable model with a fixed *forward process* (encoding), and a learnable *reverse process* (decoding). Details are given below.

Forward Process. It is fixed to a Markov chain that incrementally adds Gaussian noise to each sample $\mathbf{x}_0$ in T time-steps, producing a sequence of noisy samples $\mathbf{x}_1, \dots, \mathbf{x}_T$ as latents of the same dimensionality as the original sample $\mathbf{x}_0$. Given a variance schedule $\beta_1, \dots, \beta_T$ (*i.e.*, how much noise is added at each time-step), the process is given by:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}). \quad (5.12)$$

In particular, the forward process enjoys a convenient property that allows the direct sampling of $\mathbf{x}_t$ from $\mathbf{x}_0$ at arbitrary time-step t using the noisy sample distribution:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I}), \quad \text{where } \alpha_t := 1 - \beta_t, \bar{\alpha}_t := \prod_{s=1}^t \alpha_s. \quad (5.13)$$

Reverse Process. It corresponds to another Markov chain with learned Gaussian transitions $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ parameterized by θ (discussed below), starting at $p(\mathbf{x}_T) := \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$. Each $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is computed in two steps: 1) Reconstruct $\mathbf{x}_0$ from $\mathbf{x}_t$ with $u_\theta(\mathbf{x}_t, t)$, where u_θ is a learnable U-Net [343]. 2) Compute $q(\mathbf{x}_{t-1}|\mathbf{x}_t, u_\theta(\mathbf{x}_t, t))$, which has a closed-form solution given in Section 5.3.8. Training is performed by minimizing the reconstruction error made by $u_\theta(\mathbf{x}_t, t)$ at each time-step t :

$$\mathcal{L}_{DM} = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \|\mathbf{x}_0 - u_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\|^2, \quad (5.14)$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a random noise. Note that we formulate the U-Net to predict $\mathbf{x}_0$ instead of ϵ as in DDPM [342]. As DM has no explicit feature encoder, we learn one to capture the modular attributes by leveraging the inductive bias in the diffusion time-steps.

5.3.3 Inductive Bias of Diffusion Time-steps

We reveal that in the forward diffusion process, fine-grained attributes, whose alteration leads to fewer pixel-level visual changes, are lost at an earlier time-step compared to coarse-grained ones. Recall that when DM fails to distinguish the noisy samples drawn from two overlapping noisy sample distributions $q(\mathbf{x}_t|\mathbf{x}_0)$ and $q(\mathbf{y}_t|\mathbf{y}_0)$, the attribute that distinguishes them becomes lost. We formalize it below:

Definition. (Attribute Loss) *Given a θ -parameterized DM optimized by Eq. (5.14), we say that attribute $\mathcal{Z}_i$ is lost with degree τ at t when $\mathbb{E}_{\mathbf{x}_0 \in \mathcal{X}} [\text{Err}(\mathbf{x}_0, \mathbf{y}_0 = g_i \cdot \mathbf{x}_0, t)] \geq \tau$, where*

$$\text{Err}(\mathbf{x}_0, \mathbf{y}_0, t) := \frac{1}{2} \left[\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \mathbb{1}(\|\hat{\mathbf{x}}_0 - \mathbf{x}_0\| > \|\hat{\mathbf{x}}_0 - \mathbf{y}_0\|) + \mathbb{E}_{q(\mathbf{y}_t|\mathbf{y}_0)} \mathbb{1}(\|\hat{\mathbf{y}}_0 - \mathbf{x}_0\| < \|\hat{\mathbf{y}}_0 - \mathbf{y}_0\|) \right], \quad (5.15)$$

with $\mathbb{1}(\cdot)$ denoting the indicator function and $\hat{\mathbf{x}}_0 = u_\theta(\mathbf{x}_t)$, $\hat{\mathbf{y}}_0 = u_\theta(\mathbf{y}_t)$ denoting the DM reconstructed samples from $\mathbf{x}_t, \mathbf{y}_t$, respectively.

Intuitively, $\text{Err}(\mathbf{x}_0, \mathbf{y}_0, t)$ measures attribute loss degree by computing how likely a DM falsely reconstructs $\mathbf{x}_t$ drawn from $q(\mathbf{x}_t|\mathbf{x}_0)$ closer to $\mathbf{y}_0$ instead of $\mathbf{x}_0$ (vice versa). Hence, when $\mathbf{x}_0$ and $\mathbf{y}_0$ differ in attribute $\mathbf{z}_i$ modified by g_i , the increased $\text{Err}(\mathbf{x}_0, \mathbf{y}_0, t)$ means that the attribute is more likely lost.

Theorem. (Attribute Loss and Time-step) 1) For each $\mathcal{Z}_i$, there exists a smallest time-step $t(\mathcal{Z}_i)$, such that $\mathcal{Z}_i$ is lost with degree τ at each $t \in \{t(\mathcal{Z}_i), \dots, T\}$. 2) $\exists \{\beta_i\}_{i=1}^T$ such that $t(\mathcal{Z}_i) > t(\mathcal{Z}_j)$ whenever $\|\mathbf{x}_0 - g_i \cdot \mathbf{x}_0\|$ is first-order stochastic dominant over $\|\mathbf{x}_0 - g_j \cdot \mathbf{x}_0\|$ with $\mathbf{x}_0 \sim \mathcal{X}$ uniformly.

Intuitively, the first part of the theorem states that a lost attribute will not be regained as time-step t increases, and there is a time-step $t(\mathcal{Z}_i)$ when $\mathcal{Z}_i$ becomes lost for the first time. The second part states that when $\|\mathbf{x}_0 - g_i \cdot \mathbf{x}_0\|$ is more likely to take on larger value than $\|\mathbf{x}_0 - g_j \cdot \mathbf{x}_0\|$ (i.e., first-order stochastic dominant), the attribute $\mathcal{Z}_i$ (corresponding to a coarse-grained attribute) is lost at a larger time-step compared to $\mathcal{Z}_j$ (corresponding to a fine-grained attribute). We have the following proof sketch:

1) *Increasing t induces further attribute loss.* We show in Section 5.3.8 that $\text{Err}(\mathbf{x}_0, \mathbf{y}_0, t) = \frac{1}{2} \text{OVL}(q(\mathbf{x}_t|\mathbf{x}_0), q(\mathbf{y}_t|\mathbf{y}_0))$ for an optimal DM, where OVL is the overlapping coefficient [344], i.e., the overlapping area of probability density functions (PDFs). In particular, we have

$$\text{Err}(\mathbf{x}_0, \mathbf{y}_0, t) = \frac{1}{2} \text{OVL}(q(\mathbf{x}_t|\mathbf{x}_0), q(\mathbf{y}_t|\mathbf{y}_0)) = \frac{1}{2} \left[1 - \text{erf} \left(\frac{\|\sqrt{\bar{\alpha}_t}(\mathbf{y}_0 - \mathbf{x}_0)\|}{2\sqrt{2(1 - \bar{\alpha}_t)}} \right) \right]. \quad (5.16)$$

As $\bar{\alpha}_t$ decreases with an increasing t from Eq. (5.13), and the error function $\text{erf}(\cdot)$ is strictly increasing, $\text{Err}(\mathbf{x}_0, \mathbf{y}_0, t)$ is strictly increasing in t given any $\mathbf{x}_0, \mathbf{y}_0$. Hence $\mathbb{E}_{\mathbf{x}_0 \in \mathcal{X}} [\text{Err}(\mathbf{x}_0, \mathbf{y}_0 = g_i \cdot \mathbf{x}_0, t)] \geq \mathbb{E}_{\mathbf{x}_0 \in \mathcal{X}} [\text{Err}(\mathbf{x}_0, \mathbf{y}_0 = g_i \cdot \mathbf{x}_0, t(\mathcal{Z}_i))]$ for every $t \geq t(\mathcal{Z}_i)$, which completes the proof.

2) *Coarse-grained attributes are lost at a larger t .* Given that $\text{erf}(\cdot)$ is strictly increasing and $\|\mathbf{x}_0 - g_i \cdot \mathbf{x}_0\|$ is first-order stochastic dominant over $\|\mathbf{x}_0 - g_j \cdot \mathbf{x}_0\|$, we have $\mathbb{E}_{\mathbf{x}_0 \in \mathcal{X}} [\text{Err}(\mathbf{x}_0, g_i \cdot \mathbf{x}_0, t)] > \mathbb{E}_{\mathbf{x}_0 \in \mathcal{X}} [\text{Err}(\mathbf{x}_0, g_j \cdot \mathbf{x}_0, t)]$ at every time-step t using Eq. (5.16). Hence $t(\mathcal{Z}_i) > t(\mathcal{Z}_j)$ under any variance schedule $\{\beta_i\}_{i=1}^T$ such that $\mathcal{Z}_i$

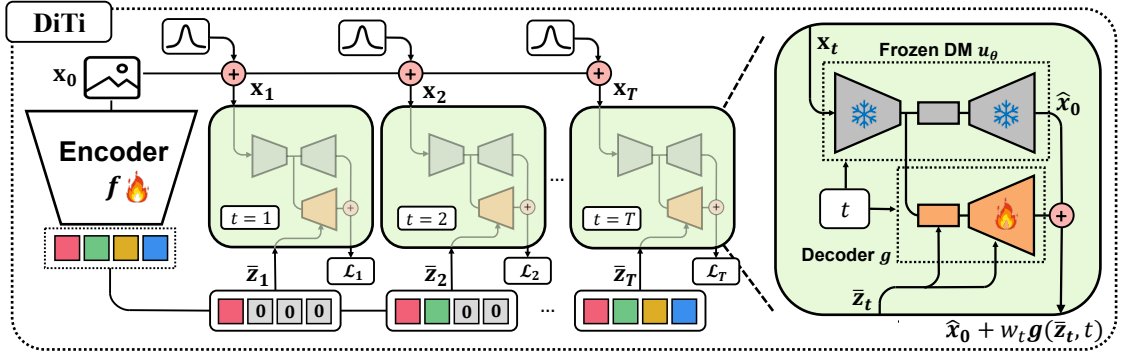


FIGURE 5.12: Illustration of our DiTi. On the left, we break down the learning with Eq. (5.17) at each time-step, and the loss at t is denoted as $\mathcal{L}_t$. On the right, we show the detailed network design. We follow PDAE [21] for the decoder design. $\hat{\mathbf{x}}_0$ denotes the reconstructed $\mathbf{x}_0$ by the pre-trained DM.

is *not* lost at $t(\mathcal{Z}_j)$, completing the proof. Note that in practice, DM leverages a large T with small $\{\beta_i\}_{i=1}^T$. This ensures that $\text{Err}(\mathbf{x}_0, \mathbf{y}_0, t)$ slowly increases with t according to Eq. (5.16). Hence empirically, this theorem tends to hold.

5.3.4 Proposed Approach

The previous theoretical analysis leads to two interesting groundings: 1) Attribute loss is linked with the error of reconstructing $\mathbf{x}_0$ from the noisy $\mathbf{x}_t$. 2) Increasing time-step t in forward diffusion process causes an expanding cumulative loss of attributes, with fine-grained attributes lost at a smaller t and coarse-grained ones lost at a larger t . They serve as an effective inductive bias for unsupervised learning: 1) making up for the reconstruction error can retrieve lost attributes, 2) an expanding set of features captures the expanding cumulatively lost attributes. This motivates our approach below.

As illustrated in Figure 5.12, we term our approach as **DiTi** to highlight our leverage of Diffusion Time-step as an inductive bias for feature learning. Our model includes a frozen DM U-Net u_θ pre-trained on $\mathcal{D}$ using Eq. (5.14) until convergence, a trainable encoder f (trained to become a disentangled representation) that maps each image (without noise) to a d -dimensional feature $\mathbf{z} \in \mathbb{R}^d$, and a trainable decoder g that maps $\mathbf{z}$ back to image space given t . In training, given $\mathbf{z} = f(\mathbf{x}_0)$, we partition it into T disjoint subsets $\{\mathbf{z}_i\}_{i=1}^T$. At each time-step t , we construct a new feature $\bar{\mathbf{z}}_t = [\mathbf{z}_1, \dots, \mathbf{z}_t, \mathbf{0}, \dots, \mathbf{0}]$, which is only about $\{\mathbf{z}_i\}_{i=1}^t$ by masking

$\mathbf{z}_{t+1}, \dots, \mathbf{z}_T$ as $\mathbf{0}$. Our training loss requires $\bar{\mathbf{z}}_t$ to compensate for the reconstruction error made by u_θ at each time-step t , given by:

$$\mathcal{L} = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \underbrace{\lambda_t \|\mathbf{x}_0 - (u_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) + w_t g(\bar{\mathbf{z}}_t, t))\|^2}_{\mathcal{L}_t}, \quad (5.17)$$

where λ_t, w_t are time-step weight and compensate strength, which are fixed values computed from the variance schedule $\{\beta_t\}_{t=1}^T$, and we follow the settings in PDAE [21].

To see the effect of Eq. (5.17), consider $t = 1$, only $\mathbf{z}_1$ is used to compensate for reconstruction error, hence $\mathbf{z}_1$ effectively captures a (fine-grained) modular attribute $\mathcal{Z}_i$ with $t(\mathcal{Z}_i) = 1$ (lost at $t = 1$) (such $\mathcal{Z}_i$ always exists for some threshold τ in our definition). At $t = 2$, the cumulatively lost attributes correspond to $\mathcal{Z}_i$ and $\mathcal{Z}_j$, where $t(\mathcal{Z}_j) = 2$, and because $\mathbf{z}_1$ already captures the attribute $\mathcal{Z}_i$, $\mathbf{z}_2$ is encouraged to further learn the attribute $\mathcal{Z}_j$ to minimize the reconstruction error at $t = 2$. Eventually with a large enough T , $\mathbf{x}_T$ becomes pure noise and all attributes are lost. By the above induction, $\{\mathbf{z}_i\}_{i=1}^T$ captures all the hidden attributes as a modular feature. In contrast, the most related work PDAE differs from Eq. (5.17) by replacing $g(\bar{\mathbf{z}}_t)$ with $g(\bar{\mathbf{z}})$. This means that PDAE uses the full-dimensional feature $\mathbf{z}$ to compensate for the reconstruction error at all time-steps, which fails to leverage the inductive bias discussed in Section 5.3.3. We show in Section 5.3.5 that this simple change leads to drastic differences in both classification and generation tasks.

Implementation Details. We highlight some design considerations with ablations in Section 5.3.6.

- Number of Subsets k . In practice, the number of subsets for partitioning $\mathbf{z}$ can be smaller than T , *e.g.*, we use feature dimension $d = 512$ that is smaller than $T = 1,000$. This means that adjacent time-steps may share a modular feature $\mathbf{z}_i$. We compare the choice of k later.
- Partition Strategy. The simplest way is to use the same number of feature dimensions for all modular features $\mathbf{z}_i$ (*i.e.*, balanced). However, we empirically find that features corresponding to time-step 100-300 require more dimensions to improve convergence of Eq. (5.17). We design an imbalanced partition strategy and compare it with the balanced one.

- **Optimization Strategy.** In training, we experiment detaching the gradient of $\mathbf{z}_1, \dots, \mathbf{z}_{t-1}$ to form $\bar{\mathbf{z}}_t$, *i.e.*, training only $\mathbf{z}_t$ at time-step t . We find that it leads to improved disentanglement quality at the cost of slower convergence.

5.3.5 Experiments

Datasets. We adopted 3 training datasets: 1) Celebrity Faces Attributes (CelebA) [345] is a large-scale face attributes dataset. It is split into 162,770 images for training, 19,867 images for validation, and 19,962 images for testing. CelebA has 40 attribute labels per image, including various facial features and characteristics such as smiling, wearing glasses, gender, etc. 2) Flickr-Faces-HQ (FFHQ) [315] dataset is a high-quality image dataset of human faces obtained from Flickr. It contains 70,000 high-quality face images. 3) Bedroom is part of the Large-scale Scene Understanding (LSUN) dataset [346] that contains 3 million images, depicting bedrooms in a wide variety. We additionally used the Labeled Faces in the Wild (LFW) dataset [347] to evaluate attribute regression.

Training Protocols. Our model training followed the encoder and decoder design of PDAE and utilized its pre-trained DM. We trained the model for 290k iterations on CelebA, 500k iterations on FFHQ, and 540k iterations on Bedroom, with a batch size of 128 consistent with PDAE. CelebA images were resized to 64×64 while FFHQ and Bedroom images were resized to 128×128 . We use the Adam optimizer with a learning rate $1e^{-4}$. To train our DiTi model, we employed four NVIDIA A100 40GB GPUs with batch size = 32 on each gpu. Training on the FFHQ and Bedroom datasets generally requires approximately 5 days, while training on CelebA takes around 1 day.

5.3.6 Inference Results

	Methods	AP $\uparrow$	Pearson's r $\uparrow$	MSE $\downarrow$
CelebA	PDAE [21]	60.2	0.596	0.410
	SimCLR [341]	59.7	0.474	0.603
	SimCLR–Aug [341]	34.7	0.176	0.717
	SimSiam [340]	51.7	0.464	0.525
	DiTi (Ours)	62.3	0.617	0.392
FFHQ	PDAE [21]	59.7	0.603	0.416
	SimCLR [341]	60.8	0.481	0.638
	DiTi (Ours)	61.4	0.622	0.384

TABLE 5.5: AP (%) on CelebA attribute classification and Pearson’s r, MSE on LFW attribute regression. The first column shows the training dataset.

Imbalance	Detach	k	AP $\uparrow$	Pearson's r $\uparrow$	MSE $\downarrow$
✓		16	62.1	0.619	0.389
✓		32	62.0	0.615	0.392
✓		64	62.3	0.617	0.392
✓		128	61.9	0.616	0.391
		64	61.9	0.604	0.410
	✓	64	62.5	0.590	0.422
✓	✓	64	62.4	0.604	0.405

TABLE 5.6: Ablations on k , partition and optimization strategy. Models were trained on CelebA. *Imbalance* and *Detach* means using our devised partition and optimization strategy.

Evaluation Methods. We took only the learned encoder f on CelebA [345] and FFHQ [315] and evaluated its output feature on attribute classification and regression. For classification, we used the 40 attributes from CelebA. We trained a linear classifier to predict the 40 attributes simultaneously (*i.e.*, multi-task classification), and evaluated it using Average Precision (AP) as the metric. For regression, we trained a linear regression model on 65 attributes from LFW (labeled with real numbers), and the evaluation metric is Pearson’s r and Mean Squared Error (MSE). We choose 3 baselines: 1) PDAE [21] as a representative approach that leverages DM to learn representation. 2) SimCLR [341] as a typical contrastive learning method. 3) SimSiam [340] as a self-supervised learning method without using contrastive negative pairs.

Main Results. As shown in Table 5.5, our proposed DiTi significantly outperforms all baselines regardless of task and training dataset. We have the following observations: 1) We improve PDAE by a large margin, *e.g.*, +2.1% and +1.7% on CelebA AP with the two training datasets, respectively. We further break down the improved precision on each attribute in Figure 5.13 top, which shows consistent improvements on all attributes. The result strongly validates our approach to disentangle modular attributes, and further supports our analysis on PDAE’s deficiency, *i.e.*, it may digress to use spurious correlation to supplement reconstruction error, *e.g.*, the most improved attribute of DiTi over PDAE is “Brown Hair”, which strongly correlated with the female gender. 2) SimCLR does not perform well in attribute classification or regression, surpassed even by PDAE.

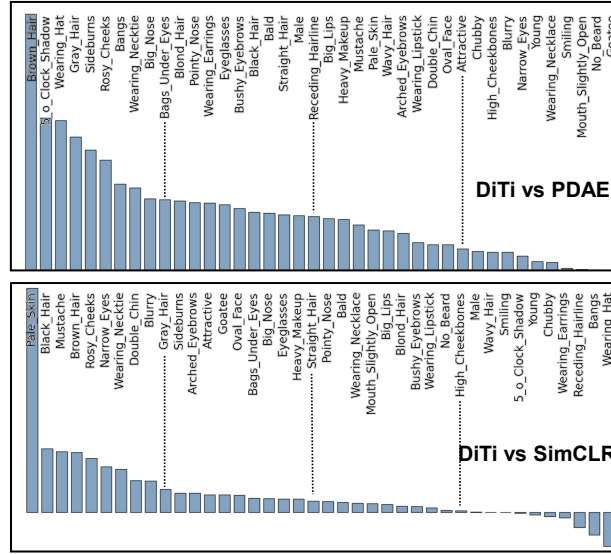


Figure 5.13: Improvements in attribute classification precision of our DiTi over PDAE (top) and SimCLR (bottom).

This is because SimCLR learns an augmentation-invariant feature, which explicitly discards many augmentation-related attributes, *e.g.*, hair or skin color. This is verified by how much DiTi improves “Pale Skin” precision over SimCLR in Figure 5.13 bottom.

This deficiency of SimCLR cannot be addressed by simply removing augmentations linked with attributes (SimCLR–Aug in Table 5.5) due to the reliance of contrastive learning on augmentations [348]. Note that SimCLR’s contrastive objective has some effect in regularizing f as an injective mapping, leading to slightly improved precision on some accessory-related attributes (*e.g.*, “Wearing Hat”). 3) In the absence of the beneficial regularizing effect from the negative pairs, SimSiam performs even worse than SimCLR. 4) The models trained on FFHQ generally perform slightly worse compared with those trained on CelebA, despite using an increased image size (128 vs 64). This is largely because CelebA more than doubles FFHQ in the training image number.

Ablation on #Subsets k . From Table 5.6 first 4 lines, we observe that $k = 64$ achieves the best performance, *i.e.*, dividing total time-steps T and feature dimensions d into 64 segments. With a smaller k , we may not be able to capture all hidden attributes. However, when k becomes too large, the dimension of each $\mathbf{z}_i$ can be extremely small (*e.g.*, 4 when $k = 128$ in our case), which may not be expressive enough to capture the hidden attribute. Nonetheless, DiTi is not sensitive to the choice of k overall, and we choose the best performing $k = 64$.

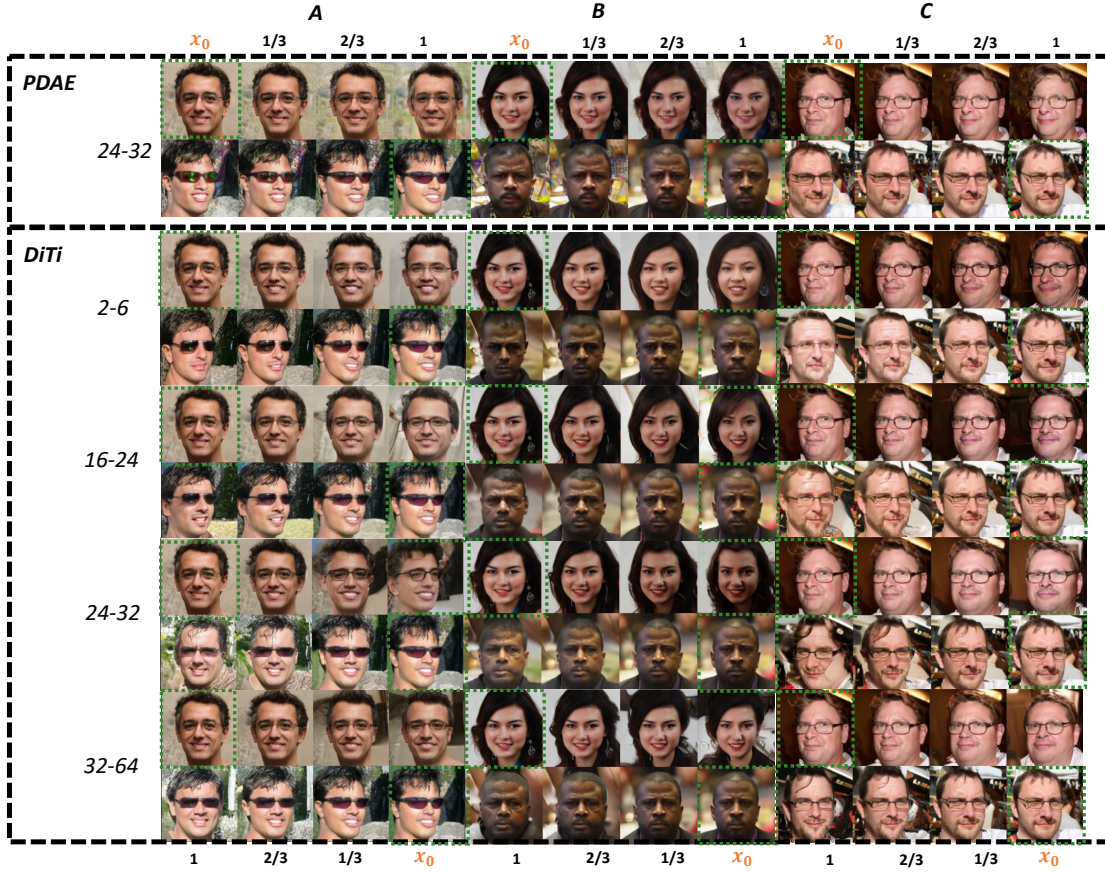


FIGURE 5.14: Counterfactual generations on FFHQ. $\mathbf{x}_0$ and $\mathbf{x}'_0$ are marked with a green box.

Ablation on Partition & Optimization Strategy. For partition strategy, we empirically observe that our training loss $\mathcal{L}_t$ with t in 100-300 contributes the most to the overall loss. Hence we devise an imbalanced partition strategy to allocate feature dimensions, *i.e.*, using increased dimensions for each $\mathbf{z}_i$ corresponding to time-step 100-300. We observe this technique leads to improved loss convergence and improved performance in Table 5.6, *e.g.*, 62.3% AP, 0.617 Pearson’s r in line 3 vs 61.9% AP, 0.604 Pearson’s r in line 5. Regarding optimization strategy, we tried detaching the gradient of $\mathbf{z}_1, \dots, \mathbf{z}_{t-1}$ at time-step t to prevent them from capturing the lost attribute at t . This technique leads to slower convergence, as each $\mathbf{z}_i$ is only trained $\frac{1}{k}$ iterations, which especially hurts the hard task of attribute regression. However, we do observe improved disentanglement quality in counterfactual generation task.

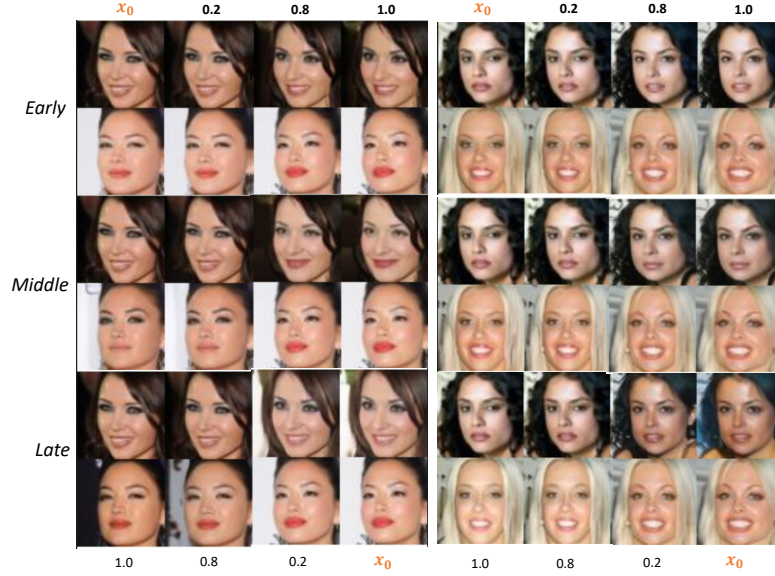


Figure 5.15: Counterfactual generation on CelebA.

5.3.7 Counterfactual Generation

Evaluation Methods. We aim to generate counterfactual images by interpolating a *subset* $\mathcal{S} \subset \{1, \dots, k\}$ of the k modular attributes. For each pair of images $\mathbf{x}_0, \mathbf{x}'_0$, we perform DDIM inversion [187] using the pre-trained DM to obtain the DDIM-inversed $\mathbf{x}_T, \mathbf{x}'_T$, and encode $\mathbf{x}_0, \mathbf{x}'_0$ by f into the modular attributes $\mathbf{z} = \{\mathbf{z}_i\}_{i=1}^k, \mathbf{z}' = \{\mathbf{z}'_i\}_{i=1}^k$. To generate counterfactuals for $\mathbf{x}_0$, we start from the noisy sample $\text{Slerp}(\mathbf{x}_T, \mathbf{x}'_T; \lambda)$ (*i.e.*, spherical linear interpolation [334] of $\mathbf{x}_T$ into $\mathbf{x}'_T$ with scale λ). Then we run DDIM sampling under the guidance of $g(\text{lerp}(\mathbf{z}, \mathbf{z}', \mathcal{S}; \lambda), t)$, where we *only* perform linear interpolation (lerp) with scale λ on the subset $\mathcal{S}$ of $\mathbf{z}$, *without* altering the value of other modular attributes. Generating counterfactuals for $\mathbf{x}'_0$ follows a similar process.

FFHQ Results.

In Figure 5.14, we compare the counterfactual generations by PDAE and our DiTi on 3 image pairs (*i.e.*, A, B, C) with 4 different subsets (*i.e.*, 2-6, 16-24, 24-32, 32-64). For each pair with a specific subset $\mathcal{S}$, the top left image is $\mathbf{x}_0$ and the bottom right image is $\mathbf{x}'_0$. From $\mathbf{x}_0$, we perform counterfactual generation by interpolating with $\mathbf{x}'_0$ with 3 scales (*i.e.*, 1/3, 2/3, 1) with result listed from left to right. We do the same for $\mathbf{x}'_0$ and list the results from right to left. Note that a scale of 1 means the feature values corresponding to $\mathcal{S}$ are replaced with those of the other sample.

For PDAE, we only show the results on the subset 24-32 and leave full results in Section 5.3.8. This is because regardless of the chosen subset $\mathcal{S}$, PDAE always generates meaningless counterfactuals. For example, we observe significant artifacts on pair A, and meaningless edits on pair A by the subset 24-32. This means that the interpolated feature on the subset $\text{lerp}(\mathbf{z}, \mathbf{z}', \mathcal{S}; \lambda)$ loses its semantic meaning. With the theoretical results that link disentangled representation with the capability to generate faithful counterfactuals (*i.e.*, lying on image data manifold) [44], PDAE’s failure further validates that it entangles the attributes without explicit design to enforce modularity.

In contrast, our DiTi generates much more realistic (*i.e.*, faithful) counterfactuals on all pairs, subsets and interpolation scales. In particular, we observe that a subset with larger indices (*e.g.*, 32-64) corresponds to more coarse-grained attributes than that with smaller indices (*e.g.*, 2-6). We observe that subset 2-6 corresponds to fine-grained edit such as mouth and eyebrows (*e.g.*, arched eyebrows and smiling for pair A), 16-24 for chin and face shape (eg, double chin for pair C), 24-32 for larger edit such as head pose (*e.g.*, facing direction for pair A, C), and 32-64 for overall edit such as hairstyle (*e.g.*, receding hairline for C).

This strongly validates that our DiTi learns a modular feature, and supports our theoretical analysis in Section 5.3.3 (*i.e.*, coarse-grained attributes are lost later and captured by $\mathbf{z}_i$ with a large i).

CelebA Results. On CelebA, we used 3 subsets (early→late corresponding to one with smaller→ larger indices) due to space constraints. As shown in figure 5.15, they correspond to the attribute “eye direction”, “eye shadows”, and “backgrounds”, respectively.

Ablation on Evaluation Methods. Recall that in previous counterfactual generation, the starting noisy samples for subsequent DDIM sampling is obtained by spherical interpolation between the DDIM-inversed $\mathbf{x}_T, \mathbf{x}'_T$. In Figure 5.16, we compare this approach (DDIM-inverse $\mathbf{x}_T$) with an alternative one by randomly sampling from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ (random $\mathbf{x}_T$) on 3 subsets (*i.e.*, early, middle, late). Note that each counterfactual is generated with an interpolation scale of 1.0.

For PDAE, without attribute information provided by the DDIM-inversed $\mathbf{x}_T$, it generates counterfactuals that are drastically different from the original samples, suggesting that many attributes are not faithfully captured by the PDAE feature.

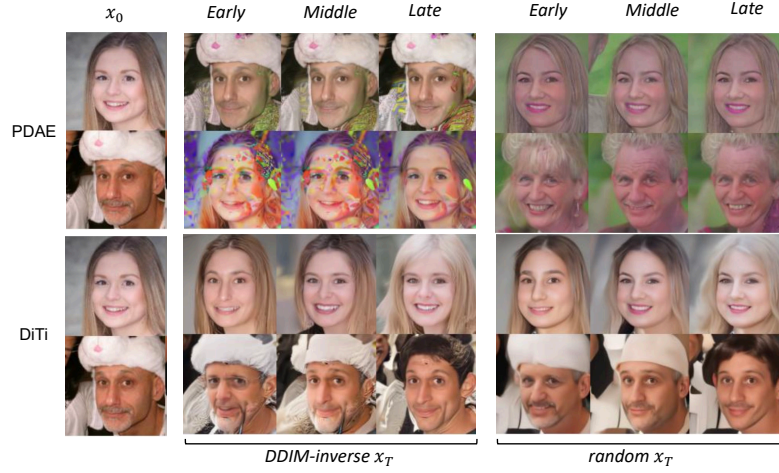


Figure 5.16: Ablation on evaluation methods.

In contrast, our DiTi is still able to generate realistic counterfactuals by interpolating the corresponding attributes (*e.g.*, smile for the early subset, hairstyle for the late subset) and retaining other attributes (*e.g.*, background). This further validates the effectiveness of our approach.

5.3.8 Supplementary Information

Closed Form of $q(\mathbf{x}_{t-1}|\mathbf{x}_t, u_\theta(\mathbf{x}_t, t))$. Given by $\mathcal{N}(\mathbf{x}_{t-1}|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I})$, where

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}\mathbf{x}_t, \quad \tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\beta_t. \quad (5.18)$$

Equivalent Formulation. The simplified objective in DDPM [342] is given by:

$$\mathcal{L}_{DM} = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\|^2, \quad (5.19)$$

where ϵ_θ is a θ -parameterized U-Net [343] to predict the added noise. From Eq. (2), we have

$$\epsilon = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1 - \bar{\alpha}_t}}, \quad \epsilon_\theta = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}u_\theta}{\sqrt{1 - \bar{\alpha}_t}}, \quad (5.20)$$

where we slightly abuse the notation to denote the reconstructed $\mathbf{x}_0$ from our U-Net as u_θ . Taking Eq. (5.20) into Eq. (5.19) yields Eq. (3).

Time-step Weight λ_t and Compensate Strength w_t . The PDAE objective is given by

$$\mathcal{L}_{PDAE} = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} [\lambda_t^p \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) + w_t^p g(\mathbf{z}, t)\|^2], \quad (5.21)$$

where

$$\lambda_t^p = \left(\frac{1}{1 + \text{SNR}(t)} \right)^{0.9} \left(\frac{\text{SNR}(t)}{1 + \text{SNR}(t)} \right)^{0.1}, \text{SNR}(t) = \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t}, w_t^p = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{\sqrt{1 - \bar{\alpha}_t}}. \quad (5.22)$$

Taking Eq. (5.20) into Eq. (5.21) yields:

$$\lambda_t = \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \lambda_t^p, \quad w_t = \frac{\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} w_t^p = \sqrt{\frac{\alpha_t}{\bar{\alpha}_t}} (1 - \bar{\alpha}_t). \quad (5.23)$$

Full Proof to the Theorem. We list the theorem below for reference.

Theorem. (Attribute Loss and Time-step) 1) For each $\mathcal{Z}_i$, there exists a smallest time-step $t(\mathcal{Z}_i)$, such that $\mathcal{Z}_i$ is lost with degree τ at each $t \in \{t(\mathcal{Z}_i), \dots, T\}$. 2) $\exists \{\beta_i\}_{i=1}^T$ such that $t(\mathcal{Z}_i) > t(\mathcal{Z}_j)$ whenever $\|\mathbf{x}_0 - g_i \cdot \mathbf{x}_0\|$ is first-order stochastic dominant over $\|\mathbf{x}_0 - g_j \cdot \mathbf{x}_0\|$ with $\mathbf{x}_0 \sim \mathcal{X}$ uniformly.

Proof. We start by showing $\text{Err}(\mathbf{x}_0, \mathbf{y}_0, t) = \frac{1}{2} \text{OVL}(q(\mathbf{x}_t|\mathbf{x}_0), q(\mathbf{y}_t|\mathbf{y}_0))$. Without loss of generality, we show a 1-D sample space $\mathcal{X}$ in Figure 5.17. The minimum Err_θ is obtained when given each noisy sample $\mathbf{x}$, DM reconstructs towards $\mathbf{x}_0$ if $q(\mathbf{x}|\mathbf{x}_0) > q(\mathbf{x}|\mathbf{y}_0)$ and vice versa for $\mathbf{y}_0$, e.g., reconstructing $\star$ as $\mathbf{y}_0$. However, this maximum likelihood estimation fails when a noisy sample is drawn from $q(\mathbf{x}_t|\mathbf{x}_0)$ (green PDF), but with a value larger than the intersection point of the two PDFs ($\blacklozenge$), and similar arguments go for $q(\mathbf{y}_t|\mathbf{y}_0)$ (blue PDF). The error rate caused by the two failure cases corresponds to the green shaded area and blue one, respectively, leading to an average Err_θ of $\frac{1}{2}$ of the OVL.

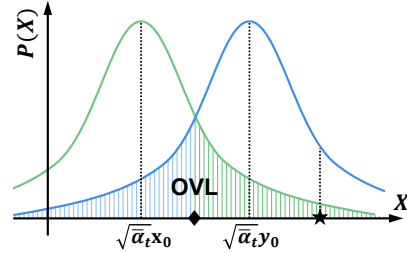


Figure 5.17: The PDFs of $q(\mathbf{x}_t|\mathbf{x}_0)$ (green) and $q(\mathbf{y}_t|\mathbf{y}_0)$ (blue). Without loss of generality, we consider the 1-D case of $y_0 > x_0$. Their means are computed from Eq. (2).

To compute the OVL, it is trivial in the 1-D case by leveraging the Cumulative Distribution Function (CDF) of Gaussian distribution. Given that the two distributions have an equal variance from Eq. (2), the intersection point is given by

$\frac{\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{\bar{\alpha}_t}\mathbf{y}_0}{2}$. For a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, its CDF is given by $\frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma}\right) \right]$. Combining two results, one can easily show that the blue shaded area, corresponding to half of the OVL, or $\operatorname{Err}(x_0, y_0, t)$, is given by:

$$\operatorname{Err}(x_0, y_0, t) = \frac{1}{2} \operatorname{OVL}(q(x_t|x_0), q(y_t|y_0)) = \frac{1}{2} \left[1 - \operatorname{erf}\left(\frac{\sqrt{\bar{\alpha}_t}(y_0 - x_0)}{2\sqrt{2(1 - \bar{\alpha}_t)}}\right) \right]. \quad (5.24)$$

To generalize the results to multi-variate Gaussian distributions, we use the results in [349], which shows that by projecting the data to Fisher’s linear discriminant axis, the OVL defined on the discriminant densities is equal to that defined on the multivariate densities. Specifically, the mean of the discriminant densities is given by

$$\mu_0 = \sqrt{\bar{\alpha}_t}(\mathbf{y}_0 - \mathbf{x}_0)^\top \Sigma^{-1} \mathbf{x}_0, \quad \mu_1 = \sqrt{\bar{\alpha}_t}(\mathbf{y}_0 - \mathbf{x}_0)^\top \Sigma^{-1} \mathbf{y}_0, \quad (5.25)$$

where $\Sigma = \beta_t \mathbf{I}$. The common variance of the discriminant densities is given by $\sqrt{\bar{\alpha}_t}(\mathbf{y}_0 - \mathbf{x}_0)^\top \Sigma^{-1}(\mathbf{y}_0 - \mathbf{x}_0)$. Following the calculation steps to compute OVL for the 1-D case, one can show that for both 1-D and multi-variate cases, we have

$$\operatorname{Err}(\mathbf{x}_0, \mathbf{y}_0, t) = \frac{1}{2} \operatorname{OVL}(q(\mathbf{x}_t|\mathbf{x}_0), q(\mathbf{y}_t|\mathbf{y}_0)) = \frac{1}{2} \left[1 - \operatorname{erf}\left(\frac{\|\sqrt{\bar{\alpha}_t}(\mathbf{y}_0 - \mathbf{x}_0)\|}{2\sqrt{2(1 - \bar{\alpha}_t)}}\right) \right]. \quad (5.26)$$

As $\bar{\alpha}_t$ decreases with an increasing t from Eq. (2), and the error function $\operatorname{erf}(\cdot)$ is strictly increasing, $\operatorname{Err}(\mathbf{x}_0, \mathbf{y}_0, t)$ is strictly increasing in t given any $\mathbf{x}_0, \mathbf{y}_0$. Hence $\mathbb{E}_{\mathbf{x}_0 \in \mathcal{X}} [\operatorname{Err}(\mathbf{x}_0, \mathbf{y}_0 = g_i \cdot \mathbf{x}_0, t)] \geq \mathbb{E}_{\mathbf{x}_0 \in \mathcal{X}} [\operatorname{Err}(\mathbf{x}_0, \mathbf{y}_0 = g_i \cdot \mathbf{x}_0, t(\mathcal{Z}_i))]$ for every $t \geq t(\mathcal{Z}_i)$, which completes the proof of Theorem 1.

Given that $\operatorname{erf}(\cdot)$ is strictly increasing and $\|\mathbf{x}_0 - g_i \cdot \mathbf{x}_0\|$ is first-order stochastic dominant over $\|\mathbf{x}_0 - g_j \cdot \mathbf{x}_0\|$, we have $\mathbb{E}_{\mathbf{x}_0 \in \mathcal{X}} [\operatorname{Err}(\mathbf{x}_0, g_i \cdot \mathbf{x}_0, t)] > \mathbb{E}_{\mathbf{x}_0 \in \mathcal{X}} [\operatorname{Err}(\mathbf{x}_0, g_j \cdot \mathbf{x}_0, t)]$ at every time-step t using Eq. (5.26). Hence $t(\mathcal{Z}_i) > t(\mathcal{Z}_j)$ under any variance schedule $\{\beta_i\}_{i=1}^T$ such that $\mathcal{Z}_i$ is *not* lost at $t(\mathcal{Z}_j)$, completing the proof of Theorem 2.

5.4 Conclusion

In Section 5.2, we presented a novel counterfactual framework for Zero-Shot Learning (ZSL) and Open-Set Recognition (OSR) to provide a theoretical ground for balancing and improving the seen/unseen classification imbalance. Specifically, we

proposed a Generative Causal Model to generate faithful counterfactuals, which allows us to use the Consistency Rule for balanced binary seen/unseen classification. Extensive results in ZSL and OSR show that our method indeed improves the balance and hence achieves the state-of-the-art performance. As a future direction, we will seek new definitions on disentanglement [167] and devise practical implementations to achieve improved disentanglement [311].

In Section 5.3, we presented a novel unsupervised disentangled representation learning method by leveraging the inductive bias provided by Denoising Diffusion Probabilistic Model (DM)’s time-step. In particular, we theoretically reveal an inherent connection between time-step and hidden modular attributes that generate data faithfully, *i.e.*, each incremental time-step in the forward diffusion process induces additional attribute loss, causing the failure of DM to reconstruct the original sample from the noisy ones. Hence we learn an expanding set of modular features as the time-step increases to compensate for the cumulatively lost attributes by making up for DM’s reconstruction error. The learned feature supports accurate downstream inference on facial attributes and enables counterfactual generation on CelebA, FFHQ, and Bedroom datasets, validating its disentanglement quality. As future work, we will seek additional inductive bias to improve disentanglement, *e.g.*, using text as a disentangled template by exploring text-to-image diffusion models [334, 335], and devise practical optimization techniques to enable faster convergence.

Chapter 6

Summary

In this thesis, we aim to improve machine generalization when the deployment environment has different distribution from the training one. This has become a pressing need: In sensitive applications such as AI healthcare, a tumor classifier trained in a hospital A can fail in hospital B. The AI fairness problems already raise social concerns, *e.g.*, social media AI labels black men as “primates”. Even worse, autonomous pilot system starts to pose threats to public security, *e.g.*, an autonomous pilot ignores uncommon obstacles to crash. We hope by applying the causal inference principles in machine learning, we can build an AI that mimics human reasoning and pave way for artificial general intelligence [350].

We view machine generalization through the lens of causality, which reveals that the key to achieving generalization is removing the spurious correlation. We then identify three methods that mitigate spurious correlation: **1) backdoor adjustment**, **2) invariant learning**, and **3) disentangled representation learning**. Next, we apply the intervention methods in staple machine generalization tasks: Few-Shot Learning (FSL), Unsupervised Domain Adaptation (UDA), Semi-Supervised Learning (SSL), Zero-Shot Learning (ZSL), and Open-Set Recognition (OSR).

The current paradigm of these tasks is summarized in Figure 6.1. All of them can be broken down into a two-stage training scheme:

1) The first stage aims to learn a powerful representation as prior knowledge, which takes the form of a pre-trained neural network (*i.e.*, backbone). We denote the backbone as a mapping $\mathcal{I} \rightarrow \mathcal{X}$ from the sample space $\mathcal{I}$ to the feature vector

	FSL	UDA/SSL	ZSL/OSR	Unsupervised
1. Prior: Representation Learning	$\mathcal{I} \rightarrow \mathcal{X}$ Pre-trained on base classes	$\mathcal{I} \rightarrow \mathcal{X}$ Pre-trained on ImageNet	$\mathcal{I} \leftrightarrow (\mathcal{Z}, \mathcal{Y})$ $\mathcal{Z}$: sample attribute $\mathcal{Y}$: class attribute	$\mathcal{I} \rightarrow \mathcal{X}$ Pre-trained with pretext task
2. Fine-tuning: Downstream learning	Fine-tune a classifier $P(Y X)$ (or meta-learning before fine-tuning)	Make source and target features indistinguishable and optimize $P(Y X)$	N.A.	N.A.
Inference	$\arg\max_y P(Y = y X)$	$\arg\max_y P(Y = y X)$	Given $\mathcal{Y}_U$ (e.g., owl is an eagle with cat face) k -NN in $\mathcal{Y}$ or $\mathcal{I}$	Various

FIGURE 6.1: Existing paradigm of generalization tasks.

space $\mathcal{X}$. In FSL, the backbone is pre-trained on abundant training samples from a set of base classes (disjoint from the few-shot learning classes). In UDA and SSL, the backbone is directly initialized from one pre-trained on ImageNet [286]. In ZSL and OSR, the backbone is pre-trained on ImageNet. Moreover, using labeled class attributes (e.g., if an animal has a wing) on the training classes, existing methods either learn an encoder mapping from images in $\mathcal{I}$ to the attributes in $\mathcal{Y}$, or learn a generative model from $\mathcal{Y}$ and a learnable sample attribute in $\mathcal{Z}$ to images in $\mathcal{I}$. We denote the two approaches with a double arrow in Figure 6.1. In unsupervised learning, the training samples in $\mathcal{I}$ have no label. Hence the backbone is pre-trained with a label-agnostic pre-text task, e.g., learning augmentation-invariant representation by contrastive pre-text task [351].

2) The second stage aims to fine-tune a classifier using labeled data in the downstream task, where the backbone is either frozen or partially trainable. FSL provides a few training samples (e.g., 1 or 5) per class in the FSL task. In UDA, we have training samples in a domain different from that of testing. In SSL, we are given a small set of training samples (e.g., approximately 100 in total). All of them use maximum likelihood as the training objective (e.g., using cross-entropy loss in classification tasks). In ZSL, OSR, and unsupervised learning, we have no further knowledge about the task, i.e., the knowledge obtained in the first stage is directly used in inference.

In inference, FSL, UDA, and SSL directly leverage the fine-tuned classifier and backbone from the 2nd stage. ZSL and OSR provide the set of unknown class attributes in inference, denoted as $\mathcal{Y}_U$, and the prediction of each test sample is from a k Nearest Neighbor (k -NN) classifier in either attribute space $\mathcal{Y}$ or sample

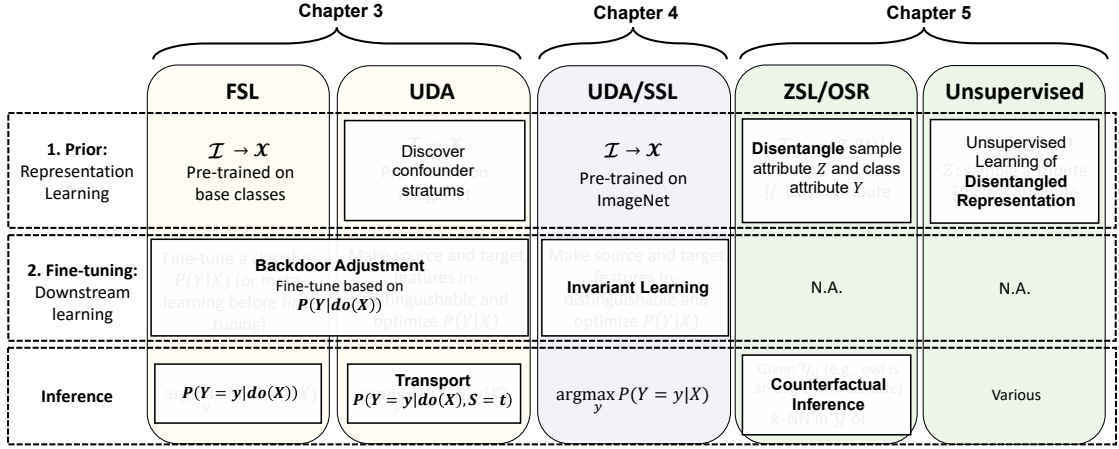


FIGURE 6.2: Our proposed paradigm that incorporates causal intervention. We include the relevant chapter on the top.

space $\mathcal{I}$. For unsupervised learning, the backbone can be evaluated on various downstream tasks, *e.g.*, image classification, segmentation, or object detection.

In Figure 6.2, we show our proposed paradigm, where the white box highlights how we incorporate causal intervention. Specifically,

- In Chapter 3, we first presented a novel casual framework in Section 3.2: Interventional Few-Shot Learning (IFSL) to address the confounding effect from pre-training. We proposed a structural causal model of the causalities in the process of FSL and then developed three practical implementations based on the backdoor adjustment. Our training objective and inference pursue the causal effect $P(Y|do(X))$ instead of maximum likelihood, which is prone to spurious correlation. Then in Section 3.3, we presented a practical implementation of the transportability theory for UDA called Transporting Causal Mechanisms (TCM). In particular, we proposed to identify the confounder stratification by discovering disentangled causal mechanisms (1st stage) and implementing backdoor adjustment by proxy variables (2nd stage). Finally, we transport the learned causal effect to the target domain in inference to address the domain shift.
- In Chapter 4, we first presented a novel UDA method in Section 4.2 based on invariant learning called Invariant CONSistency learning (ICON). ICON pursues an invariant classifier that is simultaneously consistent with the labels in S and the clusters in T , hence removing the spurious correlation inconsistent in T . We show that ICON achieves generalization in T with a

theoretical guarantee. In Section 4.3, we extended ICON to SSL by presenting a novel debiasing regularization for semi-supervised semantic segmentation called Debaised Pseudo-Labeling (DePL). Our approach stems from the fundamental clustering assumption in semi-supervised learning, enabling us to discover the inherent feature distribution in the unlabeled data by unsupervised clustering. Then, by regularizing the classifier to simultaneously separate the labeled classes and unlabeled clusters, the context bias that fails in the unlabeled data is removed.

- In Chapter 5, we first presented a novel counterfactual framework for Zero-Shot Learning (ZSL) and Open-Set Recognition (OSR) in Section 5.2. Specifically, we proposed a Generative Causal Model that group disentangles class attributes and sample attributes, which generates faithful counterfactuals. This allows us to use counterfactual inference for balanced binary seen/unseen classification. In Section 5.3, we presented a novel unsupervised disentangled representation learning method by leveraging the inductive bias of diffusion time-steps. The learned feature supports accurate downstream inference on facial attributes and enables counterfactual generation, validating its disentanglement quality.

Notably, the field of machine learning is undergoing a large paradigm shift towards foundation models, where gigantic models trained with internet-scale data demonstrate zero-shot generalizability to a wide variety of tasks. This implies that representation learning (*i.e.*, 1st stage in Figure 6.2) in the future may be unified by a small set of foundation models. We expect them to become new baselines that can be fine-tuned to downstream tasks (*i.e.*, 2nd stage). Future work can identify the bias in foundation models, and explore practical approaches for unbiased fine-tuning.

Besides the methods discussed in this thesis, we highlight some alternative approaches to achieve generalization. In light of the rapid improvements in computer graphic, one can potentially build a simulator that generates diverse and realistic training data, where no attribute is spurious. In fact, this approach is being actively explored in robotic AI [352]. Other potential approaches include (but are not limited to) improving neural network designs to incorporate additional inductive bias [353], or delving into brain-inspired or connectionism AI [354].

List of Author's Awards, and Publications¹

Awards

- 2022 PREMIA Best Student Paper Awards, *PREMIA*.

Conference Proceedings

- Jiali Ma, **Zhongqi Yue**, Tomoyuki Kagaya, Tomoki Suzuki, Karlekar Jayashree, Sugiri Pranata, Hanwang Zhang. Invariant Feature Regularization for Fair Face Recognition. 2023. *International Conference on Computer Vision (ICCV)*.
- Yanghao Wang, **Zhongqi Yue**, Xian-Sheng Hua, Hanwang Zhang. Random Boxes Are Open-world Object Detectors. 2023. *International Conference on Computer Vision (ICCV)*.
- Hui Lv, **Zhongqi Yue**, Qianru Sun, Bin Luo, Zhen Cui, Hanwang Zhang. Unbiased Multiple Instance Learning for Weakly Supervised Video Anomaly Detection. 2023. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Tan Wang, **Zhongqi Yue**, Jianqiang Huang, Qianru Sun, Hanwang Zhang. Self-Supervised Learning Disentangled Group Representation as Feature. 2021. *Conference on Neural Information Processing Systems (NeurIPS)*. (Spotlight Presentation 260/9122; 2022 PREMIA Best Student Paper).

¹The superscript * indicates joint first authors

- **Zhongqi Yue**, Qianru Sun, Xian-Sheng Hua, Hanwang Zhang. Transporting Causal Mechanisms for Unsupervised Domain Adaptation. 2021. *International Conference on Computer Vision (ICCV)*. (Oral Presentation 210/6236).
- **Zhongqi Yue***, Tan Wang*, Qianru Sun, Xian-Sheng Hua, Hanwang Zhang. Counterfactual Zero-Shot and Open-Set Visual Recognition. 2021. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- **Zhongqi Yue**, Hanwang Zhang, Qianru Sun, Xian-Sheng Hua. Interventional Few-Shot Learning. 2020. *Conference on Neural Information Processing Systems (NeurIPS)*.

Bibliography

- [1] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 2012. [xix](#), [xxi](#), [3](#), [66](#)
- [2] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008. [xix](#), [xxi](#), [3](#), [66](#), [69](#)
- [3] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. [xix](#), [xxii](#), [xxv](#), [5](#), [41](#), [49](#), [50](#), [51](#), [58](#), [68](#), [73](#), [75](#), [78](#)
- [4] 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*, 2017. [xix](#), [xx](#), [5](#), [31](#), [33](#), [78](#)
- [5] 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*, 2016. [xx](#), [xxi](#), [xxiv](#), [24](#), [25](#), [27](#), [52](#), [53](#), [60](#), [116](#)
- [6] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008. [xx](#), [xxi](#), [xxiii](#), [25](#), [51](#), [52](#), [104](#)
- [7] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. [xx](#), [xxi](#), [xxv](#), [xxvi](#), [38](#), [41](#), [48](#), [49](#), [50](#), [52](#), [58](#), [59](#), [60](#), [68](#), [69](#), [73](#), [75](#), [76](#)
- [8] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. [xx](#), [15](#), [43](#), [44](#), [111](#)
- [9] Imageclef-da. <http://imageclef.org/2014/adaptation>, 2014. [xx](#), [xxv](#), [41](#), [48](#), [49](#), [50](#), [51](#), [58](#)

- [10] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, 2020. [xxi](#), [48](#), [49](#), [51](#), [52](#), [53](#), [60](#), [74](#), [75](#)
- [11] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. 2013. [xxi](#), [66](#)
- [12] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. In *NeurIPS*, 2021. [xxii](#), [15](#), [66](#), [74](#), [75](#), [77](#), [78](#)
- [13] Lihe Yang, Lei Qi, Litong Feng, Wayne Zhang, and Yinghuan Shi. Revisiting Weak-to-Strong Consistency in Semi-Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [xxii](#), [xxiii](#), [xxvi](#), [17](#), [82](#), [90](#), [91](#), [93](#), [94](#), [95](#), [96](#), [97](#), [98](#)
- [14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. [xxii](#), [82](#), [84](#), [89](#)
- [15] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)*, 2015. [xxiii](#), [92](#), [96](#), [97](#)
- [16] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical report, California Institute of Technology, 2010. [xxiii](#), [23](#), [30](#), [104](#), [112](#), [114](#)
- [17] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *ECCV*, 2020. [xxiii](#), [xxvi](#), [xxvii](#), [17](#), [104](#), [113](#), [114](#), [115](#)
- [18] Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional gaussian distribution learning for open set recognition. In *CVPR*, 2020. [xxiv](#), [xxvii](#), [17](#), [107](#), [114](#), [116](#)
- [19] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, 2019. [xxiv](#), [116](#)
- [20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. [xxiv](#), [112](#), [117](#)
- [21] Zijian Zhang, Zhou Zhao, and Zhijie Lin. Unsupervised representation learning from pre-trained diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 35:22117–22130, 2022. [xxiv](#), [18](#), [122](#), [125](#), [126](#), [128](#)

- [22] 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*, 2016. [xxv](#), [21](#), [27](#), [30](#), [34](#), [46](#), [48](#), [49](#), [68](#), [74](#), [75](#), [114](#), [120](#)
- [23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015. [xxv](#), [48](#), [49](#), [70](#), [74](#), [76](#)
- [24] Shiori Sagawa, Pang Wei Koh, Tony Lee, Irena Gao, Sang Michael Xie, Kendrick Shen, Ananya Kumar, Weihua Hu, Michihiro Yasunaga, Henrik Marklund, et al. Extending the wilds benchmark for unsupervised adaptation. In *ICLR*, 2022. [xxvi](#), [66](#), [68](#), [73](#), [74](#)
- [25] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-Supervised Semantic Segmentation Using Unreliable Pseudo Labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [xxvi](#), [89](#), [90](#), [91](#), [93](#)
- [26] He Huang, Changhu Wang, Philip S Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. In *CVPR*, 2019. [xxvi](#), [113](#)
- [27] Yongqin Xian, H. Christoph Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. 2020. URL <https://drive.google.com/file/d/1p9gtkuHCCyjkyezSarCw-1siCSXUyKH/view>. [xxvi](#), [113](#)
- [28] Guo-Sen Xie, Li Liu, Xiaobo Jin, Fan Zhu, Zheng Zhang, Jie Qin, Yazhou Yao, and Ling Shao. Attentive region embedding network for zero-shot learning. In *CVPR*, 2019. [xxvi](#), [113](#)
- [29] Yongqin Xian, H. Christoph Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 2018. [xxvii](#), [2](#), [17](#), [103](#), [106](#), [112](#), [114](#), [115](#)
- [30] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, 2016. [xxvii](#), [17](#), [103](#), [116](#)
- [31] Steven Sloman. *Causal Models: How People Think about the World and Its Alternatives*. Oxford University Press, 08 2005. ISBN 9780195183115. [1](#)
- [32] Alison Gopnik and Laura Schulz. *Causal Learning: Psychology, Philosophy, and Computation*. Oxford University Press, 04 2007. ISBN 9780195176803. [1](#)

- [33] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *CVPR*, 2018. 2, 17, 39, 104, 113
- [34] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017. 2, 35
- [35] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations*, 2020. 2, 14, 21, 31, 33
- [36] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 2010. 2, 15, 38, 65
- [37] Gong Mingming, Zhang Kun, Liu Tongliang, Tao Dacheng, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International Conference on Machine Learning*, 2016. 2, 13
- [38] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018. 2, 8, 18, 40, 41, 54, 65, 71, 101, 105, 111, 120
- [39] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 2021. 2, 65
- [40] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2nd edition, 2009. ISBN 052189560X. 5, 6, 13, 20, 26, 39, 40, 55, 104, 106
- [41] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. 6, 38
- [42] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *Advances in Neural Information Processing Systems*, 2020. 6, 13
- [43] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 7, 17, 64, 67, 70
- [44] Michel Besserve, Rémy Sun, and Bernhard Schölkopf. Counterfactuals uncover the modular structure of deep generative models. *arXiv preprint arXiv:1812.03253*, 2018. 8, 24, 120, 132

- [45] Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect. *arXiv preprint arXiv:2003.03923*, 2020. [8](#)
- [46] Elliot Creager, Joern-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 2189–2200, 2021. [8](#)
- [47] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*, 2019. [8](#), [18](#), [41](#), [102](#), [104](#), [110](#)
- [48] Poojan Oza and Vishal M Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *CVPR*, 2019. [9](#), [17](#), [103](#), [106](#), [107](#)
- [49] Judea Pearl and Elias Bareinboim. External validity: From do-calculus to transportability across populations. *Statistical Science*, 2014. [10](#), [15](#), [39](#)
- [50] Sara Magliacane, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems*, 2018. [13](#)
- [51] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*, 2019. [13](#)
- [52] Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect. *arXiv preprint arXiv:2003.03923*, 2020. [13](#), [29](#)
- [53] Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual causal feature learning. In *Uncertainty in Artificial Intelligence*, 2015. [13](#)
- [54] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [13](#)
- [55] Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, 2019. [13](#)
- [56] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. In *Advances in Neural Information Processing Systems*, 2020. [13](#)
- [57] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. [13](#), [32](#), [33](#)

- [58] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. [13](#)
- [59] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. In *Proceedings of the International Conference on Learning Representations Workshops*, 2018. [13](#)
- [60] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems*, 2018. [13](#)
- [61] Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In *International Conference on Machine Learning*, 2018. [13](#)
- [62] Raphael Suter, ore Miladinović, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, 2019. [13](#)
- [63] Qi Jiaxin, Niu Yulei, Huang Jianqiang, and Zhang Hanwang. Two causal principles for improving visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. [13](#)
- [64] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [65] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. [13](#)
- [66] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. [14](#), [21](#), [33](#), [35](#), [36](#), [37](#)
- [67] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017. [14](#), [21](#), [23](#), [24](#), [30](#), [31](#), [33](#), [35](#), [36](#)
- [68] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018. [14](#)
- [69] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019. [14](#), [30](#), [31](#), [35](#), [37](#)

- [70] Huaiyu Li, Weiming Dong, Xing Mei, Chongyang Ma, Feiyue Huang, and Bao-Gang Hu. Lgm-net: Learning to generate matching networks for few-shot learning. In *International Conference on Machine Learning*, 2019. [14](#)
- [71] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016. [14](#), [23](#), [24](#), [30](#), [31](#), [35](#), [37](#)
- [72] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover’s distance and structured classifiers. 2020. [14](#), [33](#)
- [73] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. [14](#), [30](#), [31](#), [35](#), [36](#)
- [74] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, 2018. [14](#)
- [75] Shell Xu Hu, Pablo Moreno, Yang Xiao, Xi Shen, Guillaume Obozinski, Neil Lawrence, and Andreas Damianou. Empirical bayes transductive meta-learning with synthetic gradients. In *International Conference on Learning Representations*, 2020. [14](#), [23](#), [30](#), [31](#), [33](#), [35](#), [37](#)
- [76] Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *Advances in Neural Information Processing Systems*, 2019. [14](#)
- [77] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. [14](#), [33](#)
- [78] Yaoyao Liu, Bernt Schiele, and Qianru Sun. An ensemble of epoch-wise empirical bayes for few-shot learning. In *European Conference on Computer Vision*, 2020. [14](#)
- [79] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. In *ECCV*, 2022. [14](#)
- [80] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. [14](#)
- [81] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2009. [14](#)
- [82] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. [14](#)

- [83] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2015.
- [84] Youshan Zhang and Brian D Davison. Impact of imagenet model selection on domain adaptation. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, 2020. [14](#)
- [85] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, 2014. [14](#)
- [86] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019. [14](#)
- [87] Tiago Ramalho, Thierry Soubie, and Stefano Peluchetti. An empirical study of pretrained representations for few-shot classification. *arXiv preprint arXiv:1910.01319*, 2019. [14](#)
- [88] Marc Tanti, Albert Gatt, and Kenneth P Camilleri. Transfer learning from language models to image caption generators: Better models may not transfer better. *arXiv preprint arXiv:1901.01216*, 2019. [14](#)
- [89] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 2007. [14](#), [38](#), [39](#)
- [90] Qingchao Chen, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In *CVPR*, 2018. [14](#)
- [91] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018. [14](#), [39](#), [49](#)
- [92] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, 2018. [14](#)
- [93] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 2016. [15](#), [48](#), [49](#), [51](#)
- [94] Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018. [15](#), [46](#), [48](#), [49](#), [51](#), [66](#), [73](#), [74](#)

- [95] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR*, 2020. 15
- [96] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 15, 38, 39, 48, 49, 51
- [97] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 15, 66
- [98] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *CVPR*, 2014. 15
- [99] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NIPS*, 2016. 15, 39
- [100] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, 2013. 15, 38, 66
- [101] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019. 15, 46, 74
- [102] Yabin Zhang, Bin Deng, Kui Jia, and Lei Zhang. Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. In *ECCV*, 2020. 15, 49
- [103] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *ICML*, 2017. 15
- [104] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, 2019. 15, 48, 49
- [105] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2004. 15
- [106] David Berthelot, Nicholas Carlini, Ian Goodfellow, Avital Oliver, Nicolas Papernot, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.
- [107] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 15, 77

- [108] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *NeurIPS*, 2016. 15
- [109] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *ICML*, 2017. 15
- [110] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *ICLR*, 2018. 15, 74, 75
- [111] Jian Liang, Dapeng Hu, Jiashi Feng, and Ran He. Dine: Domain adaptation from single and multiple black-box predictors. In *CVPR*, 2022. 15
- [112] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, 2016. 15, 39
- [113] Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. In *IJCAI*, 2019. 15
- [114] Shuhao Cui, Xuan Jin, Shuhui Wang, Yuan He, and Qingming Huang. Heuristic domain adaptation. In *NeurIPS*, 2020. 15
- [115] J. Pearl, M. Glymour, and N.P. Jewell. *Causal Inference in Statistics: A Primer*. Wiley, 2016. ISBN 9781119186847. 15, 23, 24, 26, 27, 40, 54, 104, 108
- [116] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, 2020. 15
- [117] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *CVPR*, 2021. 15, 66, 68, 75
- [118] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-supervised learning*. MIT Press, 1st edition, 2006. 16, 84, 86
- [119] Andrew B Goldberg, Xiaojin Zhu, Ajit Singh, Zoubin Xu, and Robert D Nowak. Multi-manifold semi-supervised learning. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pages 169–176, 2009. 16, 84, 86
- [120] Dong-Hyun Lee, Joonseok Kim, and Junmo Kang. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. In *ICML Workshop*, 2013. 16
- [121] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 16, 82, 85, 90, 91

- [122] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019. 16, 82, 85
- [123] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 2013. 16, 82
- [124] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-Training With Noisy Student Improves ImageNet Classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 82
- [125] Nayeem Rizve, Kevin Duarte, Yogesh Rawat, and Mubarak Shah. In Defense of Pseudo-Labeling: An Uncertainty-Aware Pseudo-label Selection Framework for Semi-Supervised Learning. In *International Conference on Learning Representations*, 2021. 16, 82
- [126] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, 2016. 16, 82, 85
- [127] Samuli Laine and Timo Aila. Temporal Ensembling for Semi-Supervised Learning. In *International Conference on Learning Representations*, 2016. 16, 82, 85
- [128] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 16, 81, 84, 89, 90, 92
- [129] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 16
- [130] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-Supervised Semantic Segmentation with High- and Low-Level Consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 16
- [131] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. In *British Machine Vision Conference (BMVC)*, 2018. 16
- [132] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 16

- [133] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. [16](#)
- [134] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. ST++: Make Self-training Work Better for Semi-supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [16](#), [90](#), [91](#)
- [135] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-Supervised Semantic Segmentation via Adaptive Equalization Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [17](#), [89](#), [91](#), [93](#)
- [136] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. PseudoSeg: Designing Pseudo Labels for Semantic Segmentation. In *International Conference on Learning Representations (ICLR)*, 2021. [16](#)
- [137] Kwonjoon Lee, Junsoo Lee, and Jinwoo Shin. Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2020. [16](#), [82](#)
- [138] Zirui Wang, Zihang Dai, Barnabas Póczos, and Jaime Carbonell. Characterizing and Avoiding Negative Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [16](#)
- [139] Xin Lai, Zhuotao Tian, Li Jiang, Shu Liu, Hengshuang Zhao, Liwei Wang, and Jiaya Jia. Semi-Supervised Semantic Segmentation with Directional Context-Aware Consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [17](#), [89](#), [91](#)
- [140] Geoff French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. In *British Machine Vision Conference*. BMVA Press, 2020.
- [141] Jianlong Yuan, Yifan Liu, Chunhua Shen, Zhibin Wang, and Hao Li. A Simple Baseline for Semi-supervised Semantic Segmentation with Strong Data Augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. [17](#)
- [142] Yuyuan Liu, Yu Tian, Yuanhong Chen, Fengbei Liu, Vasileios Belagiannis, and Gustavo Carneiro. Perturbed and Strict Mean Teachers for Semi-Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [17](#), [90](#), [91](#), [93](#)

- [143] Samuli Laine and Timo Aila. Temporal Ensembling for Semi-Supervised Learning. In *International Conference on Learning Representations (ICLR)*, 2017. [17](#)
- [144] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. [17](#)
- [145] Minsung Hyun, Jisoo Jeong, and Nojun Kwak. Class-imbalanced semi-supervised learning. *arXiv preprint arXiv:2002.06815*, 2020. [17](#)
- [146] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. *arXiv preprint arXiv:2102.09559*, 2021. [17](#)
- [147] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Ar’echiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. [17](#)
- [148] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [17](#)
- [149] Baixu Chen, Janguang Jiang, Ximei Wang, Pengfei Wan, Jianmin Wang, and Mingsheng Long. Debaised self-training for semi-supervised learning, 2022. [17](#)
- [150] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. [17](#), [103](#)
- [151] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc’Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013. [17](#), [103](#), [113](#)
- [152] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *TPAMI*, 2015. [113](#)
- [153] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Transferable contrastive network for generalized zero-shot learning. In *ICCV*, 2019. [17](#)
- [154] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018. [17](#), [104](#), [107](#), [114](#)
- [155] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *CVPR*, 2019. [106](#), [107](#), [113](#), [114](#), [115](#)

- [156] Ayyappa Pambala, Titir Dutta, and Soma Biswas. Generative model with semantic embedding and integrated classifier for generalized zero-shot learning. In *WACV*, 2020. [17](#)
- [157] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boulton. Toward open set recognition. *TPAMI*, 2013. [17](#)
- [158] Walter J Scheirer, Lalit P Jain, and Terrance E Boulton. Probability models for open set recognition. *TPAMI*, 2014. [17](#), [103](#)
- [159] Sergey Demyanov Zongyuan Ge and Rahil Garnavi. Generative openmax for multi-class open set classification. In *BMVC*, 2017. [17](#)
- [160] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*, 2020. [17](#)
- [161] Weitang Liu, Xiaoyun Wang, John Owens, and Sharon Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020. [17](#)
- [162] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. Anomalous instance detection in deep learning: A survey. *arXiv preprint arXiv:2003.06979*, 2020. [18](#)
- [163] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot action recognition. In *CVPR*, 2019. [18](#)
- [164] Xingyu Chen, Xuguang Lan, Fuchun Sun, and Nanning Zheng. A boundary based out-of-distribution classifier for generalized zero-shot learning. In *ECCV*, 2020. [18](#)
- [165] Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009. [18](#)
- [166] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [18](#)
- [167] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *ICML*, 2019. [41](#), [45](#), [105](#), [111](#), [136](#)
- [168] Sjoerd Van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? In *Advances in neural information processing systems*, 2019.
- [169] F. Locatello, M. Tschannen, S. Bauer, G. Rätsch, B. Schölkopf, and O. Bachem. Disentangling factors of variations using few labels. In *8th International Conference on Learning Representations (ICLR)*, 2020. [18](#)

- [170] Kien Do and Truyen Tran. Theory and evaluation metrics for learning disentangled representations. In *International conference on learning representations*, 2020. [18](#)
- [171] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International conference on learning representations*, 2018. [18](#)
- [172] Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. In *Advances in neural information processing systems*, 2018. [18](#)
- [173] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, 2016. [18](#)
- [174] Raphael Suter, Djordje Miladinovic, Stefan Bauer, and Bernhard Schölkopf. Interventional robustness of deep latent variable models. *arXiv*, 2018. [18](#)
- [175] Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Multi-view perceptron: a deep model for learning face identity and view representations. *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 2014. [18](#)
- [176] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in neural information processing systems*, 2018.
- [177] Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. In *IJCAI: proceedings of the conference*, 2019.
- [178] Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *International conference on machine learning*, 2014.
- [179] Theofanis Karaletsos, Serge Belongie, and Gunnar Rätsch. Bayesian representation learning with oracle constraints. *International conference on learning representations*, 2015. [18](#)
- [180] Utkarsh Ojha, Krishna Kumar Singh, Cho-Jui Hsieh, and Yong Jae Lee. Elastic-infogan: Unsupervised disentangled representation learning in class-imbalanced data. In *Advances in neural information processing systems*, 2020. [18](#)
- [181] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *International Conference on Machine Learning*, 2020.

- [182] Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Do generative models know disentanglement? contrastive learning is all you need. *arXiv preprint arXiv:2102.10543*, 2021. [18](#)
- [183] I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. [18](#), [111](#)
- [184] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in neural information processing systems*, 2018.
- [185] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [186] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, 2018. [18](#)
- [187] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. [18](#), [131](#)
- [188] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. [18](#)
- [189] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. *ICLR*, 2022. [18](#)
- [190] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10619–10629, 2022. [18](#), [122](#)
- [191] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 1995. [19](#)
- [192] Yaqing Wang and Quanming Yao. Few-shot learning: A survey. *arXiv preprint arXiv:1904.05046*, 2019. [21](#)
- [193] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. [21](#)
- [194] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. [21](#)

- [195] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems Deep Learning Workshop*, 2014. [21](#), [28](#)
- [196] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006. [21](#)
- [197] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. [21](#)
- [198] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. [21](#), [28](#)
- [199] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019. [21](#), [27](#)
- [200] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, 2016. [21](#)
- [201] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. [21](#)
- [202] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. [21](#), [24](#), [33](#)
- [203] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018. [23](#), [30](#)
- [204] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000. [24](#)
- [205] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000. [24](#)
- [206] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, 2019. [24](#)

- [207] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 2014. 24, 27
- [208] Matthew Turk and Alex Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991. 24
- [209] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 2011. 24
- [210] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 24, 120
- [211] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012. 25
- [212] F.M. Dekking, C. Kraaikamp, H.P. Lopuhaä, and L.E. Meester. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Springer Texts in Statistics. Springer, 2005. ISBN 9781852338961. 26
- [213] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016. ISBN 0262035618. 26
- [214] Joshua D Angrist and Alan B Krueger. Instrumental variables and the search for identification: From supply and demand to natural experiments. *Journal of Economic Perspectives*, 2001. 26
- [215] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. <https://github.com/facebookresearch/detectron>, 2018. 27
- [216] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, 2015. 29
- [217] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017. 30
- [218] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016. 30, 34
- [219] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 33

- [220] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. [33](#)
- [221] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019. [35](#)
- [222] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, 2007. [38](#)
- [223] Lixin Duan, Ivor W Tsang, Dong Xu, and Tat-Seng Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 289–296, 2009. [38](#)
- [224] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010. [38](#)
- [225] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *ICML*, 2013. [38](#)
- [226] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 2009. [38](#)
- [227] Sandeepkumar Satpal and Sunita Sarawagi. Domain adaptation of conditional probability models via feature subsetting. In *European Conference on Principles of Data Mining and Knowledge Discovery*, 2007. [38](#)
- [228] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *ICML*, pages 2747–2755, 2018. [39](#)
- [229] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. [39](#)
- [230] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. [39](#)

- [231] Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In *ICML*, 2018. [40](#), [42](#), [44](#), [105](#)
- [232] Wang Miao, Zhi Geng, and Eric J Tchetgen Tchetgen. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 2018. [40](#), [45](#), [55](#), [56](#), [57](#)
- [233] M. Besserve, A. Mehrjou, R. Sun, and B. Schölkopf. Counterfactuals uncover the modular structure of deep generative models. In *ICLR*, 2020. [42](#), [54](#), [105](#), [110](#), [118](#)
- [234] Zhongqi Yue, Tan Wang, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Counterfactual zero-shot and open-set visual recognition. In *CVPR*, 2021. [42](#), [120](#)
- [235] Diederik Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. [45](#), [111](#), [120](#)
- [236] Jiajun Lu, Theerasit Issaranon, and David Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *ICCV*, 2017. [47](#)
- [237] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018. [48](#)
- [238] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, 2019. [48](#), [51](#)
- [239] Vinod Kumar Kurmi, Shanu Kumar, and Vinay P Namboodiri. Attending to discriminative certainty for domain adaptation. In *CVPR*, 2019. [48](#)
- [240] Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, and Chuan-Xian Ren. Enhanced transport distance for unsupervised domain adaptation. In *CVPR*, 2020. [48](#), [49](#)
- [241] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. [49](#), [66](#)
- [242] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, 2018. [49](#)
- [243] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, 2019. [49](#)
- [244] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, 2018. [51](#)
- [245] Yuan Wu, Diana Inkpen, and Ahmed El-Roby. Dual mixup regularized learning for adversarial domain adaptation. In *ECCV*, 2020. [51](#)

- [246] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 2010. [52](#)
- [247] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *ICML*, 2019. [52](#), [66](#)
- [248] Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. 2018. ISBN 046509760X. [53](#), [105](#)
- [249] Michael Puthawala, Konik Kothari, Matti Lassas, Ivan Dokmanić, and Maarten de Hoop. Globally injective relu networks. *arXiv preprint arXiv:2006.08464*, 2020. [54](#), [118](#)
- [250] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [58](#)
- [251] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33: 18661–18673, 2020. [63](#)
- [252] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, 2021. [64](#), [70](#), [76](#)
- [253] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012. [66](#)
- [254] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *CVPR*, 2021. [66](#)
- [255] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, 2019. [66](#)
- [256] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019. [66](#)
- [257] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *ECCV*, 2020.
- [258] Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In *ECCV*, 2020. [66](#)

- [259] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *CVPR*, 2022. [68](#), [75](#)
- [260] Saquib Sarfraz, Marios Koulakis, Constantin Seibold, and Rainer Stiefelhagen. Hierarchical nearest neighbor graph embedding for efficient dimensionality reduction. In *CVPR*, 2022. [69](#)
- [261] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. [69](#)
- [262] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *ICLR*, 2020. [69](#)
- [263] Tan Wang, Zhongqi Yue, Jianqiang Huang, Qianru Sun, and Hanwang Zhang. Self-supervised learning disentangled group representation as feature. In *NeurIPS*, 2021. [71](#), [72](#), [101](#), [120](#)
- [264] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *ICCV*, 2021. [74](#), [75](#)
- [265] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. [74](#)
- [266] Zhongqi Yue, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Transporting causal mechanisms for unsupervised domain adaptation. In *ICCV*, 2021. [75](#)
- [267] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, Zhizheng Zhang, and Zhibo Chen. Toalign: Task-oriented alignment for unsupervised domain adaptation. In *NeurIPS*, 2021. [75](#)
- [268] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *CVPR*, 2021. [75](#)
- [269] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.
- [270] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.
- [271] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

- [272] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013.
- [273] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, 2016. [81](#)
- [274] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Veronica Villena-Martinez, Pablo Martinez-Gonzalez, and Jose Garcia-Rodriguez. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, 2018. [81](#)
- [275] Ziyuan Wang, Jie Cui, Hongbin Zha, Masataka Kagesawa, Satoshi Ono, and Katsushi Ikeuchi. Foreground object detection by motion-based grouping of object parts. *International Journal of Intelligent Transportation Systems Research*, 2014. [81](#)
- [276] Jose M Alvarez, Theo Gevers, Yann LeCun, and Antonio M Lopez. Road Scene Segmentation from a Single Image. In *European Conference on Computer Vision*. Springer, 2012. [81](#)
- [277] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving. *arXiv preprint arXiv:1612.07695*, 2016. [81](#)
- [278] A Sinha and J Dolz. Multi-scale Self-guided Attention for Medical Image Segmentation. *IEEE Journal of Biomedical and Health Informatics*, 2020. [81](#)
- [279] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015. [81](#)
- [280] A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Systems with Applications*, 2021. [81](#)
- [281] Yanzhou Su, Jian Cheng, Haiwei Bai, Haijun Liu, and Changtao He. Semantic segmentation of very-high-resolution remote sensing images via deep multi-feature learning. *Remote Sensing*, 2022. [81](#)
- [282] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 2013. [81](#)
- [283] Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2016. [83](#)

- [284] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. [85](#), [91](#)
- [285] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [85](#), [91](#)
- [286] 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. [85](#), [138](#)
- [287] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *ArXiv*, abs/1706.05587, 2017. [85](#)
- [288] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-Supervised Semantic Segmentation with Cross-Consistency Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7847–7856, 2020. [86](#)
- [289] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *International Conference on Learning Representations*, 2020. [88](#)
- [290] Bharath Hariharan, Pablo Arbel'aez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic Contours from Inverse Detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2011. [89](#)
- [291] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-Supervised Semantic Segmentation with Cross Pseudo Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7585–7594, 2021. [90](#), [91](#), [93](#)
- [292] Shikun Liu, Shuaifeng Zhi, Edward Johns, and Andrew Davison. Bootstrapping semantic segmentation with regional contrast. In *International Conference on Learning Representations (ICLR)*, 2022. [90](#)
- [293] Ying Jin, Jiaqi Wang, and Dahua Lin. Semi-Supervised Semantic Segmentation via Gentle Teaching Assistant. In *Advances in Neural Information Processing Systems*, 2022. [90](#), [93](#)
- [294] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised Data Augmentation for Consistency Training. 2019. [90](#)

- [295] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. [90](#)
- [296] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. [90](#)
- [297] Ning Qian. On the momentum term in gradient descent learning algorithms. In *Neural Networks*, 1999. [91](#)
- [298] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations (ICLR)*, 2021. [92](#)
- [299] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016. [92](#)
- [300] Judea Pearl. Causality: Models, reasoning, and inference, second edition. *Causality*, 2000. [92](#)
- [301] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. CReST: A Class-Rebalancing Self-Training Framework for Imbalanced Semi-Supervised Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [94](#)
- [302] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing Biased Pseudo Labels for Semi-supervised Semantic Segmentation: A Baseline Investigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. [94](#)
- [303] Dayan Guan, Jiaxing Huang, Aoran Xiao, and Shijian Lu. Unbiased Subclass Regularization for Semi-Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9968–9978, 2022. [94](#)
- [304] D. Marr. *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. 2010. ISBN 9780262514620. [102](#)
- [305] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016. [103](#), [106](#), [113](#)
- [306] D. O. Cardoso, F. França, and J. Gama. A bounded neural network for open set recognition. In *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015. [103](#)

- [307] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 2017. 103
- [308] Bernhard Schölkopf. Causality for machine learning. *arXiv preprint arXiv:1911.10500*, 2019. 103
- [309] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, 2019. 103
- [310] Michel Besserve, Naji Shajarisales, Bernhard Schölkopf, and Dominik Janzing. Group invariance principles for causal generative models. In *International Conference on Artificial Intelligence and Statistics*, 2018. 105
- [311] Michel Besserve, Rémy Sun, Dominik Janzing, and Bernhard Schölkopf. A theory of independent mechanisms for extrapolation in generative models. *arXiv preprint arXiv:2004.00184*, 2020. 105, 111, 136
- [312] Eli Schwartz, Leonid Karlinsky, Rogerio Feris, Raja Giryes, and Alex M Bronstein. Baby steps towards few-shot learning with multiple semantics. In *CVPR*, 2019. 106
- [313] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *CVPR*, 2020. 106
- [314] Kosuke Imai, Dustin Tingley, and Teppei Yamamoto. Experimental designs for identifying causal mechanisms. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 2013. 106
- [315] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 111, 127, 128
- [316] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. *Proceedings of Machine Learning Research*. PMLR, 2017. 112
- [317] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 112, 114
- [318] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 112, 114
- [319] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL <http://yann.lecun.com/exdb/mnist/>. 112
- [320] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 112

- [321] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *ECCV*, 2018. [112](#)
- [322] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *CVPR*, 2019. [112](#)
- [323] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016. [113](#)
- [324] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. [113](#), [115](#)
- [325] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *CVPR*, 2019. [113](#), [115](#)
- [326] Xin Sun, Chi Zhang, Guosheng Lin, and Keck-Voon Ling. Open set recognition with conditional probabilistic generative models. *arXiv preprint arXiv:2008.05129v2*, 2020. [114](#)
- [327] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016. [115](#)
- [328] Mark Anthony Armstrong. *Basic topology*. Springer Science & Business Media, 2013. [118](#)
- [329] 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. [120](#)
- [330] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. [120](#)
- [331] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021. [120](#)
- [332] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34: 8780–8794, 2021. [120](#)

- [333] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. [120](#)
- [334] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. [120](#), [131](#), [136](#)
- [335] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. [120](#), [136](#)
- [336] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018. [120](#)
- [337] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015. [120](#)
- [338] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.
- [339] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017. [120](#)
- [340] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. [122](#), [128](#)
- [341] 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. [122](#), [128](#)
- [342] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. [122](#), [123](#), [133](#)
- [343] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. [123](#), [133](#)

- [344] Henry F Inman and Edwin L Bradley Jr. The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. *Communications in Statistics-theory and Methods*, 18(10):3851–3874, 1989. [124](#)
- [345] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. [127](#), [128](#)
- [346] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. [127](#)
- [347] Gary B. Huang, Vidit Jain, and Erik Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, 2007. [127](#)
- [348] Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. [129](#)
- [349] Ruey-Pyng Lu, Eric P Smith, and IJ Good. Multivariate measures of similarity and niche overlap. *Theoretical Population Biology*, 35(1):1–21, 1989. [135](#)
- [350] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. [137](#)
- [351] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. [138](#)
- [352] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, C. Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks, 2021. [140](#)
- [353] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022. [140](#)
- [354] Ashok K. Goel. Looking back, looking ahead: Symbolic versus connectionist ai. *AI Mag.*, 42(4):83–85, dec 2021. ISSN 0738-4602. doi: 10.1609/aaai.12026. URL <https://doi.org/10.1609/aaai.12026>. [140](#)