Causally-Inspired Generalizable Deep Learning Methods under Distribution Shifts

Jiaxian Guo

SID: 480414221

Supervisor: Prof Dacheng Tao
Auxiliary Supervisor: Dr Tongliang Liu

A thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy

School of Computer Science
Faculty of Engineering
The University of Sydney
Australia

11 April 2023

Statement of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes. I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Jiaxian Guo
School of Computer Science
Faculty of Engineering
The University of Sydney
11 April 2023

Acknowledgements

Throughout my Ph.D. studies, I received tremendous help from my supervisors, colleagues, friends, girlfriend, and family. This thesis will never be finished without their assistance.

First and foremost, I would like to express my heartfelt gratitude to Prof. Dacheng Tao, my primary supervisor, for his meticulous guidance, insightful advice, and kind support throughout my Ph.D. study. I will never forget how Prof. Tao helped me revise my research paper and rebuttals until midnight. I could not have completed my Ph.D. without his academic discussions and hardware support. Furthermore, Prof. Tao's work ethic, perceptive vision, and rigors of research approach will inspire and guide me for the rest of my life.

I am also particularly grateful to my collaborator, Dr. Mingming Gong (senior lecturer), who has offered me countless support and suggestions throughout my Ph.D. candidature. Dr. Mingming Gong provides many detailed suggestions for my research. The patience and responsible teaching attitude towards students of Dr. Gong will guide me for my rest life.

I would like to thank my friends and colleagues: Mr. Xinqi Zhu, Mr. Youjian Zhang, Dr. Shuo Yang, Mr. Chenwei Ding, Mr. Zhen Wang, Mr. Huihui Gong, Ms. Qi Zheng, Ms. Sihan Ma, Mr. Cheng Wen, Mr. Sen Zhang, Dr. Shanshan Zhao, Mr. Xikun Zhang, Mr. Kaining Zhang, Mr. Hao Guan, Mr. Lianbo Zhang, Dr. Zeyu Feng, Dr. Baosheng Yu, Dr. Liang Ding, Mr. Yufei Xu, Mr. Qiming Zhang, Mr. Haoyu He, Dr. Dongxu Li, Dr. Junnan Li, Ms. Yifen Li, Dr. Benteng Ma, Dr. Chaoyue Wang, Ms. Xiaofei Liu, Ms. Chen Chen, Ms. Haoning Xi and Ms. Jindou Zhong , for supports, company and discussions during my Ph.D. time, especially during the tiring COVID pandemic.

Finally, I would like to thank my girlfriend, Maruge Zhao. It is your company, encouragement, and support that keeps my life full of love and enjoyment, especially at

the anxious deadline of paper submission. This thesis is also dedicated to my parents for all the years of their countless love and support.

List of Publications

(1) **Jiaxian Guo**, Mingming Gong, Tongliang Liu, Kun Zhang and Dacheng Tao. " LTF: A Label Transformation Framework for Correcting Label Shift ", in *International Conference on Machine Learning, 3843-3853* (ICML), 2020. (Long Talk / Oral) [In Chapter 2]

(2) **Jiaxian Guo**, Jiachen Li, Mingming Gong Huan Fu, Kun Zhang and Dacheng Tao.
" Alleviating Semantics Distortion in Unsupervised Low-Level Image-to-Image Translation via Structure Consistency Constraint ", in *Computer Vision and Patter Recognition* (CVPR), 2022 (Poster) [In Chapter 3]

(3) **Jiaxian Guo**, Mingming Gong and Dacheng Tao.
" A Relational Intervention Approach for Unsupervised Dynamics Generalization in Model-Based Reinforcement Learning ", in *Tenth International Conference on Learning Representations* (ICLR), 2022 (Poster) [In Chapter 4]

(4) **Jiaxian Guo**, Mingming Gong, Yali Du, Zhen Wang, Dacheng Tao
"Hierarchical Prototypes for Unsupervised Dynamics Generalization in Model-Based Reinforcement Learning" , in *International Conference on Learning Representations* (ICLR), 2023 (In Submission) [In Chapter 5]

(5) **Jiaxian Guo**, Junnan Li, Dongxu Li, Anthony Tiong, Boyang Li, Dacheng Tao, Steven HOI
"From Images to Textual Prompts: Zero-shot VQA with Frozen Large Language Models" , in *Computer Vision and Patter Recognition* (CVPR), 2023 (Poster) [In Chapter 6]

Abstract

Deep learning methods have recently achieved remarkable success in various areas of artificial intelligence, such as computer vision and reinforcement learning, due to their potent distribution matching capabilities. However, these successes rely heavily on the i.i.d assumption, *i.e.*, the data distributions in the training and test datasets should be the same. In this way, current deep learning methods typically exhibit poor generalization under *distribution shift*, *i.e.*, they perform poorly on test data whose distribution differs from that of the training data. This significantly hinders the application of deep learning methods to real-world scenarios, as the distribution of test data is not always the same as the training distribution in this rapidly evolving world. For instance, image classification models in autonomous vehicles are incapable of predicting obstacles in atypical weather conditions, which can result in catastrophic accidents.

This thesis aims to discuss how to construct generalizable deep learning methods under a distribution shift. To achieve this, the thesis first models one prediction task as a structural causal model, which establishes the relationship between fine-grained variables using directed acyclic graphs. Among the distributions of variables in SCM, some of them are easily changed across domains while others are not. For example, dog images in cartoon and real-world styles may have different textures but similar shapes in their noses and ears. However, deep learning methods usually unconsciously mix up invariant variables and easily changed variables, and thus deviate the learned model from the true one, resulting in the poor generalization ability under distribution shift. To remedy this issue, we propose specific algorithms to model such an invariant part of the structural causal model with deep learning methods, and experimentally show it is beneficial for the trained model to generalize well into different distributions of the same task. Last, we further propose to identify and model the variant information in the new test distribution so that we can fully adapt the trained deep learning model accordingly.

We show the method can be extended for several practical applications, such as classification under *label shift*, image translation under *semantics shift*, robotics control in *dynamics generalization* and generalizing large language models into visual question-answer tasks. To fulfill the above applications, the thesis specifically proposes a variety of novel methods to model the invariant and variant parts of a structural causal model, including label transformation networks, color transformation consistency, and causal intervention methods. Furthermore, this thesis discusses recent cutting-edge research topics, including generative adversarial networks, contrastive learning, mutual information estimation and causal effect estimation.

Contents

Statement of Originality **ii**

Acknowledgements **iii**

List of Publications **v**

Abstract **vi**

Contents **viii**

List of Figures **xv**

Chapter 1 Introduction **1**

1.1 Contributions 5

Chapter 2 LTF: A Label Transformation Framework for Correcting Target Shift **8**

2.1 Introduction 9

2.2 Related Work 11

2.3 Methodology 12

2.3.1 Review of Previous Methods 12

2.3.2 Our Framework 13

2.3.2.1 Label Transformation Network 14

2.3.2.2 Label Influence Network 15

2.3.2.3 Distribution Matching 16

2.3.2.4 Shift Correction 18

2.4 Experiments 19

2.4.1 Discrete Target Shift Experiments 19

2.4.1.1 Shift Settings 19

2.4.1.2 Evaluation metrics and Results 20
2.4.2 Continuous Target Shift Experiments 22
2.4.2.1 Synthetic Data Experiment 23
2.4.2.2 Object 1D location Prediction Experiment 25
2.4.3 Multi-Dimensional Target Shift Experiments 28
2.5 Conclusion 29

Chapter 3 Alleviating Semantics Distortion in Unsupervised Low-Level Image-to-Image Translation via Structure Consistency Constraint 30

3.1 Introduction 30
3.2 Methodology 34
3.2.1 Motivation 35
3.2.2 Approximate Representation of Mutual Information 37
3.2.3 Full Objective 40
3.3 Experiments 41
3.3.1 Quantitative Evaluation 43
3.3.1.1 Digits Translation 43
3.3.1.2 Segmentation in Cityscapes 44
3.3.1.3 Maps 45
3.3.1.4 Simulation to Real: GTA to Cityscapes 46
3.3.2 Qualitative Evaluation 46
3.3.3 Sensitivity Analysis 48
3.4 Related Work 48
3.5 Conclusion 49

Chapter 4 A Relational Intervention Approach for Unsupervised Dynamics Generalization in Model-Based Reinforcement Learning 51

4.1 Introduction 52
4.2 Related Work 55
4.3 Method 56

4.3.1 Problem Setup 56
4.3.2 Overview 57
4.3.3 Relational Context Encoder 58
4.3.4 Interventional Prediction 60
4.4 Experiments 62
4.4.1 Enviromental Setup 62
4.4.2 Performance Comparisons with Baselines 64
4.4.2.1 Prediction Error Comparisons 64
4.4.2.2 Performance Comparisons 65
4.4.3 Ablation Study 66
4.5 Conclusion 67

Chapter 5 Hierarchical Prototypes for Unsupervised Dynamics Generalization in Model-Based Reinforcement Learning 68

5.1 Introduction 68
5.2 Related Work 71
5.3 Method 72
5.3.1 Problem setup 72
5.3.2 Hierarchical environment prototypes construction 74
5.3.3 Prototypical relational learning 76
5.3.4 Difference to RIA 78
5.4 Experiment 78
5.4.1 Environmental setup 78
5.4.2 Cluster visualization and analysis 80
5.4.3 Performance comparisons 81
5.4.4 Ablation study 83
5.5 Limitation 84
5.6 Conclusion 84

Chapter 6 From Images to Textual Prompts: Zero-shot VQA with Frozen Large Language Models 85

6.1 Introduction 86

6.2 Related Work 88

6.2.1 Recent Advances in VQA Methods 88

6.2.2 LLM for Zero/Few-Shot VQA Tasks 88

6.3 Method 89

6.3.1 Answer Extraction 90

6.3.2 Question Generation 91

6.3.3 Question-relevant Caption Prompt 92

6.3.4 Prompt Design 93

6.4 Experiment 94

6.4.1 Environment Setup 94

6.4.2 Main Results 95

6.4.3 Experimental Results of Different LLMs 98

6.4.4 Analysis on Question Generation Methods 98

6.4.5 Ablation on Caption Selection 100

6.4.6 Ablation Study on Prompt Design 100

6.4.7 Examples and Failure Case Analysis 100

6.5 Limitation 101

6.6 Conclusion 102

Bibliography **103**

Appendix A Appendix of Chapter 2 **133**

A0.1 Proof of Proposition 1 133

A0.2 Results of Fashion-MNIST and MNIST 134

A0.2.1 Results of CIFAR-10 134

A0.2.2 Results of Fashion-MNIST 135

A0.2.3 Results of MNIST 136

A0.3 Label Weights Visualization of Continuous Synthetic Data Experimens 139

A0.3.1 Results of Shift A 139

A0.3.2 Results of Shift B 142

A0.3.3 Results of Shift C 144

A0.3.4 Results of Shift D 146

A0.4 Results of dsprite Dataset 149

Appendix B Appendix of Chapter 3 150

B0.1 Details of Solving α 150

B0.2 Experimental Analysis 151

B0.2.1 β Analysis 152

B0.2.2 Generation Diversity Analysis 152

B0.2.3 Stability Analysis 153

B0.3 Experiments 153

B0.3.1 KID scores of Qualitative evluation 153

B0.4 Experimental Details 153

B0.4.1 Digits 153

B0.4.2 Cityscapes 153

B0.4.3 Maps 154

B0.4.4 Style Transfer 155

B0.5 Analysis on the Cat2Dog Dataset 156

B0.6 Generated Samples 157

B0.7 GTA → Cityscapes 157

B0.7.1 Maps 159

B0.7.2 Cityscapes 160

B0.7.3 Qualitative Results 160

B0.7.4 Digits 165

B0.7.5 Ablation Study 168

Appendix C Appendix of Chapter 4 169

C0.1 Environmental Settings 169

C0.2 Algorithm 169

C0.3 Training Details 170

C0.4 Network Details 170

C0.5 Connection between Relation Loss and Mutual Information 171
C0.6 Fair Comparison with TMCL 172
C0.7 Similarities Visualiaztion 173
C0.8 Prediction Errors on Traing Environments 174
C0.9 Prediction Errors on Test Environments 175
C0.10 Prediction Errors on Specified Environment 176
C0.11 The Average Returns on Test Environments during Training Process 176
C0.12 Quantitative Clustering Performance Comparison 177
C0.13 Visualization 178
C0.14 T-SNE Visualization 178
C0.15 PCA Visualization 178

Appendix D Appendix of Chapter 5 184

D1 Appendix 184
D1.1 Environmental Settings 184
D1.2 Algorithm 185
D1.3 Training Details 185
D1.4 Prediction Error 186
D1.5 Network Details 187
D1.6 Direct Causal Effect between Trajectory Prototypes 187
D1.7 Connection between Relation Loss and Mutual Information 188
D1.8 Visualization and Analysis 189

Appendix E Appendix of Chapter 6 191

E0.1 Details about Question-Relevant Caption Generation 191
E0.2 Experimental Results of Supervised Learning Methods in A-OKVQA 192
E0.3 Template-Based Question Design 192
E0.4 Sensitive Analysis 193
E0.5 Examples 194
E0.6 Details about Question-Relevant Caption Generation 194
E0.7 Experimental Results of Supervised Learning Methods in A-OKVQA 195

E0.8 Template-Based Question Design 195
E0.9 Sensitive Analysis 196
E0.10 Examples 196

List of Figures

2.1 The illustration of the LTF framework. Here we make an example which assumes that $X = Y + \epsilon$. Firstly, the **Label Transformation Model** LT transforms the training label distribution P_Y^S (blue one) to a new label distribution P_Y^R (green one), and then the **Label Influence Recovery Model** G generates the sample distribution P_X^R from the data generated from P_Y^R. By matching the target sample distribution P_X^T (red one) and P_X^R and fixing the G, the P_Y^R from LT is expected to be close to P_Y^T. As such, the target label distribution P_Y^T can be approximated by P_Y^R. 14

2.2 (a) Mean squared errors of estimated label weights (lower is better), (b) accuracy, and (c) F-1 score (higher is better) on CIFAR10 for uniform training set and random Dirichlet shifted test set, where the smaller *alpha* corresponds to larger shift. 21

2.3 (a) Mean squared errors of estimated label weights (lower is better), (b) accuracy, and (c) F-1 score (higher is better) on CIFAR10 for uniform training set and Tweak-One shifted test set, where *alpha* is the probability of tweaked class. 22

2.4 (a) Mean squared errors of estimated label weights (lower is better), (b) accuracy, and (c) F-1 score (higher is better) on CIFAR10 for uniform training set and minority-class shifted test set, where *alpha* is the ratio of minority classes. 22

2.5 (a) The illustration of Moon Synthetic Data (Shift C), where the generated two quarter circles training set as blue symbols show. (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 24

2.6 Illustration of the sprites dataset. This sprites in this dataset have 3 shapes(square, ellipse, heart), 6 scales values linearly spaced in [0.5, 1], 40 orientation values in [0, 2 pi], 32 X position values in [0, 10], 32 Y position values in [0, 10] 26

2.7 The prediction mean square error of 1D sprite position prediction (lower is better). (a) Random Dirichlet shift, where the smaller *alpha* corresponding to the bigger shift. (b) Large target shift, where *alpha* is the possibility of shifted label. (c) Minority shift, where *alpha* is the ratio of minority classes. 27

2.8 The prediction mean square error of 2D sprite position prediction (Lower is better). (a) Random Dirichlet shift, where the smaller *alpha* corresponds to larger shift. (b) Large target shift, where *alpha*is the possibility of a shifted label (c) Minority shift, where *alpha* is the ratio of minority classes. 28

3.1 Class distributions in GTA and Cityscapes. We can see that the ratio of the sky in GTA is significantly higher than it in Cityscapes, and thus the distribution matching based method has to translate the sky to vegetation/building to align the distributions. 31

3.2 The illustration about the inconsistent geometry structure translation causes the semantic-distortion problem in unsupervised low-level image translation. Visually, we can see that the geometry structures of the sky and human face are distorted during translation in CycleGAN, which causes the semantical distortion *e.g.*, sky to vegetation, a face without fringe to face with fringe. 32

3.3 Unsupervised image translation examples on GTA → Cityscapes. Portrait → Photo. The top row is the translated results by each method. The bottom row is the scatter plot of the pixel values in the input image x and its corresponding pixel value in the translated image $\hat{y}$, which shows the non-linear dependency of pixel values in two images. Obviously, the stronger the dependency between pixel values in the input image (X-axis) and the translated images (Y-axis), the better the geometry structure of the input image is maintained. MI stands for the mutual information estimated by our rSMI method. Specifically, the VGG refers to the Contextual loss [175] of VGG features. 35

3.4 An illustration of structure consistency constraint. The left figure shows that the pixel value in the input image x and its corresponding pixel value in the translated image $\hat{y}$ have strong non-linear dependencies, so we add the structure consistency constraint to model the dependencies of pixel values in two domain images. 39

3.5 Qualitative comparisons on SVHN→MNIST. From Figure (a) and (b), we can see that the GAN method has no collapse solution by combining with our SCC. Also, the semantics distortion problem in CycleGAN is alleviated after incorporating with SCC. 39

3.6 Unsupervised image translation examples on GTA → Cityscapes. The generated examples clearly show that our SCC can alleviate the semantic distortion problem *e.g.*, sky to tree/building in mainstream translation models. More examples are given at Appendix A.7 43

3.7 Qualitative results on Selfie → Anime, Portrait → Photo, Horse → Zebra datasets. More qualitative results are given in A.7.3. We can see that the no matter personal identification or horse shape is better preserved by the translation model empowered by our SCC. 45

3.8 Sensitivity analysis examples on Selfie → Anime and GTA → Cityscapes. Obviously, the semantics distortion problem in CycleGAN is alleviated after incorporating with our SCC. 47

4.1 (a) The illustration of why historical states and actions are encoded in environment-specified factor Z, (b)(c)(d) The PCA visualizations of estimated context (environmental-specific) vectors in **Pendulum** task, where the dots with different colors denote the context vector (after PCA) estimated from different environments. More visualization results are given at Appendix C0.13. 53

4.2 An overview of our Relational Intervention approach, where Relational Encoder, Prediction Head and Relational Head are three learnable functions, and the circles denote states (Ground-Truths are with red boundary, and estimated states are with black boundary), and the rectangles denote the estimated vectors. Specifically, $prediction\ Loss$ enables the estimated environmental-specified factor can help the Prediction head to predict the next states, and the $relation\ Loss$ aims to enforce the similarity between factors estimated from the same trajectory or environments. 57

4.3 (a) The illustration of causal graph, and the red line denotes the direct causal effect from Z to S_{t+1}. (b) The illustration of estimating the controlled causal effect. 59

4.4 The average prediction errors of dynamics models on training environments during training process (over three times). Specifically, the x axis is the training timesteps and y axis is the log value of average prediction prediction errors. More figures are given at Appendix C0.8. 65

4.5 The average rewards of trained model-based RL agents on unseen test environments. The results show the mean and standard deviation of returns averaged over three runs. The fair comparison between TMCL (no adaptation) and our method can be found in Appendix C0.6 66

4.6 (a) The average rewards of trained model-based RL agents on unseen environments. The results show the mean and standard deviation of returns averaged over three runs. (b) The average prediction errors over the training procedure. Prediction errors on test environments are given in Appendix C0.9 67

5.1 An overview of our Hierarchical Prototypical Method, where the context encoder estimates the environmental-specific factor $\hat{z}$ and environments includes four ants with different destroyed leg with red color. Items extracted from different environments are different colors. We construct prototypes for each trajectory and environment, and denote them as diamond and star, respectively. Each estimated $\hat{z}$ are optimized with its corresponding trajectory and environment prototype using our prototypical relational learning as dotted line shows. 72

5.2 (a) The illustration of causal direct effect estimation between two trajectory prototypes, where we calculate the mean difference over a batch of predicted next states as the similarity w of the two prototypes. (b) The illustration constructed the similarity matrix between trajectory prototypes. We use the mean of top_k (denoted by the red color) similar trajectory prototypes as the corresponding environmental prototypes. 75

5.3 The PCA visualization of environmental-specific factors estimated by TMCL [219], CaDM [135], RIA [70] and ours on the Half-cheetah (upper part) and Pendulum (lower part) task. 80

5.4 The average returns of model-based RL agents on unseen test environments. The results show the mean and standard deviation of returns averaged over five runs. Specifically, we use the no adaptation version of TMCL for a fair comparison. The performance comparisons on dynamics prediction errors are given at Appendix D1.4. 82

5.5 Left Image: The sensitivity analysis about how many trajectory prototypes should be combined into environmental prototypes. Right Two Images: The similarity metrics used in combining trajectory prototypes into environmental prototypes. 83

6.1 The illustrative comparison of three types of methods that enable LLM to perform VQA tasks, where blue block denotes that the inner parameters are frozen while pink block indicates the inner parameters are trainable. 89

6.2 The overall pipeline of Img2Prompt, including Caption Prompt and Exemplar Prompt generation. 91

6.3 Example predictions made by Img2LLM. Specifically, (a) and (b) are successful cases, while (c) and (d) are failure cases. See more examples at Appendix A.5. 101

A.1 (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score (Higher is better) on FASHION-MNIST for uniform training set and random Dirichlet shifted test set, where smaller *alpha* corresponds to bigger shift. 135

A.2 (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score (Higher is better) on FASHION-MNIST for uniform training set and Tweak-One shifted test set, where *alpha* is the probability of tweaked class. 136

A.3 (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score (Higher is better) on FASHION-MNIST for uniform training set and minority-class shifted test set, where *alpha* is the ratio of minority classes. 136

A.4 (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score (Higher is better) on MNIST for uniform training set and random Dirichlet shifted test set, where smaller *alpha* corresponds to bigger shift. 137

A.5 (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score (Higher is better) on MNIST for uniform training set and Tweak-One shifted test set, where *alpha* is the probability of tweaked class. 138

A.6 (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score on MNIST for uniform training set and minority-class shifted test set, where *alpha* is the ratio of minority classes. 138

A.7 (a) The illustration of Moon Synthetic Data (Shift A, 1st experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 139

A.8 (a) The illustration of Moon Synthetic Data (Shift A, 2nd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 140

A.9 (a) The illustration of Moon Synthetic Data (Shift A, 3rd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 140

A.10 (a) The illustration of Moon Synthetic Data (Shift A, 4th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 141

A.11 (a) The illustration of Moon Synthetic Data (Shift A, 5th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 141

A.12 (a) The illustration of Moon Synthetic Data (Shift B, 1st experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 142

A.13 (a) The illustration of Moon Synthetic Data (Shift B, 2nd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 142

A.14 (a) The illustration of Moon Synthetic Data (Shift B, 3rd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 143

A.15 (a) The illustration of Moon Synthetic Data (Shift B, 4th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 143

A.16 (a) The illustration of Moon Synthetic Data (Shift B, 5th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 144

A.17 (a) The illustration of Moon Synthetic Data (Shift C, 1st experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 144

A.18 (a) The illustration of Moon Synthetic Data (Shift C, 2nd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 145

A.19 (a) The illustration of Moon Synthetic Data (Shift C, 3rd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 145

A.20 (a) The illustration of Moon Synthetic Data (Shift C, 4th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 146

A.21 (a) The illustration of Moon Synthetic Data (Shift C, 5th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 146

A.22 (a) The illustration of Moon Synthetic Data (Shift D, 1st experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 146

A.23 (a) The illustration of Moon Synthetic Data (Shift D, 2nd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 147

A.24 (a) The illustration of Moon Synthetic Data (Shift D, 3rd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 147

A.25 (a) The illustration of Moon Synthetic Data (Shift D, 4th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 148

A.26 (a) The illustration of Moon Synthetic Data (Shift D, 5th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth. 148

B.1 The training curves and the sensitive analysis about β on Digits datasets 151

B.2 The generation example of MUNIT+SCC on the edge2shoes. Specifically, images at first two rows are source domain images and the others are translated images by MUNIT+SCC. 152

B.3 The generation example of MUNIT on the edge2shoes. Specifically, images at first two rows are source domain images and the others are translated images by MUNIT. 152

B.4 Qualitative results on a geometry-variant dataset, including Dog → Cat. Images at the top row are successful cases, while images at the bottom row are failure cases. 157

B.5 The overlarge λ_{SCC} example on SVHN→MNIST. 168

C.1 The estimated similarities between $\hat{Z}$s (learned by the model without Intervention module) from different environments and anchors (mass=1) on different tasks, where the box represent the range of 50% samples, the line in the middle of a box denotes the average similarity and the top/bottom lines denote the max/min similarities. 173

C.2 The estimated similarities between $\hat{Z}$s (learned by the model with Intervention module) from different environments and anchors (mass=1) on different tasks, where the box represent the range of 50% samples, the line in the middle of a box denotes the average similarity and the top/bottom lines denote the max/min similarities. 174

C.3 The estimated similarities between $\hat{Z}$s (learned by the model **without** Intervention module) from different environments and anchors (mass=1) on different tasks, where the box represent the range of 50% samples, the line in the middle of a box denotes the average similarity and the top/bottom lines denote the max/min similarities. 174

C.4 The average prediction errors of dynamics models on training environments during training process (over three times). Specifically, the x axis is the training timesteps and y axis is the log value of average prediction prediction errors. More figures are given at Appendix C0.8. 175

C.5 The average prediction errors of dynamics models on test environments during training process (over three times). Specifically, the x axis is the training timesteps and y axis is the average log value of prediction prediction errors. 175

C.6 The average rewards of trained model-based RL agents on unseen environments. The results show the mean and standard deviation of returns averaged over three runs. 177

C.7 The T-SNE visualization of estimated context (environmental-specific) vectors in the **Pendulum** task, where mass = 0.5 and mass =1.3 are from test environments. 179

C.8 The T-SNE visualization of estimated context (environmental-specific) vectors in the **Halfcheetah** task, where mass = 0.5 and mass =1.5 are from test environments. 180

C.9 The PCA of estimated context (environmental-specific) vectors in **Pendulum** task, where mass = 0.5 and mass =1.3 are from test environments. 181

C.10 The PCA of estimated context (environmental-specific) vectors in **HalfCheetah** task, where mass = 0.5 and mass =1.5 are from test environments. 182

C.11 The PCA of estimated context (environmental-specific) vectors in the **Hopper** task. 182

C.12 The PCA of estimated context (environmental-specific) vectors in the **Cripple_Halfcheetah** task. 182

C.13 The PCA of estimated context (environmental-specific) vectors in the **Slim_Humanoid** task. 183

D.1 The average prediction errors of model-based RL agents on training environments. The results show the mean and standard deviation of prediction errors averaged over five runs. Specifically, we use the no adaptation version of TMCL for a fair comparison. 186

D.2 The PCA visualization of environmental-specific factors estimated by TMCL [219], CaDM [135], RIA [70] and ours on the Pendulum task. 189

D.3 The PCA visualization of environmental-specific factors estimated by TMCL [219], CaDM [135], RIA [70] and ours on the Halfcheetah task. 189

D.4 The PCA visualization of environmental-specific factors estimated by TMCL [219], CaDM [135], RIA [70] and ours on the slim_humanoid task. 190

D.5 The PCA visualization of environmental-specific factors estimated by TMCL [219], CaDM [135], RIA [70] and ours on the swimmer task. 190

E.1 Success case analysis for OK-VQA. Green color indicates answer cues and correct prediction. 198

E.2 Failure case analysis for OK-VQA. Red color indicates incorrect prediction. 199

E.3 Success case analysis for A-OKVQA. Green color indicates answer cues and correct prediction. 200

E.4 Failure case analysis for A-OKVQA. Red color indicates incorrect prediction. 201

E.5 Success case analysis for VQAv2. Green color indicates answer cues and correct prediction. 202

E.6 Failure case analysis for VQAv2. Red color indicates incorrect prediction. 203

CHAPTER 1

Introduction

In the past decades, deep learning methods [131] have achieved great success in many areas of artificial intelligence, such as computer vision [81, 189], natural language processing [297, 319], multi-modal learning [88] and reinforcement learning [226, 227]. The success of deep learning methods is contingent on two crucial factors: 1) It has been mathematically demonstrated that neural networks with infinite hidden neurons in deep learning can be used as universal approximation functions [84, 77, 162, 112], and experimentally demonstrate that the neural networks with larger size parameters [25, 80, 121] have stronger function approximation ability than those with smaller size parameters, which provides them with strong data distribution matching capabilities. 2) In the Internet age, it is feasible to collect large amounts of annotated data for training neural networks, and these datasets can better represent and reflect the statistics of real data than smaller-scale datasets [131]. In this way, deep learning methods are able to approximate the real data distribution with their strong distribution matching ability and achieve human-surpassing performance under the *i.i.d* assumption, *i.e.*, the test data distribution is identical and independent of the training data distribution. For example, recent large language models, such as GPT-3 [25] with 175 billion parameters, trained on massive texts with over 400 billion tokens, have demonstrated surprisingly superior performance over traditional machine learning models in NLP tasks.

However, current deep learning methods still suffer from *distribution shift* problem [156, 155, 154, 147, 91, 146, 8, 72], *i.e.*, they typically perform poorly on certain domains with different distributions than the training data. This has become a major obstacle to the deployment of deep learning techniques in real-world applications, as test data

in real-world scenarios do not always conform to the distribution of training data. The reasons for the distribution shift in real-world applications mainly lie in the following:
(1). The data in some domains is limited or expensive costly to obtain, making it difficult to collect the large-scale dataset for training the deep learning methods. In this way, the feasible solution [280, 37] is to collect data in other domains as the training dataset, and expect the trained deep learning model to be able to generalize well in data with different distributions. For example, because of privacy concerns and labelling difficulties, publicly labelled medical images for diagnosis are usually limited, *e.g.*, the public chest X-Ray dataset [245] for COVID diagnosis only contains images from 517 patients. This is obviously insufficient to train a deep learning model, so recent methods [317, 79] tried to train a generalizable deep learning method from the Typical Pneumonia dataset rather than using only COVID dataset.
(2). In this rapidly evolving world, test data distributions for many tasks may not always remain unchanged. Taking disease prediction as an example, our goal is to predict disease Y from symptoms X, the distribution of the disease can change over location and time. Consider the flu prediction task, the data available for flu prediction is always has a regular morbidity rate, but if a model is trained on these data, the performance of this model will decrease when it is used to detect flu in a location or over a period with a high morbidity rate when flu outbreak [252].
As stated in the preceding two reasons, *distribution shift* problem is both a natural occurrence and a significant barrier to the deployment of deep learning techniques in real-world applications.

In this thesis, we mainly discuss how to adaptively improve the generalization ability of deep learning methods under distribution shift with the thought of causality [196], making deep learning methods more practical for real-world applications. We will first summarise the main idea for solving the distribution shift problem, and then briefly introduce how to extend our idea into several practical applications, such as classification under *label shift*, image translation under *semantics shift*, robotics control under *dynamics generalisation*, and generalising large language models into visual question-answer tasks.

In order to develop the generalizable deep learning model under distribution shift, we assume that data distributions from different domains under the same task are independent and identically *(i.i.d)* drawn from a “mother” distribution [310]. This “mother” distribution typically contains many fine-grained variables, but some of which are easy to change across domains, and we call such easily changeable variables as domain-dependent variables for convenience. For example, dog images in the cartoon style have a smoother texture than their real-world counterparts, resulting in a different distribution in the cartoon image compared to the real-world image. In contrast, even if the data distributions differ across domains due to domain-dependent variables, they should share some variables whose distributions remain unchanged in the same task. For instance, the ears, nose, and walking posture of dogs depicted in cartoons and in real-world styles are comparable. However, deep learning methods usually unconsciously mix up variant variables and invariant variables, and learn a deviated prediction function for the prediction task [240]. For example, deep learning models may rely on the fur texture instead of the shape of the body to predict the species of the animal, resulting in poor performance in the new domains.

To address this issue, this thesis aims to model such “mother" distribution [310] using a structural causal model (SCM) [196], which establishes the relationship between fine-grained variables using directed acyclic graphs [198, 120]. The SCM framework allows us to construct a model that is based on our understanding of the physical world, where causal relationships between variables are explicitly encoded. For instance, consider a scenario where we wish to model the relationship between the altitude and average temperature of a city. Intuitively, we know that the altitude of a city plays a critical role in determining its average temperature. Therefore, we can encode this causal relationship into an SCM with the directed edge $X \rightarrow Y$, where X represents the altitude of the city, and Y represents the average temperature. The directionality of the edge indicates that the altitude of the city causally influences its average temperature, rather than the other way around. SCMs provide an interpretable and concise method for modelling causal relations between variables and factorizing the "mother" distribution within a single task. Using the given SCM [196], we can easily identify domain-invariant variables in this task according to our understanding of the world, and then model their information using deep learning techniques. Experimentally, we demonstrate in our thesis that this reduces the discrepancy between the ground-truth data distribution and the modelled distribution, thereby enhancing the generalisation ability of deep learning models [41]. In addition, we further model the information of domain-dependent variables using deep learning techniques by exploiting the domain-invariant information with the known structural causal model, allowing us to fully generalize the learned model to a new data distribution from unknown domains.

Following this idea, we have several questions to answer. First, the domain-invariant information varies across different tasks and settings, how to find and model such domain-invariant information across domains with deep learning methods given a structural causal model? Second, how can we estimate domain-dependent variables in a structural causal model? Third, how can we make use of such learned information to make generalizable predictions in new domains?

To answer these questions, this thesis proposes a mutual information based method to model domain-invariant information in the image translation task [83, 321] in Chapter 3. In Chapter 2, we further demonstrate that domain-invariant information is beneficial for learning domain-dependent information given a structural causal model, and can further improve the generalization ability of the deep learning models. In Chapter 4 and Chapter 5, we propose a causal inference-based method to automatically learn the domain-dependent information without access to the data from new domains, so that the learned model can automatically generalize to data from new domains. In Chapter 6, we propose a new method in which domain-dependent information can be denoted using natural language, allowing us to interpretably generalise the deep learning method to new domains.

1.1 Contributions

The works in my PhD try to develop generalizable deep learning models, and my contributions are summarized as follows, organized in different chapters.

Chapter 2: In this chapter, we focus on the *label shift problem* whose structural causal model is very simple, where the label Y is the cause of data X and assume that the label distribution P_Y changes but the conditional distribution $P_{X|Y}$ stays the same [311, 91, 146, 8]. To address this problem, we propose the Label Transformation Framework (LTF) which leverages the generative adversarial network-based generative methods [61] to model the invariant distribution $P_{X|Y}$, and use the adversarial training to learn the changed label distribution P_Y. The model performed significantly better than the original after fine-tuning the deep learning models with the new label distribution P_Y. Given the structural causal model, the framework demonstrates that domain-invariant information facilitates the acquisition of domain-dependent information. This is also the first framework that can handle both discrete and continuous label shift problems, *e.g.* classification and regression tasks, as well as high-dimensional label shift problems.

Chapter 3: In this chapter, we focus on *semantic shift* problem in the unpaired image translation task. This task aims to translate an image in the source domain $\mathcal{X}$ properly to the target domain $\mathcal{Y}$ without the paired data. However, In the majority of unpaired datasets, not only the domain information but also the underlying semantic distributions vary between source and target datasets [95]. Thus, previous methods [321, 51] fail to preserve the semantics that is domain-invariant during the image translation. Typically, they incorrectly translate the digit 3 in an image to the digit 1, distorting the image's semantics. To address this issue, we propose a mutual information-based method to preserve domain-invariant information during image translation. The method is quite novel and can significantly improve the task of image translation's semantic consistency.

Chapter 4: In this chapter, we introduce a causal intervention method aiming to learn the underlying domain-dependent information. Specifically, we focus on the generalization of model-based reinforcement learning (MBRL) methods, which aims to train an agent that is able to generalize well into environments with unseen transition dynamics. To achieve this, we want to estimate the domain-dependent information for each environment. However, because environments are not labelled, the extracted domain-dependent information inevitably contains redundant information unrelated to the dynamics in transition segments and thus loses the environmental semantics. As a result, the learned dynamics prediction function will deviate from the true one, which undermines the generalization ability. To tackle this problem, we introduce an intervention prediction module to estimate the causal effect of extracted domain-dependent information based on the structural causal model, and thus encode more semantics into them. The environmental results demonstrate that domain-dependent information estimated by our method is more semantically meaningful than previous methods, and can significantly reduce dynamics prediction errors and improve the performance of model-based RL methods on zero-shot new environments with unseen dynamics.

Chapter 5: In this chapter, we introduce a hierarchical prototypical method for the generalization of model-based reinforcement learning (MBRL) methods mentioned in the last chapter. The proposed hierarchical prototypical method is able to cluster the domain-dependent information in a tight way, and thus can learn more semantic information than previous methods. The environmental results demonstrate that learned domain-dependent information can significantly improve the performance of model-based RL methods on zero-shot new environments with unseen dynamics.

Chapter 6: In this chapter, we introduce a prompt design method for generalizing large language models into a visual-question answer (VQA) task. Large language models have shown strong reasoning ability in NLP tasks [25], but how to leverage its strong reasoning ability into VQA tasks is a very interesting and challenging problem. This is because there are the modality disconnection and the task disconnection between large language models into visual-question answer tasks, where the modality disconnection refers to LLMs do not natively process images and encoding visual information into a format that LLMs can process, and the task disconnection refers to LLMs are usually pre-trained using generative [25] or denoising objectives [42] on language modelling tasks. As the LLMs are unaware of the tasks of question answering or VQA, they often fail to fully utilize contextual information in generating the answers. To remedy these issues, we introduce a prompt design method called Img2Prompt, which is able to reduce the modality discrepancy and task discrepancy between the large language model and VQA task by converting the image information as prompts of the large language model. Img2Prompt achieves comparable or better performance than methods relying on end-to-end training. On the challenging A-OKVQA dataset, our method outperforms some few-shot methods by as much as 20%. Img2Prompt provides an interpretive way (nature language) for generalizing large language models into new tasks, which may inspire more future works in this area.

CHAPTER 2

LTF: A Label Transformation Framework for Correcting Target Shift

Distribution shift is a major obstacle to the deployment of current deep learning models on real-world problems. Let Y be the target (label) and X the predictors (features). We focus on one type of distribution shift, *target shift*, where the marginal distribution of the target variable P_Y changes, but the conditional distribution $P_{X|Y}$ does not. Existing methods estimate the density ratio between the source- and target-domain label distributions by density matching. However, these methods are either computationally infeasible for large-scale data or restricted to shift correction for discrete labels. In this Chapter, we propose an end-to-end Label Transformation Framework (LTF) for correcting target shift, which implicitly models the shift of P_Y and the conditional distribution $P_{X|Y}$ using neural networks. Thanks to the flexibility of deep networks, our framework can handle continuous, discrete, and even multi-dimensional labels in a unified way and is scalable to big data. Moreover, for high dimensional X, such as images, we find that the redundant information in X severely degrades the estimation accuracy. To remedy this issue, we propose to match the distribution implied by our generative model and the target-domain distribution in a low-dimensional feature space that discards information irrelevant to Y. Both theoretical and empirical studies demonstrate the superiority of our method over previous approaches.

2.1 Introduction

Standard supervised learning methods typically assume that the training set (source domain) and the test set (target domain) have the same distribution. However, the data available for training is always limited and may not represent and reflect the statistics of the test data. As such, the source-domain distribution P^S_{XY} is often different from the target-domain distribution P^T_{XY}, degrading the performance of the models learned on the training set. This phenomenon is called *distribution shift*, which has become a significant obstacle to the deployment of deep learning models in the real world.

To overcome distribution shift and improve the prediction on test data, existing methods have studied various distribution shift settings, among which *covariate shift* and *target shift* have been widely considered. Covariate shift assumes that the marginal P_X changes across training and test sets, whereas the conditional distribution $P_{Y|X}$ is invariant [221, 239, 67, 156, 155, 154, 147]. Target shift assumes that the label distribution P_Y changes but the conditional distribution $P_{X|Y}$ stays the same [311, 91, 146, 8].

Here we focus on the target shift problem since it appears in a wide range of real-world learning problems. For example, in disease prediction, where our goal is to predict disease Y from symptoms X, the distribution of the disease can change over location and time, while the mechanism of symptoms $P_{X|Y}$ is rather stable. Consider the flu prediction task, the data available for flu prediction is always has a regular morbidity rate, but if a model is trained on these data, the performance of this model will decrease when it is used to detect flu in a location or over a period with a high morbidity rate [252]. In addition, target shift also exists in many computer vision applications, such as predicting object locations [290] and direction and human poses [173]. The distribution of object locations or human poses often changes across training and test sets.

Despite being a natural phenomenon in many real applications, target shift is relatively understudied compared to covariate shift. [29] proposed an expectation-maximization algorithm that requires estimation of the conditional distribution $P_{X|Y}$. Unfortunately, estimating $P_{X|Y}$ is difficult for high-dimensional X and moreover, it does not apply to regression problems. [311] proposed a nonparametric method to estimate the density ratio P_Y^T/P_Y^S by kernel mean matching of distributions, which applies to both regression and classification problems. However, this approach is not compatible with large data as the computational cost is quadratic in the sample size. Recently, [146, 8] proposed efficient and sample size-independent methods that make use of the confusion matrix of a classifier learned on the training set. These methods have shown promising performance on large-scale data but are only applicable to classification problems.

In this Chapter, we aim to propose a new framework that can correct target shift for both discrete and continuous Y. Compared to existing methods, we make the following contributions. First, instead of estimating the density ratio P_Y^T/P_Y^S, we model the change in the distribution P_Y by a neural label transformation T, which transforms the training label distribution P_Y^S to a new label distribution P_Y^R that can approximate the unknown P_Y^T in the test set. Thanks to the flexibility of neural nets, we can design different transform models T to deal with different types of Y, including discrete, continuous, and even multi-dimensional labels. Second, because of the absence of labels in the test set, we model the invariant conditional distribution $P_{X|Y}$ using a conditional generator G on the training set. By concatenating the label transformation model T with the conditional generator G, we can generate corresponding sample distribution P_X^R, which is then matched with P_X^T to estimate the parameters in T. Third, for high dimensional X, such as images, we observe that the redundant information significantly degrades the estimation accuracy. To remedy this issue, we theoretically analyze this phenomenon and propose to match the distributions of a feature representation of X that discards the information irrelevant to Y.

To demonstrate the advantage of our framework in practical applications, we apply our method to a range of label types, including classification (discrete label), regression (continuous label) and objects 2D object position prediction (multi-dimension label), in various target shift settings, such as random target shift, high probability label quantification and low probability label quantification). The empirical results demonstrate the generality, flexibility and superiority of our framework compared to previous methods.

2.2 Related Work

Covariate shift and *target shift* are two common types of *distribution shift*. The former one assumes that the feature distribution P_X changes over training set and test set, but the conditional distribution $P_{Y|X}$ from label to data remains unchanged, while the latter one assumes that the label distribution P_Y changes but $P_{X|Y}$ is invariant.

The existing methods solve *covariate shift* and *target shift* using re-weighting methods, which are also used in a wide range of problems, *e.g.*, label-noise [152, 300, 31, 48] . We firtstly introduce methods dealing with *covariate shift* problems shortly, where many methods estimate importance sample weights P^T_X/P^S_X [303, 87, 239, 67] via kernel methods [87, 67, 311] or using a discriminative classifier [157, 151]. Then they correct models by retraining a new model with re-weighted training samples using estimated P^T_X/P^S_X under the ERM framework [221]. More recent works learn domain-invariant representations $X' = h(X)$ that have similar marginal distributions across domains ($P^T_{X'} \approx P^S_{X'}$) [223, 191, 9, 258, 53, 156].

Similar to the correction of *Covariate shift*, there are two major steps to solve *target shift* problems. The first step is to estimate the label distribution P^T_Y in the target domain or the ratio P^T_Y/P^S_Y. The second step is to construct an unbiased estimate of the target domain risk based on the results from the first step. [311, 91, 188, 60] proposed to estimate P^T_Y or P^T_Y/P^S_Y by matching a weighted combination of conditionals $P^S_{X|Y}$ in the source domain the marginal distribution P^T_X in the target domain. The matching of distributions is achieved by minimizing suitable divergence measures [66, 236] w.r.t. the weights on $P^S_{X|Y}$. In the discrete Y scenario, [146] proposed a method which estimates the importance weight (P^T_Y / P^S_Y) by matching the output of trained classifier on the training set (confusion matrix), and then [8] turned this problem as a linear programming problem and iteratively minimized the error of label distributions between the training set and the test set, improving the accuracy of estimated target label distribution P^T_Y. In addition, [8] added a regularization term to make the algorithm compatible with the situation where the target sample size is small.

2.3 Methodology

Given training data $\mathcal{D}_s = \{x_i^s, y_i^s\}_{i=1}^{n_s} \subseteq \mathcal{X} \times \mathcal{Y}$ independently drawn from an unknown joint distribution P_{XY}^S, denoted as the source domain distribution, and test data $\mathcal{D}_t = \{x_i^t, y_i^t\}_{i=1}^{n_t}$ drawn from the target-domain distribution P_{XY}^T, where y_i^t is unknown, target shift assumes that $P_{X|Y}^S = P_{X|Y}^T = P_{X|Y}$ and $P_Y^S \neq P_Y^T$. Our goal is to build a model to estimate the label distribution P_Y^T in the target domain such that we can correct the label shift between the training and test data and thus improve the prediction performance on the test set. We consider both continuous Y, i.e., $\mathcal{Y} = \mathbb{R}^d$, and discrete Y, i.e., $\mathcal{Y} = \{1, \dots, K\}$.

2.3.1 Review of Previous Methods

To estimate the label distribution P_Y^T, existing methods use the relation between source and target distributions:

$$P_X^T(x) = \int_y P_{X|Y}(x|y) P_Y^T(y) dy \tag{2.1}$$

$$= \underbrace{\int_y P_{XY}^S(x,y) \frac{P_Y^T(y)}{P_Y^S(y)} dy}_{P_X^{\text{new}}} . \tag{2.2}$$

Because P_{XY}^S and P_X^T can be estimated from $\mathcal{D}_s$ and $\mathcal{D}_t$, previous methods [311, 60] estimate the density ratio $\beta^*(y) = \frac{P_Y^T(y)}{P_Y^S(y)}$ by minimizing the empirical Maximum Mean Discrepancy (MMD) [66] between P_X^T and P_X^{new}:

$$||\frac{1}{n_t}\sum_{i=1}^{n_t} \psi(x_i^t) - \frac{1}{n_s}\sum_{i=1}^{n_s} \beta(y_i^s)\psi(x_i^s)||_{\mathcal{H}}^2, \tag{2.3}$$

$$s.t. \ \beta(y_i^s) \geq 0, \text{ and } \sum_{i=1}^{n_s} \beta(y_i^s) = n_s, \tag{2.4}$$

where ψ is the feature mapping from $\mathcal{X}$ to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ associated with a kernel function $k(x, x') = \langle \psi(x), \psi(x') \rangle_{\mathcal{H}}$. For kernel functions that have no explicit ψ, for example, RBF kernels, we need to use kernel trick to calculate

(2.3). The computational cost is quadratic in the sample size and thus the algorithm is not scalable to large datasets.

When Y is discrete, recent works [146, 8] proposed to estimate $\boldsymbol{\beta} = [\beta(y = 1), \ldots, \beta(y = K)]^T$ by using the confusion matrix of a classifier f:

$$\hat{\mathbf{q}} = \hat{\mathbf{C}}\hat{\boldsymbol{\beta}}, \tag{2.5}$$

where $\hat{\mathbf{C}}$ is the confusion matrix with each element $\hat{C}_{ij} = \frac{1}{n_s}\sum_{k=1}^{n_s} \mathbb{1}\{f(x_k^s) = i, y_k^s = j\}$ and $\hat{\mathbf{q}}_i = \frac{1}{n_t}\sum_{j=1}^{n_t} \mathbb{1}\{f(x_j^t) = i\}$. It can be seen that (2.5) corresponds to a specific form of (2.3) in which the feature mapping ψ is set to $\psi(x) = one\_hot(f(x))$, where one_hot is a function mapping $\hat{y} = f(x)$ to its corresponding one-hot vector. Because the dimensionality of $\psi(x)$ is simply the number of classes K, which is usually much smaller than the sample size, $\hat{\boldsymbol{\beta}}$ can be obtained efficiently. However, this type of methods only work for discrete labels.

2.3.2 Our Framework

Instead of estimating the density ratio $\beta(y)$, our framework estimates the target-domain marginal distribution using a constructed distribution P_X^R defined as follows:

$$\begin{aligned} P_X^R &= \int_{y^r} P_{X|Y}(x|y^r) P_Y^R(y^r) dy^r \\ &= \int_{y^r} P_{X|Y}(x|y^r) \int_{y^s} P_{Y^R|Y^S}(y^r|y^s) P_Y^S(y^s) dy^s dy^r, \end{aligned} \tag{2.6}$$

where we build a new label distribution P_Y^R by transforming the training label distribution P_Y^S using the transition model $\int_{y^s} P_{Y^R|Y^S}(y^r|y^s) P_Y^S(y^s) dy^s$. Because Y is not observed in the test domain, we need to estimate the label transition model by comparing P_X^R and P_X^T. In the following sections, we will show how the transformation between P_Y^S and P_Y^T can be estimated from the labeled training set and unlabeled test set.

Figure 2.1 displays the flowchart of our framework. First, we transform the samples drawn from P_Y^S using the **Label Transformation** network LT which maps P_Y^S to a distribution P_Y^R. Because there are only unlabeled data in the target domain, we cannot directly match P_Y^R with the target-domain label distribution P_Y^T. Therefore, we then pass the transformed labels into the **Label Influence Recovery** network G, which models the conditional distribution $P_{X|Y}$ implicitly, to generate samples with distribution P_X^R. Finally, we match the generated distribution P_X^R with the target domain P_X^T to estimate the parameters in the label transformation network, such that P_Y^R can approximate the target-domain label distribution P_Y^T. After estimating P_Y^R, we can train an unbiased classifier for prediction in the target domain. In the following, we present the details of each component in our framework.

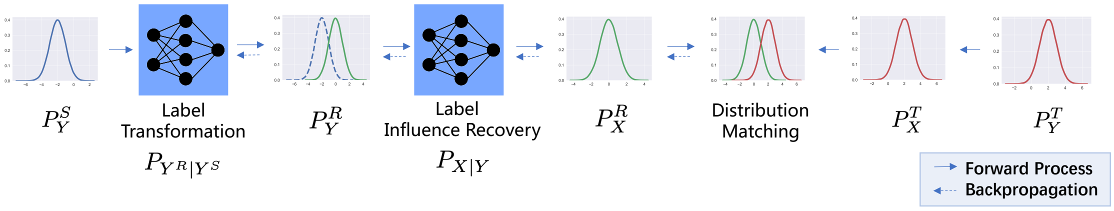


FIGURE 2.1. The illustration of the LTF framework. Here we make an example which assumes that $X = Y + \epsilon$. Firstly, the **Label Transformation Model** LT transforms the training label distribution P_Y^S (blue one) to a new label distribution P_Y^R (green one), and then the **Label Influence Recovery Model** G generates the sample distribution P_X^R from the data generated from P_Y^R. By matching the target sample distribution P_X^T (red one) and P_X^R and fixing the G, the P_Y^R from LT is expected to be close to P_Y^T. As such, the target label distribution P_Y^T can be approximated by P_Y^R.

2.3.2.1 Label Transformation Network

Here we use a neural network LT to transform the training label distribution P_Y^S to a new label distribution P_Y^R, such that we can directly generate the corresponding sample distribution P_X^R together with one generator G that models $P_{X|Y}$. Specifically, we use the following functional model:

$$Y^R = LT(Y^S, Z), \tag{2.7}$$

where LT is modeled by a neural net and Z is a random variable with distribution P_Z. (2.7) models the conditional distribution $P_{Y^R|Y^S}$ implicitly. Because $P_Y^R = \int_{y^s} P_{Y^R|Y^S}(y^r|y^s)P_Y^S(y^s)dy^s$, we can sample $y_i^r \sim P_Y^R$ by first sampling y_i^s from the source-domain labels, and then generate the corresponding $y_i^r = LT(y_i^s, z_i)$, where $z_i \sim P_Z$. Note that in some situations, such as discrete Y, it might be more convenient to directly use the parametric form of $P_{Y^R|Y^S}$.

If labeled data were available in the target domain, we can then simply match the empirical P_Y^R and P_Y^T to learn LT. Unfortunately, target-domain labels are not available in unsupervised domain adaptation, but still, in the target domain we have unlabeled data $\{x_i^t\}_{i=1}^{n_t}$, which can be used to estimate LT. To this end, we need to transform P_Y^R to a distribution P_X^R in the $\mathcal{X}$ space. Because P_X^R captures the influence of P_Y^R, we can possibly estimate P_Y^R (or LT) by matching P_X^R and P_X^T, from which we can sample data points to estimate and minimize their distance.

2.3.2.2 Label Influence Network

In order to transform P_Y^R to P_X^R, we make use of the following model:

$$X^R = G(Y^R, E), \tag{2.8}$$

where G is a neural generator, and E is a random variable with distribution P_E, which is set to normal distribution. We can use (2.8) to implicitly model $P_{X|Y}$. Due to $P_X^R = \int_{y^r} P_{X^R|Y^R}(x^r|y^r)P_Y^R(y^r)dy^r$, we can sample $x_i^r \sim P_X^R$ by first sampling y_i^r using (2.7), and then generate the corresponding $x_i^r = G(y_i^r, e_i)$, where $e_i \sim P_E$.

Since G corresponds to the generator in a conditional generative adversarial network [179, 180, 61], we can learn it from the source domain data $\mathcal{D}_s = \{x_i^s, y_i^s\}_{i=1}^{n_s}$ by adversarial training. Let $Q_{X|Y}$ denote the conditional distribution specified by G. If the input of G is drawn from P_Y^S, the joint distribution of the generated data will be $Q_{XY} = Q_{X|Y}P_Y^S$. We can estimate G by minimizing the Jensen-Shannon Divergence (JSD) between Q_{XY} and P_{XY}^S [179]:

$$\begin{aligned}\min_G \max_{D_G} \quad & \mathbb{E}_{(X,Y)\sim P_{XY}^S}[\log(D_G(X,Y))] \\ & + \mathbb{E}_{E\sim P_E, Y\sim P_Y^S}[\log(1 - D_G(G(Y,E),Y))],\end{aligned} \tag{2.9}$$

where D_G is an introduced discriminator [62] to play the mini-max game together with G. (2.9) is the negative cross entropy loss, and in some real experiments, we need to replace (2.9) by negative hinge-loss because it is more stable in image generation, as demonstrated in [181, 23].

2.3.2.3 Distribution Matching

As described above, we can then construct a new data distribution P_X^R with the label transformation network LT and the label influence network G. To estimate LT, we fix G and minimize the JSD between P_X^R and P_X^T w.r.t. LT by the following objective

$$\begin{aligned}\min_{LT}\max_{D_{LT}} & \mathop{\mathbb{E}}_{X\sim P_X^T}[\log(D_{LT}(X))]+ \\ & \mathop{\mathbb{E}}_{Z\sim P_Z, E\sim P_E, Y^S\sim\ P_Y^S}[\log(1-D_{LT}(G(LT(Y^S,Z),E)))].\end{aligned} \tag{2.10}$$

where D_{LT} is an introduced discriminator [62] to perform adversarial training with LT. In detail, when the label is continuous, the whole network is totally differentiable, so we can simply estimate LT by using backpropagation. However, in the case of discrete labels, we cannot backpropagate through the label y_i^r. Fortunately, we can assume a parametric form of P_Y^R, i.e., the categorical distribution, in case of discrete Y. Thus, we can make use of the Gumbel-softmax trick [93, 169] or the REINFORCE trick [279] to backpropagate through the discrete labels y_i^r. The two tricks have been successfully employed in various problems such as text generation [298, 71] and neural architecture search [285].

Gumbel-Softmax Trick Let Y^{Ro} and Y^{So} denote the one-hot representations of Y^R and Y^S, respectively. We can use a special LT function to sample from $P_{Y^R|Y^S}$:

$$\tilde{Y}_k^{Ro} = \frac{\exp((\log M_k Y^{So} + Z_k)/\tau)}{\sum_{i=1}^K \exp((\log M_i Y^{So} + Z_i)/\tau)}, \tag{2.11}$$

where $\tilde{Y}_k^{Ro}$ is the kth element of $\tilde{Y}^{Ro}$, $Z_k \sim Gumbel(0,1)$, τ is the temperature, and M_k is the kth row of the transition matrix $\mathbf{M}$, whose ijth element is $P(Y^R = i|Y^S = j)$. As $\tau \rightarrow 0$, $\tilde{Y}^{Ro}$ provides a good approximation of the one-hot Y^{Ro}. Since the softmax function is differentiable, it enables end-to-end learning of LT, which only contains $\mathbf{M}$ as parameters. **REINFORCE Trick** Because $P_{Y^R|Y^S}$ involves learnable parameters $\mathbf{M}$, we rewrite it as $P_{\mathbf{M}}(Y^R|Y^S)$ and reformulate (2.10) as

$$\begin{aligned} \min_{\mathbf{M}} \max_{D_{LT}} & \mathop{\mathbb{E}}_{X \sim P_X^T}[\log(D_{LT}(X))] + \\ & \mathop{\mathbb{E}}_{E \sim P_E, Y^R \sim \ P_{\mathbf{M}}(Y^R|Y^S), Y^S \sim \ P_Y^S}[\log(1 - D_{LT}(G(Y^R, E)))]. \end{aligned} \tag{2.12}$$

The gradient w.r.t. LT can be written as:

$$\begin{aligned} \mathop{\mathbb{E}}_{E \sim P_E, Y^R \sim \ P_{\mathbf{M}}(Y^R|Y^S), Y^S \sim \ P_Y^S} & [\log(1 - D_{LT}(G(Y^R, E))) \\ & \nabla_{\mathbf{M}} \log \ P_{\mathbf{M}}(Y^R|Y^S)]. \end{aligned} \tag{2.13}$$

Feature Matching Generally speaking, we estimate the prior distribution in the target domain P_Y^T by comparing the marginal distributions of X in the target domain and the transformed source domain. However, for some high dimensional data such as images, X might contain many redundant features X_R that are unrelated to Y, causing unnecessary estimation errors of P_Y^T. Intuitively, this is because the conditional distributions of these redundant features X_R satisfy $P_{X_R|Y} = P_{X_R}$, which are not helpful in identification of P_Y^T but will cause additional estimation error. To improve the estimation accuracy, we propose to estimate LT by matching $P_{h(X)}^R$ and $P_{h(X)}^T$ instead, where h is a pre-trained network that extracts compact representations from raw X data. Therefore, we replace (2.10) by

$$\begin{aligned} \min_{T} \max_{D_{LT}} & \mathop{\mathbb{E}}_{X \sim P_X^T}[\log(D_{LT}(h(X)))] + \mathop{\mathbb{E}}_{Z \sim P_Z, E \sim P_E, Y^S \sim \ P_Y^S} \\ & [\log\left(1 - D_{LT}(h(G(LT(Y^S, Z), E)))\right)]. \end{aligned} \tag{2.14}$$

Ideally, we aim to find $h(X)$ such that $Y \perp\!\!\!\perp X|h(X)$ by using the source-domain labeled data. This conditional independence property implies that $h(X)$ contains all information in X that is relevant to Y. Learning conditional invariant representation has been shown to be effective in correcting covariate shift [234]. However, since [234] set h as a linear transformation and measure conditional dependency using kernel measures [52], the method cannot learn compact representations for images and is computationally expensive. Here we use a convolutional network as h to extract feature representations and measure the dependency by assuming a (generalized) linear model for $P_{Y|h(X)}$. This is sensible because the features extracted by nonlinear neural networks are usually linearly separable. Proposition 1 shows how h can be learned to satisfy the conditional independence property. (The proof can be found at the supplementary material A.1)

PROPOSITION 1. *Assuming $P_{Y|h(X)}$ can be modeled by a (generalized) linear model, i.e., linear regression model for continuous Y and multinomial logistic regression model for discrete Y. Let sample size $n \to \infty$, h learned by minimizing the mean squared error (for continuous Y) or the cross-entropy loss (for discrete Y) satisfies $Y \perp\!\!\!\perp X|h(X)$.*

2.3.2.4 Shift Correction

After quantifying the target label distribution P_Y^T, the model with target shift problems should be corrected and adapted to the target domain. The previous work choose to re-train the model under the importance-weighted ERM framework [67, 221, 239]. In our framework, we can retrain the source-domain model with new data drawn from our model. As it is time-consuming to retrain a new model, a quick adaptation method is also provided in our framework. As described by Proposition 1, if h learned at the uniform Training set satisfies the conditional independence property with Y, the output layer of a neural network is the only module needed to be adapted to the new label distribution P_Y^R given the feature extractor h. In our framework, we fine-tune the output layer several epochs using the samples generated by our Label Influence Recovery network G with the label distribution P_Y^R learned by Label Transformation Network LT. As such, the output layer will be quickly adapted to the target domain.

2.4 Experiments

To verify the effectiveness and universality of the proposed framework, we design experiments for three target shift scenarios, i.e., the discrete, continuous, and multi-dimensional target shift, on various datasets.

2.4.1 Discrete Target Shift Experiments

We compare our method with the competitors on three datasets, e.g., MNIST, FASHION-MNIST, and CIFAR10 [125]. We follow the same setting of BBSE [146] and RLLS [8]. Specifically, for MNIST, we use a simple two-layer neural network; the Resnet-18 [81] and CNN in DCGAN [202] are chosen for CIFAR 10 and FASHION-MNIST, respectively. The learning rate is set to 0.01. Moreover, we use the network architecture of BigGAN [23] and the loss of TAC-GAN [61] to model the invariant conditional distribution $P_{X|Y}$. The original training sets given in the datasets are used as the training set for the proposed method and the baselines. The test set is sampled to have a specific label distribution P_Y^T and is of size 10,000. For the quantification of P_Y^T, we use the REINFORCE trick instead of Gumbel-softmax trick, as the temperature τ in the Gumbel-softmax trick is hard to choose.

2.4.1.1 Shift Settings

In this Chapter, the label distribution P_Y^S in the training set is a **uniform** distribution over all classes. For the test set, we consider three types of shifts: Tweak-One shift, Minority-Class shift, and Random Dirichlet shift. These settings are designed to capture diverse label probability changes, i.e., large label probability change, small label probability change, and random label distribution change. We repeat the experiments 10 times to verify the effectiveness and robustness of the proposed method.

Tweak-One Shift: To evaluate the performance on the large label probability quantification. In our experiments, the ratio of one class is set to $[0.5, 0.6, 0.7, 0.8, 0.9]$, respectively, while ratios of other classes are uniform.

Minority-Class Shift: To evaluate the performance on the small label probability quantification. In our experiments, $[20\%, 30\%, 40\%, 50\%]$ classes are set to 0.001, respectively, while ratios of other classes are uniform.

Random Dirichlet Shift: In this shift, we randomly generate a label distribution P_Y^T by employing the Dirichlet distribution with different values of the concentration parameter α. Then, we re-sample the test set according to the generated distribution P_Y^T. In our experiments, α are set to $10, 1, 0.1, 0.01$. Note that the generated label distribution P_Y^T tends to be smoother for bigger α.

2.4.1.2 Evaluation metrics and Results

As done in BBSE [146] and RLLS [8], the accuracy and F1 score [64] are used as evaluation metrics, allowing us to compare the performance of different methods more comprehensively [8]. We also evaluate the estimation error of the estimated label weights (P_Y^T/P_Y^S) by using mean square error (MSE).

We compare our method with the two recent methods: BBSE [146] and RLLS [8], which estimate label weights $\hat{\boldsymbol{\beta}}$ using the confusion matrix of a classifier f trained on the training set. To verify Proposition 1, we consider a variant of RLLS called RLLS(feature), which matches distributions on the feature space $h(X)$. RLLS(feature) can also be considered as setting $\psi(X)$ to $h(X)$ in (2.3). For the evaluation of the shift correction, we evaluate the performance of classifiers trained on the training set without adaptation (denoted as Baseline) and the classifiers trained on weighted training sets, where the weights are estimated by using target domain labels (denoted as BEST(ERM)). Similarly, we also test the classifiers trained on weighted training data, where the weights are obtained by RLLS, BBSE and RLLS(feature). For our method, we have two ways to utilize the label distributions estimated by our framework. The first one is to re-train a new classifier using the the weighted training set (Ours(ERM)). The second one is the fine-tuning method described in 2.3.2.4, which is denoted as Ours(Fine-tune). Specifically, we fine-tune the output layer of the pretrained classifier on the source domain by 10 epochs, using the data generated from our model.

Due to the page limit, we only show the results of CIFAR10 dataset in the Chapter and the results of MNIST and FASHION-MNIST can be found in the supplemental materials A.2. In terms of the estimation error of the target label distribution P_Y^T, the subfigure (a) of Figure 2.2, 2.3, 2.4 demonstrate that the label weights estimated by our framework are more accurate and stable than previous methods. In addition, the RLLS (feature) algorithm that matches label distribution on feature space of classifier trained on the training set also achieves better performance than BBSE and RLLS in most settings. For the accuracy and F1 score of the corrected classifiers, subfigures (b) and (c) of Figure 2.2, 2.3, 2.4 show that the classifier corrected by our framework can achieve better performance in both two evaluation metrics in most settings. Also, our fast fine-tune method achieves comparable performance compared with re-weighting methods.

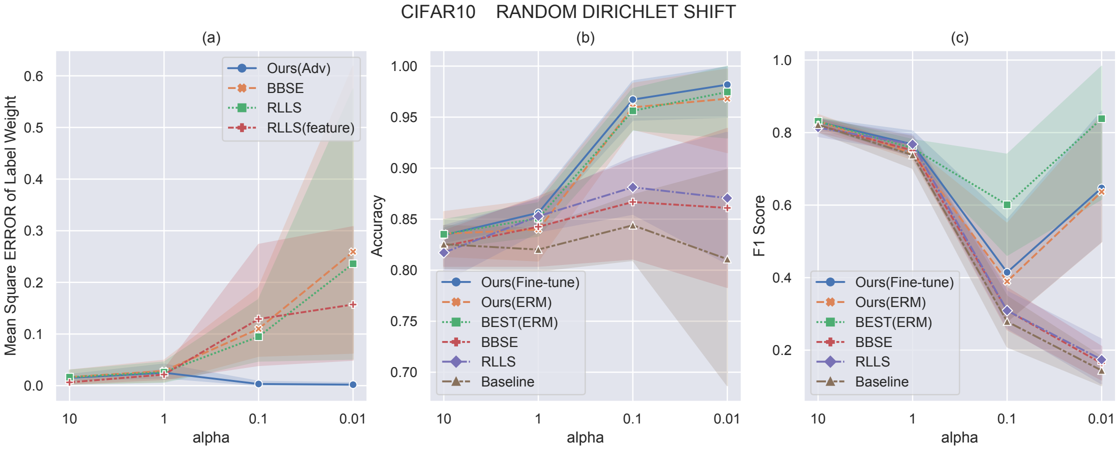


FIGURE 2.2. (a) Mean squared errors of estimated label weights (lower is better), (b) accuracy, and (c) F-1 score (higher is better) on CIFAR10 for uniform training set and random Dirichlet shifted test set, where the smaller *alpha* corresponds to larger shift.

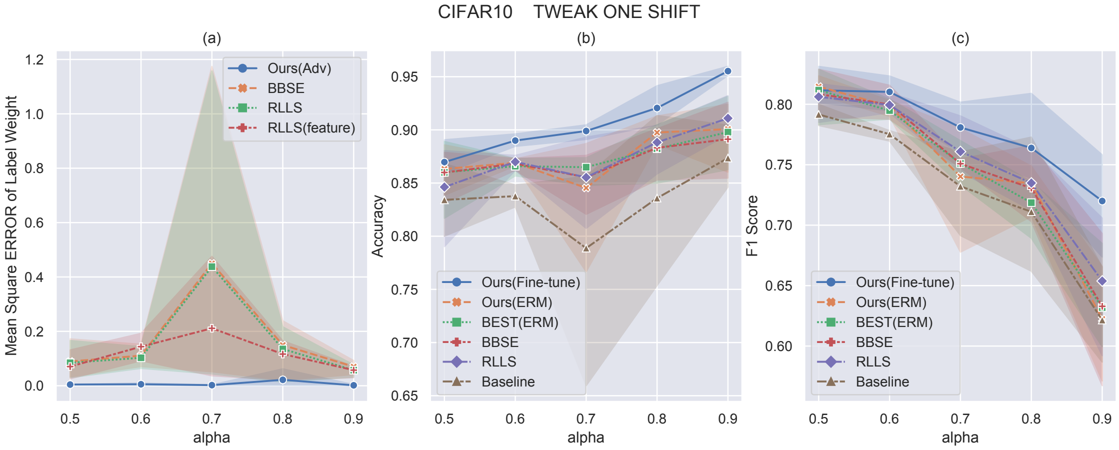


FIGURE 2.3. (a) Mean squared errors of estimated label weights (lower is better), (b) accuracy, and (c) F-1 score (higher is better) on CIFAR10 for uniform training set and Tweak-One shifted test set, where *alpha* is the probability of tweaked class.

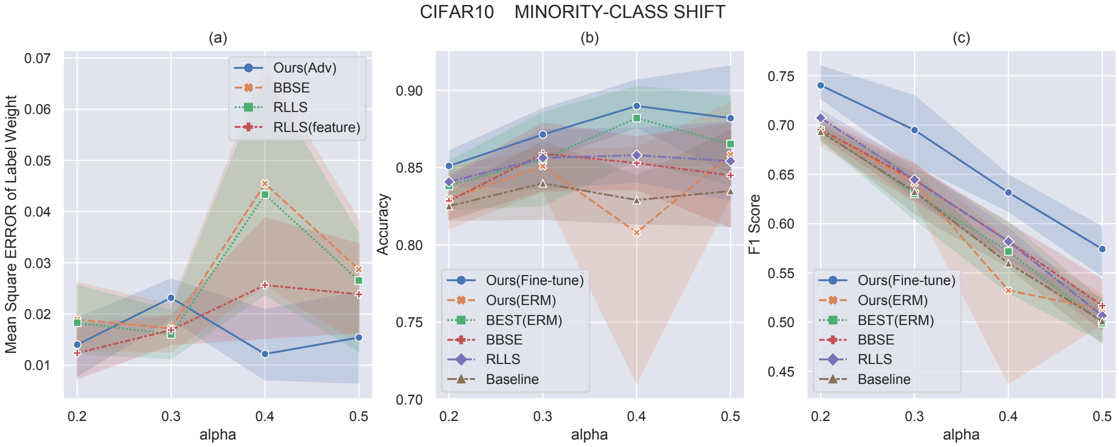


FIGURE 2.4. (a) Mean squared errors of estimated label weights (lower is better), (b) accuracy, and (c) F-1 score (higher is better) on CIFAR10 for uniform training set and minority-class shifted test set, where *alpha* is the ratio of minority classes.

2.4.2 Continuous Target Shift Experiments

In this section, we design two experiments to verify the effectiveness of our model on continuous target shift problems. Firstly, we conduct experiments on a synthetic data that evaluates the performance of our framework on simple continuous target shift problems. Then we apply our model on a real data application: Object 1D position prediction [174], which evaluates the performance of our model on the continuous target shift problem in the high-dimensional X situation.

2.4.2.1 Synthetic Data Experiment

In this experiment, we design a toy dataset by modifying a classic and popular synthetic data experiment (MOON dataset [54]) in *covariate shift*. We first generate two quarter circles with radius R 10 and sample size 1000 as the training set, which is shown in Figure 2.5a. The range of a single continuous label is from -10 to 10 and the values are uniformly distributed. Then we generate the test set with 500 samples in the same way according to several target label distributions. Here we consider 4 types of target shift to evaluate model performance and robustness.

Experimental Setting: In this experiment, the architectures of all modules in our framework are three-hidden layers neural networks with 10 hidden neurons. In the distribution matching module, the baseline KMM [311] uses MMD with the median kernel width to match the built data distribution P_X^{new} and target data distribution P_X^T. To fairly compare the methods, we also use MMD to do distribution matching.

Shift Settings: To evaluate the model's label quantification performance, we set 4 target shift situations.

Shift A: Set the target label distribution P_Y^T as a Gaussian distribution with the mean of $\frac{\sqrt{2}}{2} * R$ and variance of 1.

Shift B: Set the target label distribution P_Y^T as a Gaussian distribution with the mean of $-\frac{\sqrt{2}}{2} * R$ and variance of 1.

Shift C: The target label distribution is a mixture Gaussian distribution with Shift A and Shift B, with a mixture proportion 0.5.

Shift D: The target label distribution is a random label distribution generated by a randomly parameterized neural network.

Baselines: The classic KMM methods [311] are chosen as our baselines. We consider two variants: KMM that matches the distributions in the raw input space and KMM(feature) that matches the distributions in feature space of the regressor.

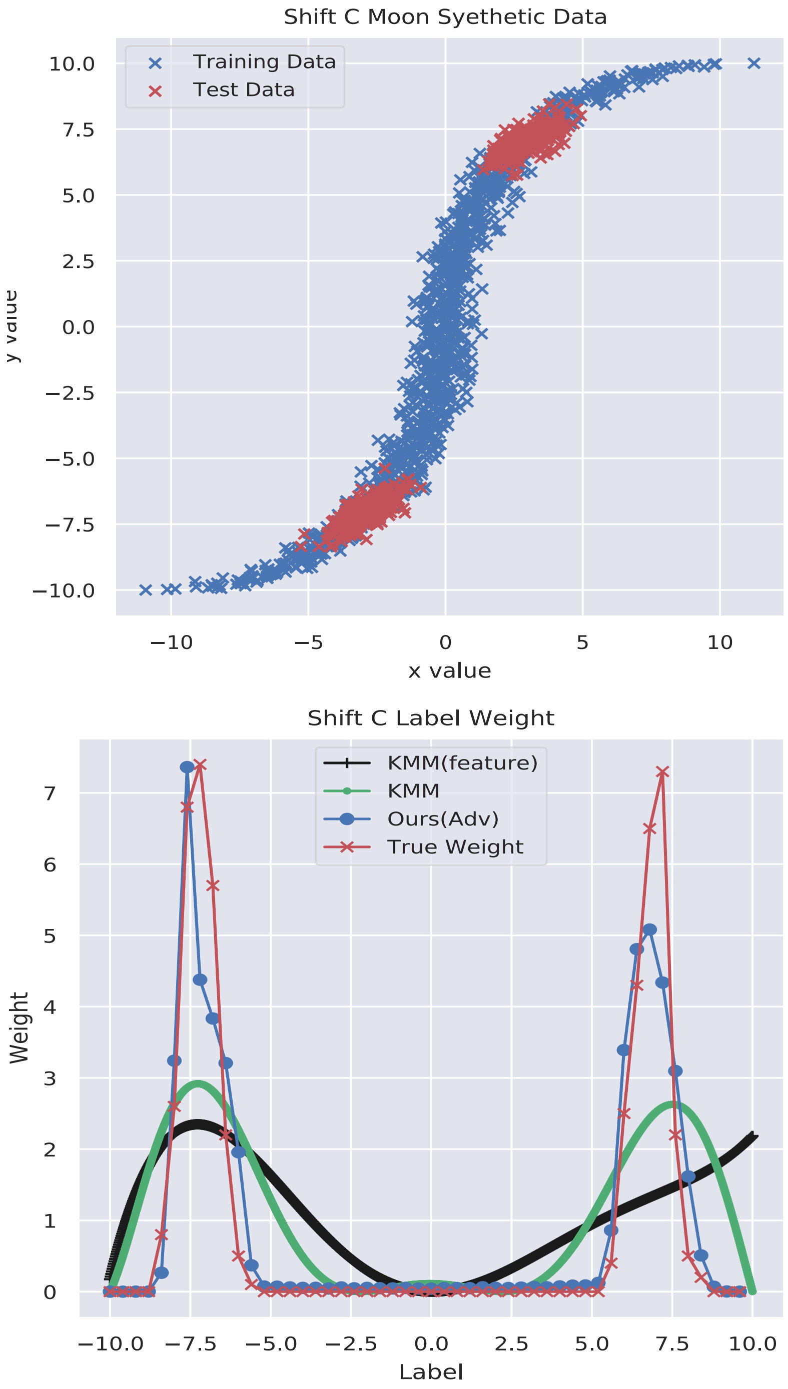


FIGURE 2.5. (a) The illustration of Moon Synthetic Data (Shift C), where the generated two quarter circles training set as blue symbols show. (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

	SHIFT A	SHIFT B	SHIFT C	SHIFT D
Baseline	0.0061 ± 0.0012	0.0059 ± 0.0008	0.0055 ± 0.0009	0.034 ± 0.0114
KMM	0.0048 ± 0.0011	0.0044 ± 0.0005	0.0044 ± 0.0004	0.0275 ± 0.0096
KMM (feature)	0.0045 ± 0.0007	0.0039 ± 0.0006	0.0043 ± 0.0004	0.0276 ± 0.0097
Ours	**0.0036** **± 0.0002**	**0.0024** **± 9e-5**	**0.0036** **± 0.0004**	**0.0251** **± 0.0121**

TABLE 2.1. The results of Continuous target shift Synthetic Data Experiments. The value is the mean square error of prediction value and groun truth. The baseline is the original regressor trained on the standard training set, and the KMM is [311]

Results: In this section, to compare the performance of estimated target label distribution qualitatively, we visualize the estimated density ratio (P_Y^T/P_Y^S) of Shift C in Figure 2.5. More figures about other shift settings can be found in supplementary materials A.3.Visually, our model has better label distribution estimation performance compared with other methods.

Then we evaluate the mean square error of the baseline regressor without adaptation and three others corrected by KMM, KMM(feature), and Ours(Adv) respectively. The results are shown in Table 2.1. It can be seen that the MSE errors of our framework are significantly lower than those of the other methods in all shift settings, and KMM (feature) achieves slightly better performance than the original KMM method in some settings, which verifies the correctness of Proposition 1.

2.4.2.2 Object 1D location Prediction Experiment

In this experiment, we use a popular disentanglement dataset called Sprites [1] [174]. This dataset consists of 737,280 2D shapes images, which are generated from 6 ground-truth independent latent factors. Some example images are shown in Figure 2.6. The factors include color, shape, scale, rotation, x, and y positions of a sprite. We choose the x or y position of sprites as the target variable and consider it as a regression problem.

[1] https://github.com/deepmind/dsprites-dataset

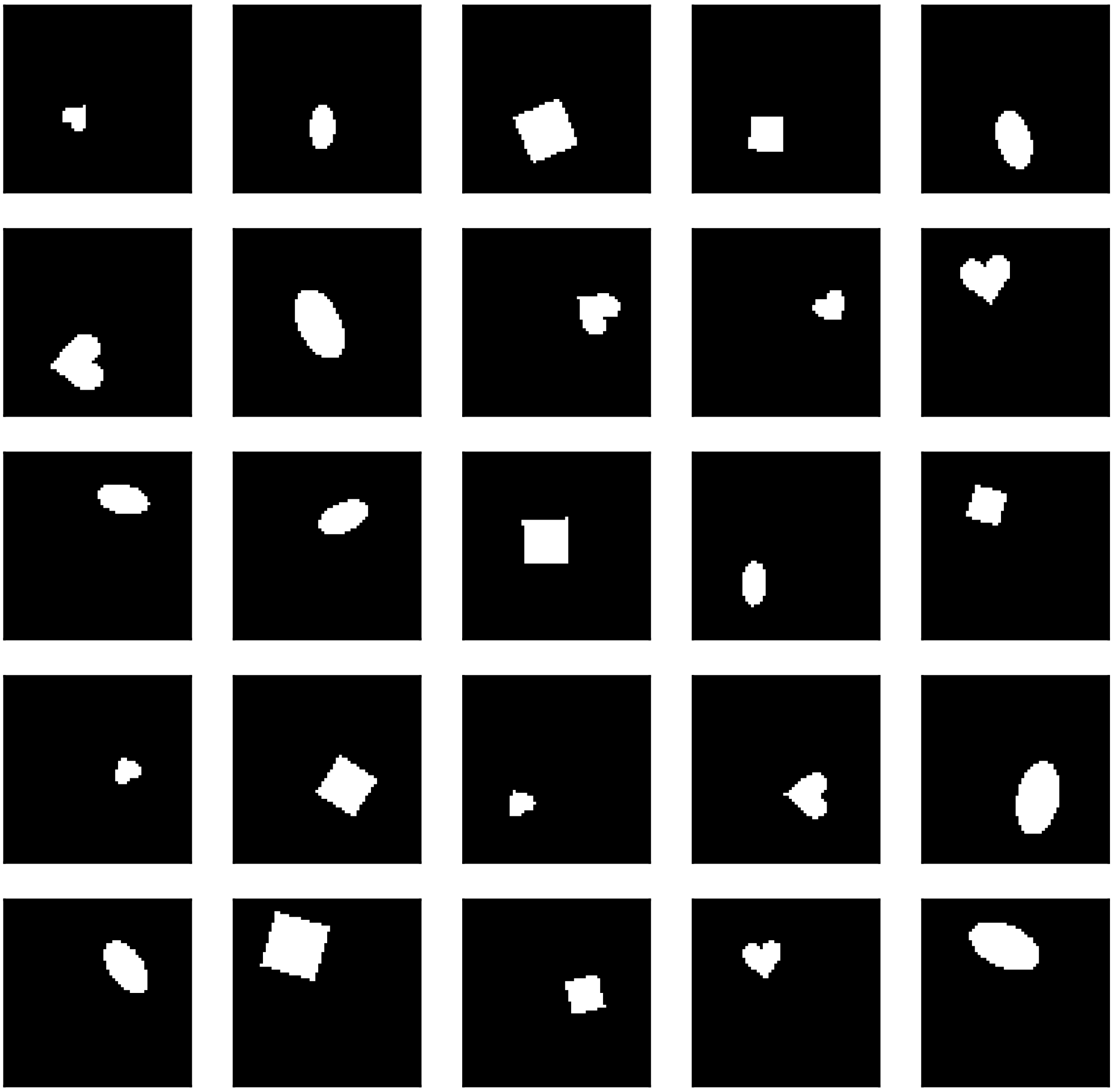


FIGURE 2.6. Illustration of the sprites dataset. This sprites in this dataset have 3 shapes(square, ellipse, heart), 6 scales values linearly spaced in [0.5, 1], 40 orientation values in [0, 2 pi], 32 X position values in [0, 10], 32 Y position values in [0, 10]

Experimental Setting: We use the network architecture in DCGAN [202] as feature extractor for the regressor, the learning rate for the regressor is set to 1e-4, which is the best learning rate according to our experiments. The DCGAN [202] is used to model the invariant distribution $P_{X|Y}$ and the Transformation Model LT is a simple 3-layer neural network.

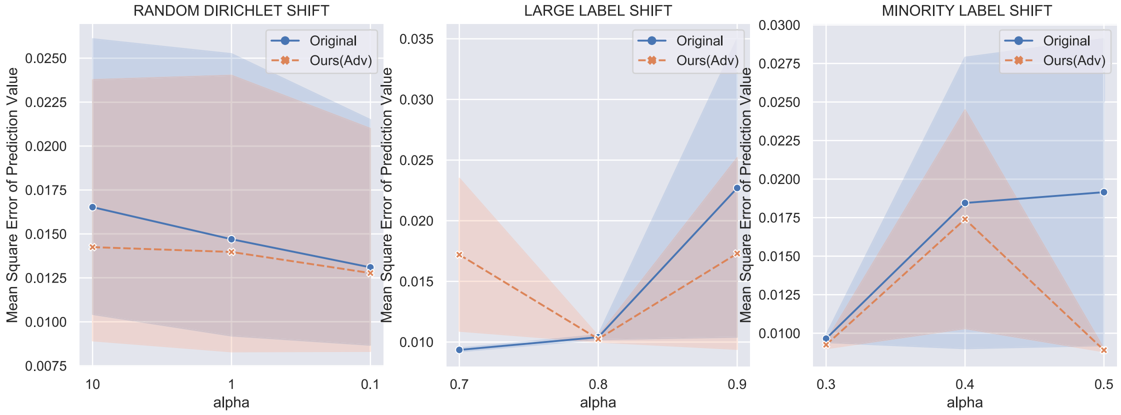


FIGURE 2.7. The prediction mean square error of 1D sprite position prediction (lower is better). (a) Random Dirichlet shift, where the smaller *alpha* corresponding to the bigger shift. (b) Large target shift, where *alpha* is the possibility of shifted label. (c) Minority shift, where *alpha* is the ratio of minority classes.

For training set, we randomly sample 40000 images with **uniform** x value distribution from overall dataset and the test set consists of 40000 samples (sampled from specified distribution for target shift).

Shift Settings: As the x position (or y position) in this dataset has 32 possible values (but we see it as a regression problem), we can use the same shift method with 2.4.1.1 to evaluate the methods' target shift quantification ability. Specifically, we repeat the experiments 3 times for each setting to evaluate the model performance.

Baselines and Results: As the KMM method cannot be applied into large-scale dataset, so the only baseline in this experiment is the baseline regressor without adaptation. We evaluate the mean square error of output value (x or y position) of original regression model and the regressor corrected by our framework. The results are shown as Figure 2.7, and the model corrected by our framework outperforms the basline model in most settings.

2.4.3 Multi-Dimensional Target Shift Experiments

In this experiment, we design a simple multi-dimensional target shift experiment, which is object 2D location prediction. We use the same dataset with the object 1D location prediction experiment, but we predict both x and y position values of a sprite. As such, the label Y in this experiment is 2-dimension, increasing the difficulty of detecting and correcting the target shift.

Experimental Setting: We use the same network architecture and train/test split as described in 2.4.2.2 . For Transformation Model LT, two networks are used to model the x and y position target shift respectively as the x and y position value in this dataset are independent. As such, using two networks will reduce the difficulty of quantifying target label distribution P_Y^T.

Shift Settings: The settings are also same with 2.4.2.2 described, but we shift the x and y position value respectively.

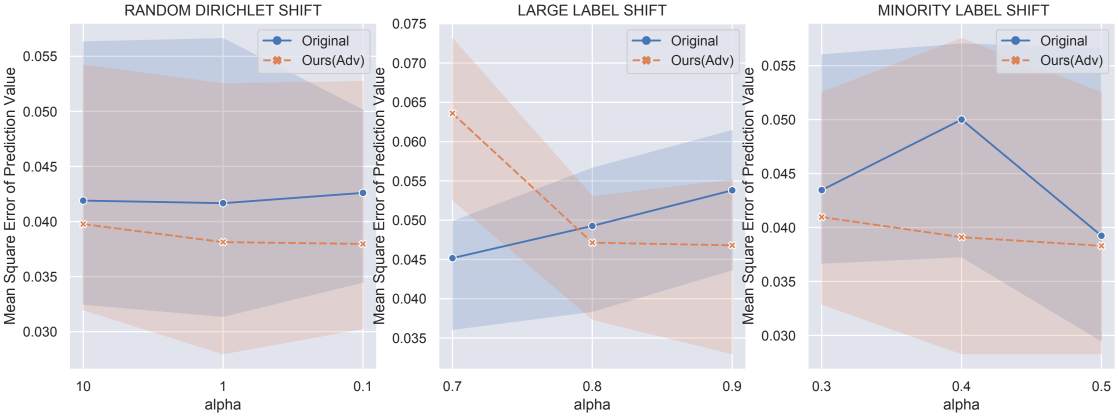


FIGURE 2.8. The prediction mean square error of 2D sprite position prediction (Lower is better). (a) Random Dirichlet shift, where the smaller *alpha* corresponds to larger shift. (b) Large target shift, where *alpha*is the possibility of a shifted label (c) Minority shift, where *alpha* is the ratio of minority classes.

Baselines and Results: Similar to the 1D position prediction, the baseline is the baseline regressor without adaptation as our methods is the first method which is compatible with large-scale multi-dimensional target shift problems. The results are shown in Figure 2.8. It can be seen that the regressor corrected by our framework can achieve lower MSE error than the baseline method in most settings.

2.5 Conclusion

In this Chapter, we propose an end-to-end target shift quantification and correction framework called Label Transformation Framework which can deal with discrete, continuous and multi-dimensional target shift problems. Based on this framework, we further find that matching the distributions of a feature representation of X that discards the information irrelevant to Y can have better performance over other methods which quantify the label distribution P_Y^T based on scratch data or biased output. In the experiments, we apply our framework to several classification and regression tasks under various target shift settings. The results show that our framework has better performance and universality than previous methods. Future work will be extending our framework to address conditional shift, where $P_{X|Y}$ also changes across domains.

CHAPTER 3

Alleviating Semantics Distortion in Unsupervised Low-Level Image-to-Image Translation via Structure Consistency Constraint

Unsupervised image-to-image (I2I) translation aims to learn a domain mapping function that can preserve the semantics of the input images without paired data. However, because the underlying semantics distributions in the source and target domains are often mismatched, current distribution matching-based methods may distort the semantics when matching distributions, resulting in the inconsistency between the input and translated images, which is known as the semantics distortion problem. In this Chapter, we focus on the low-level I2I translation, where the structure of images is highly related to their semantics. To alleviate semantic distortions in such translation tasks without paired supervision, we propose a novel I2I translation constraint, called *Structure Consistency Constraint* (SCC), to promote the consistency of image structures by reducing the randomness of color transformation in the translation process. To facilitate estimation and maximization of SCC, we propose an approximate representation of mutual information called relative Squared-loss Mutual Information (rSMI) that enjoys efficient analytic solutions. Our SCC can be easily incorporated into most existing translation models. Quantitative and qualitative comparisons on a range of low-level I2I translation tasks show that translation models with SCC outperform the original models by a significant margin with little additional computational and memory costs.

3.1 Introduction

Image-to-image translation, or domain mapping, aims to translate an image in the source domain $\mathcal{X}$ properly to the target domain $\mathcal{Y}$. It has been applied to various vision tasks

[217, 224, 58, 255, 284]. Early works [195, 90, 150] considered supervised image-to-image (I2I) translation on paired datasets, and methods based on conditional generative adversarial networks can generate high-quality translations [90, 266, 195]. However, since paired data are often unavailable or expensive to obtain, unsupervised I2I translation has attracted intense attention in recent years [321, 296, 115, 17, 88, 133, 114, 193].

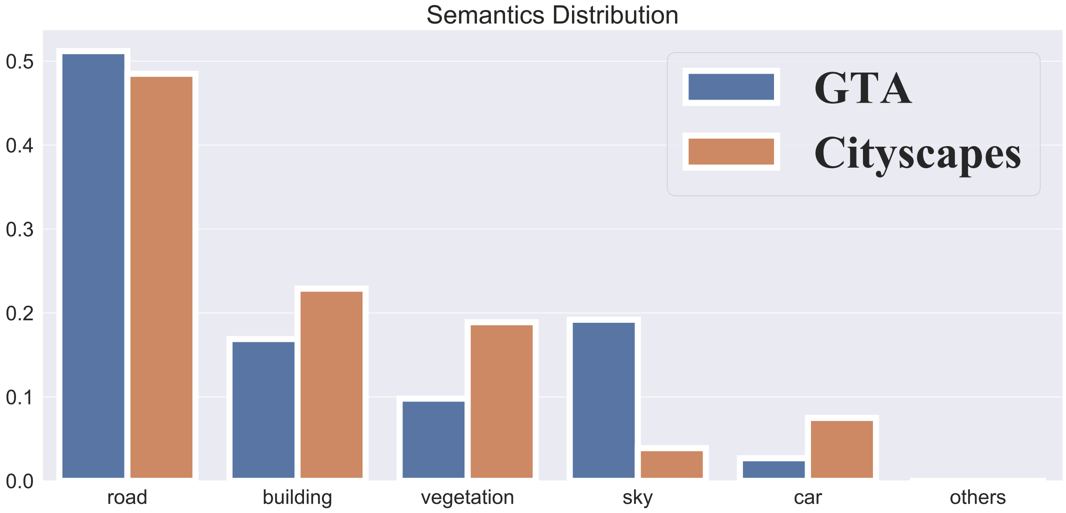


FIGURE 3.1. Class distributions in GTA and Cityscapes. We can see that the ratio of the sky in GTA is significantly higher than it in Cityscapes, and thus the distribution matching based method has to translate the sky to vegetation/building to align the distributions.

Benefiting from generative adversarial networks (GANs) [62], many works aim to perform unsupervised I2I translation by finding G_{XY} such that the translated images and target domain images have similar distributions, *i.e.*, $P_{G_{XY}(X)} \approx P_Y$. Due to an infinite number of functions that can satisfy the adversarial loss, GAN alone could learn a function far away from the true one. To remedy this issue, various constraints have been placed on the learned mapping function. For instance, the well-known cycle-consistency [321, 115, 296] enforces the translation function G_{XY} to be bijective. DistanceGAN [17] preserves the pairwise distances in the source images. GcGAN [51] forces the function to be smooth w.r.t. certain geometric transformations of input images. DRIT++ [133] and MUNIT [88] learn disentangled representations by embedding images onto a domain-invariant content space and a domain-specific attribute space and the mapping function can be then derived from representation learning components.

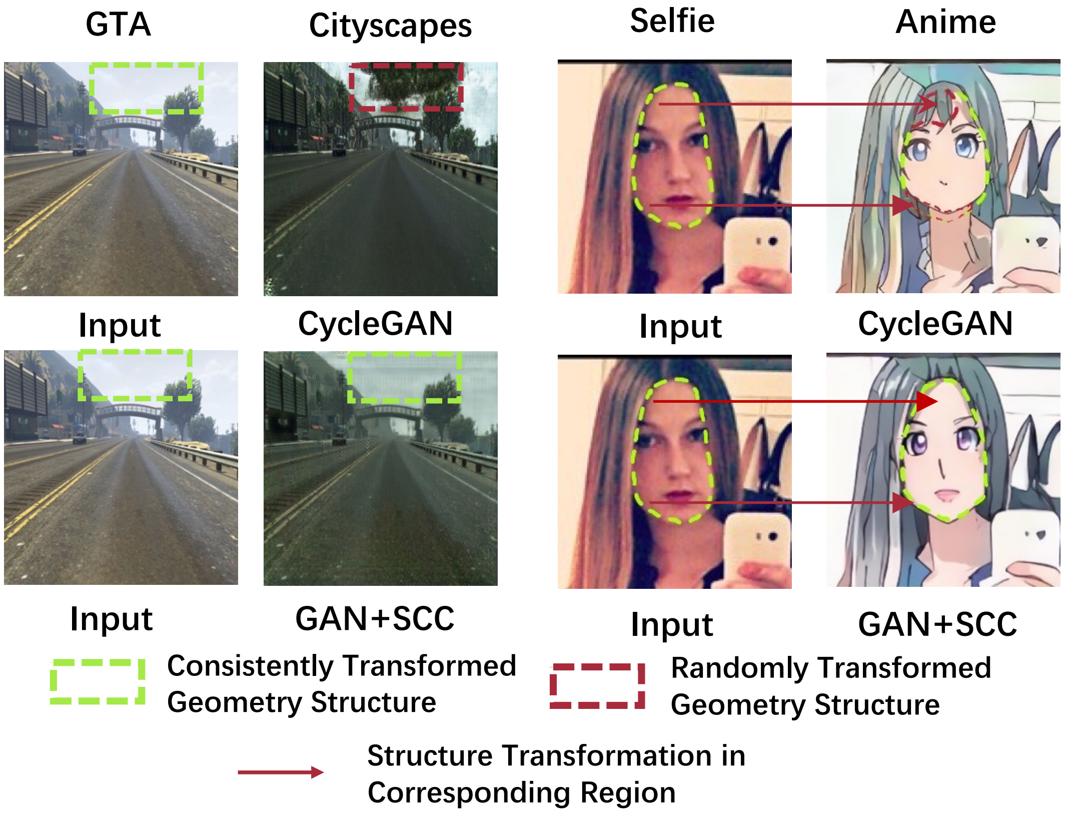

GTA
Cityscapes
Selfie
Anime
Input
CycleGAN
Input
CycleGAN
Input
GAN+SCC
Input
GAN+SCC
Consistently Transformed Geometry Structure
Randomly Transformed Geometry Structure
Structure Transformation in Corresponding Region

FIGURE 3.2. The illustration about the inconsistent geometry structure translation causes the semantic-distortion problem in unsupervised low-level image translation. Visually, we can see that the geometry structures of the sky and human face are distorted during translation in CycleGAN, which causes the semantical distortion *e.g.*, sky to vegetation, a face without fringe to face with fringe.

The above methods perform well when the two domains differ only in style information. However, in most unpaired datasets, not only style but also the underlying semantic distributions differ across source and target datasets [95]. Taking GTA to Cityscapes as an example, we perform the class statistics of GTA and Cityscapes, and the results are given as Figure 3.1. It can be seen that the class distributions in GTA are different from that in Cityscapes, *e.g.*, the proportion of sky in the GTA is significantly higher than that in Cityscapes, while the proportion of vegetation in GTA is lower than that in Cityscapes. Figure 3.2 also shows an example in selfie→anime translation, where the ratio of human faces with bangs in the Anime dataset is significantly higher than that in the Selfie dataset. In these cases, previous GAN-based methods *e.g.*, CycleGAN

[321], which aims to align the distribution between domain *i.e.*, $P_{G_{XY}(X)} \approx P_Y$, may translate sky to building/vegetation in GTA2cityscape or automatically add the bangs on the human face in selfie2anime for the sake of aligning distribution (Figure 3.2), resulting in a semantic mismatch between input and translated images *i.e.*, *semantics distortion* problem.

It is hard to solve the *semantics distortion* problem in a universal way [95] when the given source and target dataset have unmatched semantics distributions because the characterization of semantics may vary from task to task. This lack of universally best choice is usually formalized in what is called the “No-Free Lunch" theorem [281, 276, 130], indicating that there is no single I2I algorithm that can perform better than all the other algorithms on all I2I applications. As such, we need to use suitable inductive bias [15, 108] to guide the translation model to preserve the related content according to the specific requirements of different I2I applications. For example, in high-level I2I image translation tasks, the pose/location of an object may be regarded as the semantics, but the type of object (e.g. cat→human face) is the style information that should be translated, and thus [284] introduces the pose bias to preserve pose structure properly during translation.

In this Chapter, we consider a widely applicable low-level image translation problem [21], which is fundamental in a wide range of computer vision applications, such as domain adaptation [83], segmentation [321], and simulation-to-real [208]. In low-level I2I, the difference between domains arises from the low-level information *e.g.*, resolution, illumination, color rather than geometry variation, while the structure (e.g. the shapes of objects) in images is most invariant across the source and target domains, *i.e.*, the semantics of an image is highly related to its structure (*shape of objects*). Therefore, the semantic distortion can be regarded as the change of structures in the translated images, as illustrated in Figure 3.2. Motivated by this, a natural solution to alleviate semantic distortion in this translation task would be to preserve the structure of source images.

To guarantee the consistency of image structure between source and translated images, we propose an I2I translation constraint, called *Structure Consistency Constraint* (SCC)

. We observe that the pixel values before and after translation are usually highly correlated if the image structure is preserved (Figure 3.3). Based on this observation, we propose a mutual information (MI)-based dependency measure that models the nonlinear relationships between pixel values in the source and translated images. To efficiently estimate MI between pixel values, we propose the so-called relative Squared-Loss Mutual Information (rSMI) which can be estimated in an analytic form. By maximizing rSMI together with the GAN loss, our approach can significantly reduce the semantic distortion by better preserving image structures. In experiments, to show the effectiveness and compatibility of our *structure consistency constraint*, we incorporate it into the GAN framework and other existing image translation methods (*e.g.*, CycleGAN, CUT [193]). The quantitative and qualitative comparisons with existing I2I methods on several low-level tradatasets demonstrate that models with SCC outperform the corresponding baselines by a significant margin at only little computational and memory costs [1].

3.2 Methodology

Unsupervised I2I translation aims to find a mapping function G_{XY} between two domains $\mathcal{X}$ and $\mathcal{Y}$ given unpaired samples $\{x_i\}_{i=1}^{N}$ and $\{y_j\}_{j=1}^{M}$ drawn from the marginal distributions P_X and P_Y, respectively. To alleviate *semantics distortion* problem in low-level I2I translation, we directly promote the structure consistency of the source and translated images because the image structure is highly related to its semantics in this task. In the following, we first present our motivation of placing the MI-based structure consistency constraint (SCC), and then give the details about SCC, which aims to reduce the randomness of color transform in the translation process and thus promote the consistency of geometry structure between source and translated images.

[1] Codes are available at https://github.com/CR-Gjx/SCC

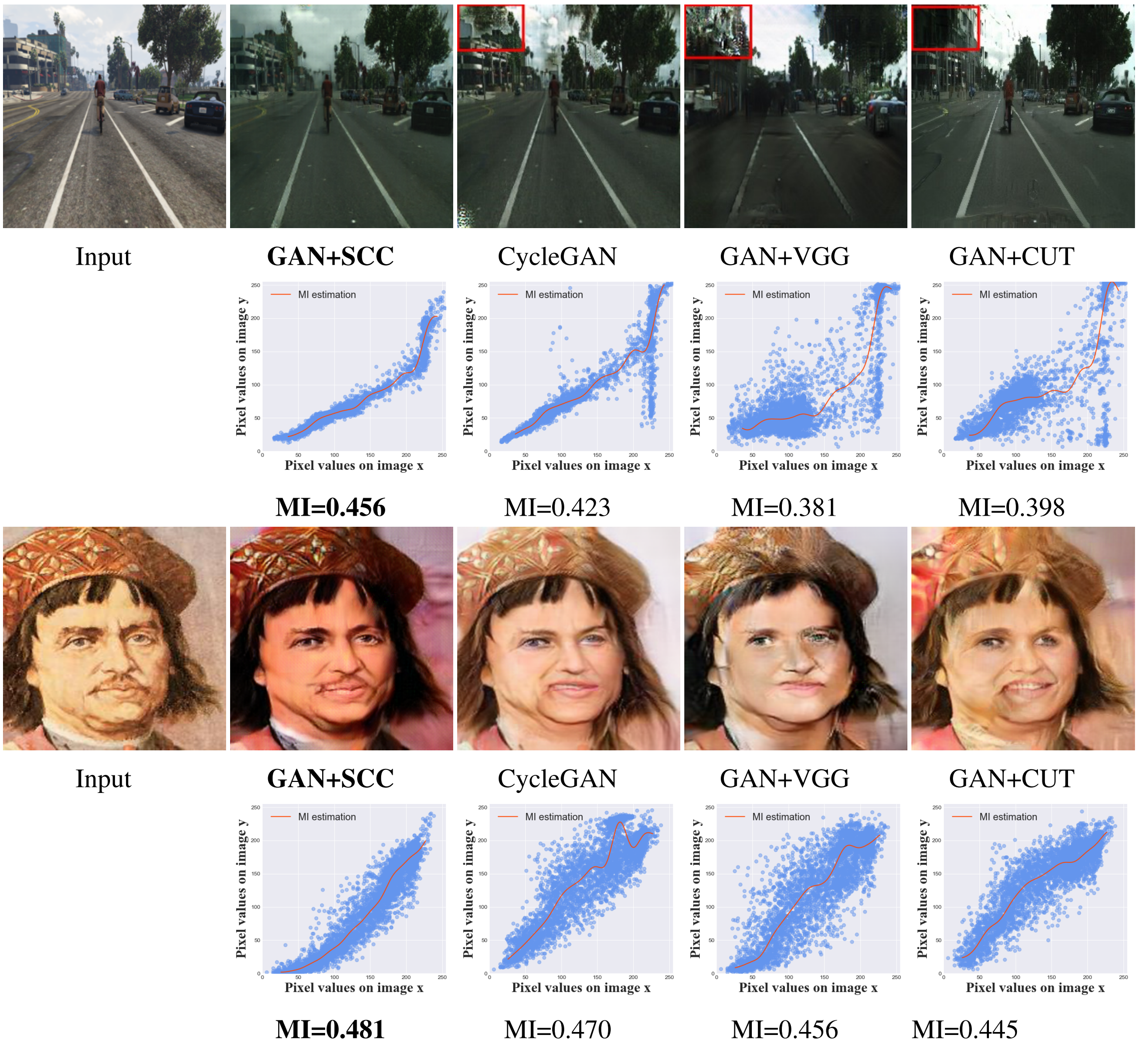


FIGURE 3.3. Unsupervised image translation examples on GTA → Cityscapes. Portrait → Photo. The top row is the translated results by each method. The bottom row is the scatter plot of the pixel values in the input image x and its corresponding pixel value in the translated image $\hat{y}$, which shows the non-linear dependency of pixel values in two images. Obviously, the stronger the dependency between pixel values in the input image (X-axis) and the translated images (Y-axis), the better the geometry structure of the input image is maintained. MI stands for the mutual information estimated by our rSMI method. Specifically, the VGG refers to the Contextual loss [175] of VGG features.

3.2.1 Motivation

As illustrated in Figure 3.3, 3.5 (a), and 3.7, advanced methods, *e.g.*, CycleGAN, CUT [193], Contexual loss [175], U-GAT-IT [114], MUNIT [88], may change the geometry structure of input images and potentially cause the semantics mismatch between input and translated images. Therefore, it is essential to enforce a constraint such that we can ensure the learned function G_{XY} change the image style with minimal structure distortion. Our work is the first to explore such constraints for unsupervised image-to-image translation.

As we know, geometric structures in an images are often outlined by colors. So, if we hope to presereve the geometry structure during translation, we would expect the color translation to be consistent between the input and output images. For example, the green leaf in summer should be translated to yellow in autumn, but we do not expect it to be translated into a colorful one, otherwise, we cannot identify it as a leaf. Based on this observation, we plot the corresponding pixel values of images before and after translation at the bottom row of Figure 3.3. We can see that if the pixel values in the translated image (Y-axis) are more dependent on the pixel values (X-axis) in the input images, more structures will be preserved. Obviously, previous methods (*e.g.*, CycleGAN, CUT, Contextual loss of VGG feature) fail to translate color within a geometry structure consistently, and such randomness of the color transformations result in the distortion of geometry structure and semantics. Therefore, reducing the randomness of color transformation is an effective way to alleviate the *semantic-distortion* problem in I2I translation.

Motivated by the analysis, we develop the *structure consistency constraint* (SCC) as a general and effective constraint to preserve the pixel-level structure during the translation process. SCC exploits mutual information to model the non-linear dependencies of pixel values between the input and translated images, thus reducing the randomness of color transformation in the translation. As illustrated in Figure 3.4, our SCC is enforced into the input and translated images and thus allows one-sided unsupervised domain mapping, *i.e.*, G_{XY} can be trained independently from G_{YX}. Applying our SCC to a vanilla GAN, the pixel values before and after translation have stronger dependency (higher MI), and the model therefore better preserves the geometric structures as shown in Figure 3.3, thus reducing semantic distortion in low-level I2I translation. In the following, we present the details of our approach.

3.2.2 Approximate Representation of Mutual Information

For a source domain image $x_i \in \mathcal{X}$ and its translation $\hat{y}_i = G_{XY}(x_i)$, we denote V^{x_i} and $V^{\hat{y}_i}$ as the random variables for pixels in x_i and $\hat{y}_i$, respectively. Thus, pixels in x_i, i.e., $\{v_j^{x_i}\}_{j=1}^M$, can be regarded as data sampled from $P_{V^{x_i}}$, and the pixels in $\hat{y}_i$, i.e., $\{v_j^{\hat{y}_i}\}_{j=1}^M$, can be considered as data sampled from $P_{V^{\hat{y}_i}}$, where M is the number of pixels of the image. Formally, the mutual information between V^{x_i} and $V^{\hat{y}_i}$ is

$$MI(V^{x_i}, V^{\hat{y}_i}) = \mathbb{E}_{(v^{x_i}, v^{\hat{y}_i}) \sim P_{(V^{x_i}, V^{\hat{y}_i})}} \left(\log \frac{P_{(V^{x_i}, V^{\hat{y}_i})}}{P_{V^{x_i}} \otimes P_{V^{\hat{y}_i}}} \right) \tag{3.1}$$

where $P_{(V^{x_i}, V^{\hat{y}_i})}$ is the joint distribution of V^{x_i} and $V^{\hat{y}_i}$, $P_{V^{x_i}} \otimes P_{V^{\hat{y}_i}}$ is the product of two marginal distributions $P_{V^{x_i}}$ and $P_{V^{\hat{y}_i}}$. Because V^{x_i} and $V^{\hat{y}_i}$ are low-dimensional, a straightforward way to estimate (3.1) is to estimate the distributions P based on the histogram of the images. Next, we will introduce how we estimate the mutual information between pixels from two domain images and backpropagate it to optimize parameters in the translation network.

To enable efficient backpropagation, we propose the relative Squared-loss Mutual Information (rSMI), which is an extension of the well-known Squared-loss Mutual Information (SMI) [244] and can be estimated analytically. For conventional presentation, we denote $P_{V^{x_i}} \otimes P_{V^{\hat{y}_i}}$ as Q_i, and $P_{(V^{x_i}, V^{\hat{y}_i})}$ as S_i. Then, the SMI based on Pearson (PE) Divergence [237] between $P_{V^{x_i}}$ and $P_{V^{\hat{y}_i}}$ is expressed as:

$$\begin{aligned} SMI(V^{x_i}, V^{\hat{y}_i}) &= D_{PE}(P_{(V^{x_i}, V^{\hat{y}_i})} || P_{V^{x_i}} \otimes P_{V^{\hat{y}_i}}) \\ &= D_{PE}(S_i || Q_i) \\ &= \mathbb{E}_{Q_i}[(\frac{S_i}{Q_i} - 1)^2]. \end{aligned} \tag{3.2}$$

Because $\frac{S_i}{Q_i}$ is unbounded, $SMI(V^{x_i}, V^{\hat{y}_i})$ can be infinity, causing numeric instability in the backpropagation. We thus use the relative Pearson(rPE) Divergence [288] to alleviate the problem:

$$D_{rPE}(S_i \mid\mid Q_i) = D_{PE}(S_i \mid\mid \beta S_i + (1 - \beta) Q_i). \tag{3.3}$$

Here, we introduce the mixture distribution $\beta S_i + (1 - \beta) Q_i$, $\beta \in (0, 1)$, to replace Q_i. Benefiting from the modification, the density ratio will be bounded to [0,$\frac{1}{\beta}$]. Thus, the proposed rSMI between V^{x_i} and $V^{\hat{y}_i}$ can be written as:

$$\begin{aligned} rSMI(V^{x_i}, V^{\hat{y}_i}) &= D_{rPE}(P_{(V^{x_i}, V^{\hat{y}_i})} || P_{V^{x_i}} \otimes P_{V^{\hat{y}_i}}) \\ &= \mathbb{E}_{\beta S_i + (1-\beta) Q_i} [(\frac{S_i}{\beta S_i + (1-\beta) Q_i} - 1)^2] \end{aligned} \tag{3.4}$$

To estimate the $rSMI(V^{x_i}, V^{\hat{y}_i})$, we directly estimate the density ratio using a linear combination of kernel functions of $\{v_j^{x_i}\}_{j=1}^M$ and $\{v_j^{\hat{y}_i}\}_{j=1}^M$:

$$\begin{aligned} \frac{S_i}{\beta S_i + (1-\beta) Q_i} &= \omega_\alpha(v^{x_i}, v^{\hat{y}_i}) \\ &= \alpha^T \phi(v^{x_i}, v^{\hat{y}_i}) \end{aligned} \tag{3.5}$$

where $\phi \in \mathbb{R}^m$ is the kernel function, $\alpha \in \mathbb{R}^m$ is the parameter vector we need to solve, and m is the number of kernels. Referring to the least-squares density-difference estimation [238], the solved optimal solution of $\hat{\alpha}$ is (the derivation is given in the appendix A.1):

$$\begin{aligned} \hat{\alpha} &= (\hat{H} + \lambda R)^{-1} \hat{h}, \\ \hat{H} &= \frac{1-\beta}{n} (K \circ L)(K \circ L)^T + \frac{\beta}{n^2} (KK^T) \circ (LL^T), \\ \hat{h} &= \frac{1}{n^2} (K 1_n) \circ (L 1_n) \end{aligned} \tag{3.6}$$

where R is a positive semi-definite regularization matrix, n is the sample number, 1_n is the n-dimensional vector filled by ones, and K and L are two $m \times n$ matrices composed by kernel functions, and the Hadamard product of K and L is used to define ϕ, that is $\phi(v^{x_i}, v^{\hat{y}_i}) = K(v^{x_i}) \circ L(v^{\hat{y}_i})$. Finally, an appropriate mutual information estimator of with smaller bias is expressed as:

$$\widehat{rSMI}(V^{x_i}, V^{\hat{y}_i}) = 2\hat{\alpha}^T \hat{h} - \hat{\alpha}^T \hat{H} \hat{\alpha} - 1. \tag{3.7}$$

Note that, the computation of $\widehat{rSMI}(V^{x_i}, V^{\hat{y}_i})$ is resource friendly, as it can be solved analytically. Thus, the parameters in the translation neural network can be efficiently updated by backprogation.

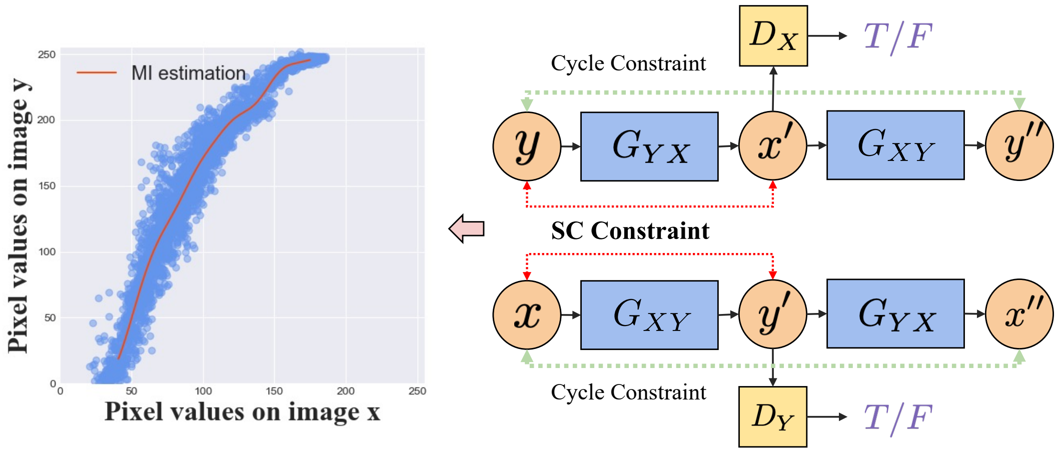
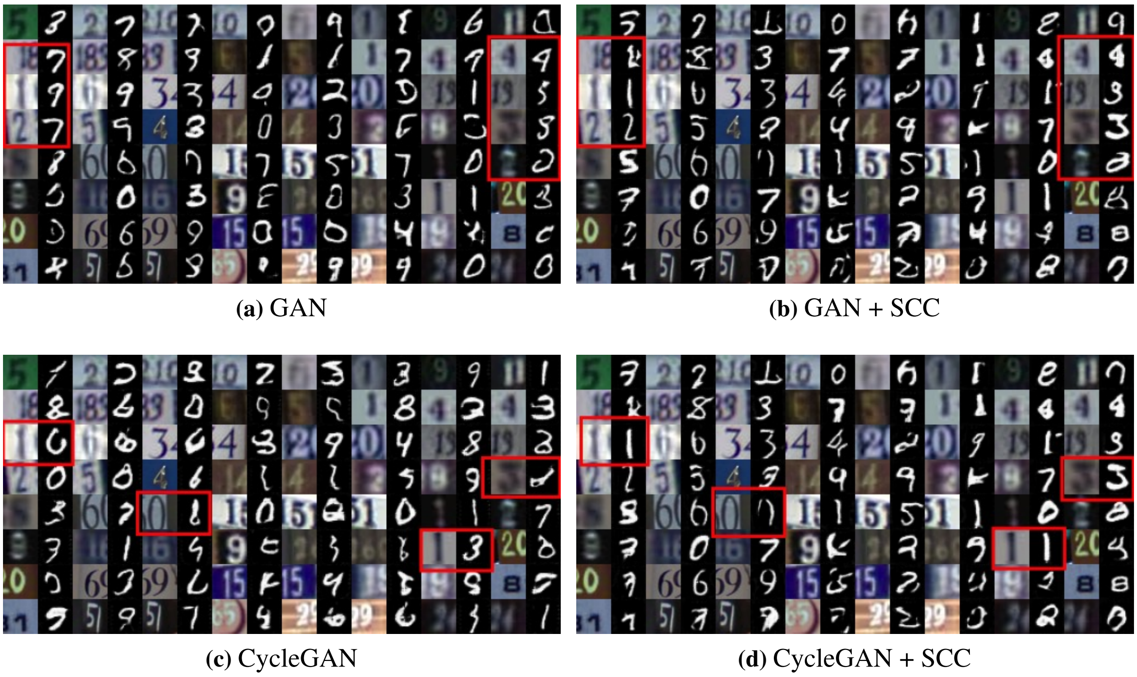


FIGURE 3.4. An illustration of structure consistency constraint. The left figure shows that the pixel value in the input image x and its corresponding pixel value in the translated image $\hat{y}$ have strong non-linear dependencies, so we add the structure consistency constraint to model the dependencies of pixel values in two domain images.



(a) GAN

(b) GAN + SCC

(c) CycleGAN

(d) CycleGAN + SCC

FIGURE 3.5. Qualitative comparisons on SVHN→MNIST. From Figure (a) and (b), we can see that the GAN method has no collapse solution by combining with our SCC. Also, the semantics distortion problem in CycleGAN is alleviated after incorporating with SCC.

TABLE 3.1. Classification accuracy for digits experiments.

	Translated Images as Test set			Translated Images as Training set		
Method	S → M	M → M-M	M-M → M	S → M	M → M-M	M-M → M
GAN alone	21.3±9.5	54.6±40.5	80.3±3.5	28.6±10.8	45.7±31.2	95.5±0.4
+ SCC	37.3±1.2	96.3±0.2	90.9±0.5	47.9±2.3	86.2±1.9	96.0±0.1
CycleGAN	26.1±8.1	95.3±0.4	84.7±2.5	31.6±5.6	83.8±3.0	95.9±0.4
+ SCC	38.0±0.5	**96.7±0.1**	91.5±0.3	47.4±2.0	87.7±2.1	**96.1±0.2**
GcGAN-*rot*	32.5±2.0	95.0±0.6	85.9±0.8	40.9±6.5	84.6±2.8	96.0±0.1
+ SCC	36.5±1.3	96.4±0.3	91.8 ±1.0	47.5±1.2	89.5±0.6	96.1±0.1
GcGAN -*vf*	33.3±4.2	95.2±0.4	84.5±1.5	31.6±5.6	83.8±3.0	95.9±0.4
+ SCC	37.0±0.8	96.6±0.3	91.8±0.8	49.5±4.9	87.8±2.3	96.0±0.1
Cyc + rot + SCC	39.0±0.5	96.5±0.3	91.8±1.0	50.5±1.8	**89.8±0.5**	**96.1±0.1**
Cyc + vf + SCC	**44.6±6.8**	**96.7±0.3**	**92.0±0.8**	**51.3±5.4**	89.0±0.8	**96.1±0.1**

3.2.3 Full Objective

Following the analysis above, our structure consistency constraint (SCC) for I2I translation using mutual information can be expressed as:

$$\mathcal{L}_{SCC} = \frac{1}{N}\sum_{i=1}^{N} \widehat{rSMI}(V^{x_i}, V^{G_{XY}(x_i)}), \tag{3.8}$$

where N is the number of samples, and $G_{XY}(x_i) = \hat{y}_i$. We directly maximize $\mathcal{L}_{SCC}$ to guarantee more local geometric structures of images being invariant in the translation process. By combining SCC with the standard adversarial loss, the image geometry will be preserved while its style is changed. As a result, one-sided unsupervised domain mapping can be targeted. The full objective will take the form:

$$\begin{aligned} &\min_{G_{XY}}\max_{D_Y} \mathcal{L}_{GAN+SCC}(G_{XY}, D_Y) \\ &\qquad = \mathcal{L}_{GAN}(G_{XY}, D_Y) - \lambda_{SCC}\mathcal{L}_{SCC}(G_{XY}), \end{aligned} \tag{3.9}$$

where $\mathcal{L}_{gan}$ is the adversarial loss [62], which introduced a discriminator D_Y, to encourage the distribution of output matches the distributions of target domain images, *i.e*, $P_{G_{XY}(X)} \approx P_Y$. In addition, to guarantee the distribution consistency in the pixel level, we use a GAN based on the 1×1 convolution. The objective function is as follows:

$$\begin{aligned} \mathcal{L}_{GAN}(G_{XY}, D_Y) &= \mathbb{E}_{y\sim P_Y}[\log D_Y(y)] \\ &\quad + \mathbb{E}_{x\sim P_X}[\log(1 \ - \ D_Y(G_{XY}(x))). \end{aligned} \tag{3.10}$$

In Equation 3.9, λ_{SCC} is a hyperparameter to weight $\mathcal{L}_{gan}$ and $\mathcal{L}_{SCC}$ in the training procedure. The proposed SCC can easily be integrated into various I2I translation frameworks, *e.g.,* CycleGAN [321] and CUT[193], by replacing the loss $\mathcal{L}_{gan}$ with the losses in these methods.

3.3 Experiments

In this section, we perform quantitative experiments on three typical unsupervised low-level image translation benchmarks: Digits Translation, Unsupervised Segmentation and Image Generation (*e.g.*, Cityscapes [36]), and Simulation-to-Real (*e.g.*, Maps [90] and GTA2cityscapes [208]). Because these benchmarks have the true label of the translation images, we can quantitatively evaluate whether the translation model causes the semantics distortion problem or not. Further, to qualitatively evaluate the translation quality of our method, we also perform experiments on Selfie → Anime, Portrait → Photo, Horse → Zebra datasets.

Effectiveness and Compatibility We couple our structure consistency constraint (SCC) with the vanilla GAN to show its effectiveness, and incorporate SCC with some popular methods such as CycleGAN [321], GcGAN [51], and U-GAT-IT [114] to show its compatibility. Then we make qualitative and quantitative comparisons with the recent published unsupervised I2I translation methods *e.g.*, CycleGAN [321], GcGAN [51], CoGAN [149], SimGAN [222], BiGAN [43] , DistanceGAN [17], CUT [193]), the VGG-based Contextual loss [175], the VGG-based Content loss [57], L1 loss of VGG feature [175], DRIT++ [133], UNIT [148], MUNIT [88], AGGAN [249], and U-GAT-IT [114]. Specifically, the current baselines have their own advantages and disadvantages: some baselines perform well on one task but perform poorly on other tasks. For example, some style transfer methods do not perform well on unsupervised image segmentation. As such, following the current literature, we compare our methods with SOTA methods for each application.

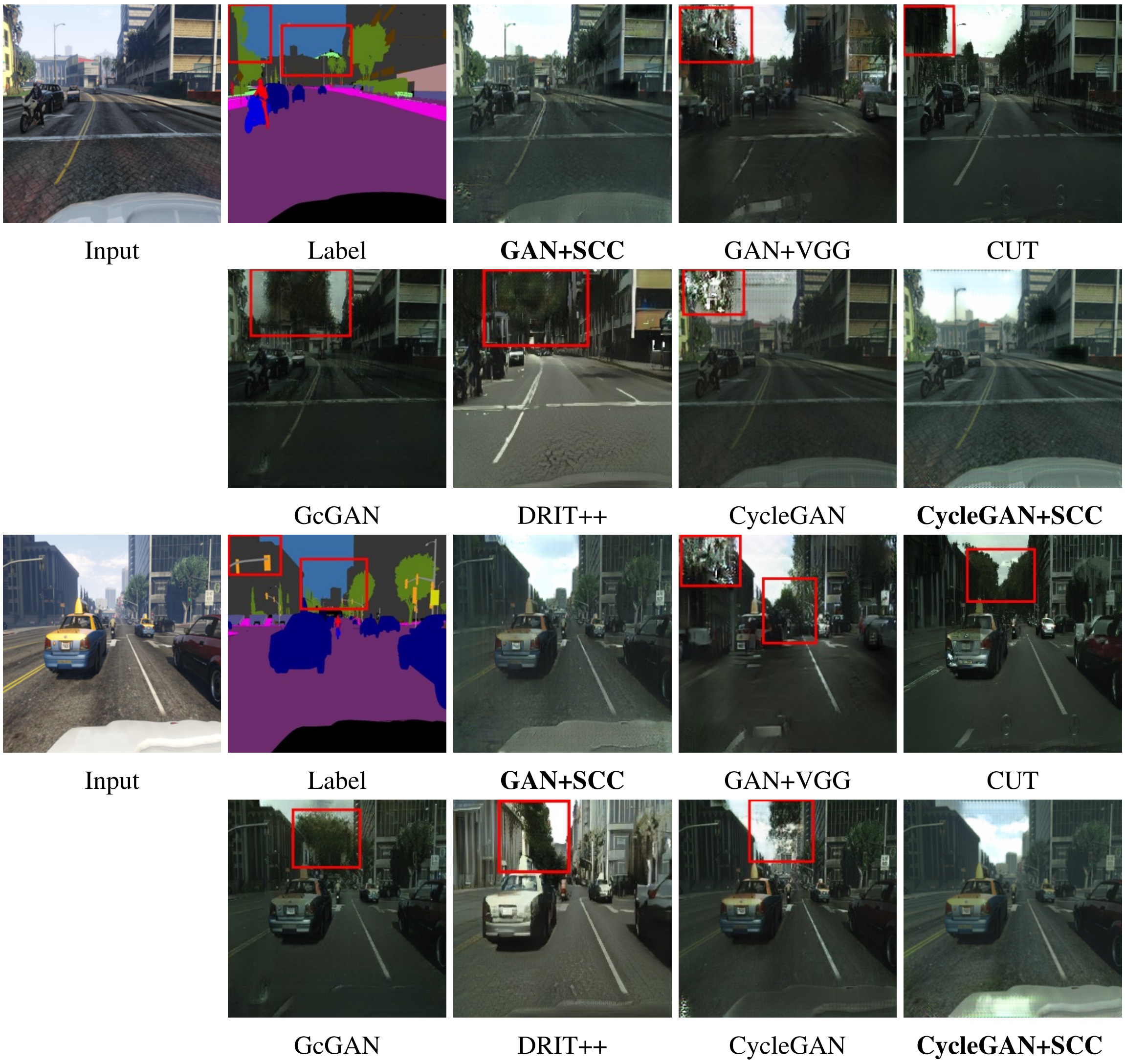
Sensitivity We perform the sensitivity analysis by varying the hyper-parameter λ_{SCC} on GTA2cityscapes.

In the appendix, we investigate the influence of our SCC on the generation diversity A.2.2 and training stability A.2.3. We examine all the experiments three times and report the average scores to reduce random errors.

For the implementation of the mutual information estimator presented in section D1.7, we set the hyperparameter β to 0.5 (more analysis about other values of β are given at the appendix A.2.1), and utilize nine Gaussian kernels for both input images x and translated images $\hat{y}$. Then we apply our SCC to all the baselines and keep other experimental details including hyper-parameters, networks in baselines the same. Due to page limit, we provide more experimental details and qualitative results in the Appendix A.6 and A.7, respectively.

TABLE 3.2. Quantitative scores on GTA $\rightarrow$ Citycapes,Citycapes parsing $\rightarrow$ image and Photo $\rightarrow$ Map. The scores with * are reproduced on a single GPU using the codes provided by the authors. More qualitative results are given at the Appendix A.7.2.

Methods	GTA $\rightarrow$ Citycapes			Citycapes parsing $\rightarrow$ image			Photo $\rightarrow$ Map		
	pixel acc $\uparrow$	class acc $\uparrow$	mean IoU $\uparrow$	pixel acc$\uparrow$	class acc$\uparrow$	mean IoU$\uparrow$	RMSE $\downarrow$	acc%(δ_1) $\uparrow$	acc%(δ_2) $\uparrow$
CoGAN	\	\	\	0.40	0.10	0.06	\	\	\
BiGAN/ALI	\	\	\	0.19	0.06	0.02	\	\	\
SimGAN	\	\	\	0.20	0.10	0.04	\	\	\
DistanceGAN	\	\	\	0.53	0.19	0.11	\	\	\
GAN + VGG	0.216	0.098	0.041	0.551	0.199	0.133	34.38	28.1	48.8
DRIT++	0.423	0.138	0.071	\	\	\	32.12	29.8	52.1
GAN *	0.382	0.137	0.068	0.437	0.161	0.098	33.22	19.3	42.0
+ SCC	0.487	0.148	0.089	0.642	0.215	0.155	28.91	38.6	61.8
GcGAN-rot *	0.405	0.139	0.068	0.551	0.197	0.129	27.98	42.8	64.6
+ SCC	0.445	0.162	0.080	0.651	0.228	0.162	**26.55**	**44.7**	**66.5**
CycleGAN *	0.232	0.127	0.043	0.52	0.17	0.11	26.81	43.1	65.6
+ SCC	0.386	0.161	0.076	0.571	0.192	0.134	26.61	44.7	66.2
CUT *	0.546	0.165	0.095	0.695	0.259	0.178	28.48	40.1	61.2
+ SCC	**0.572**	**0.185**	**0.11**	**0.699**	**0.263**	**0.182**	27.34	39.2	60.5



Input Label **GAN+SCC** GAN+VGG CUT

GcGAN DRIT++ CycleGAN **CycleGAN+SCC**

Input Label **GAN+SCC** GAN+VGG CUT

GcGAN DRIT++ CycleGAN **CycleGAN+SCC**

FIGURE 3.6. Unsupervised image translation examples on GTA → Cityscapes. The generated examples clearly show that our SCC can alleviate the semantic distortion problem *e.g.*, sky to tree/building in mainstream translation models. More examples are given at Appendix A.7

3.3.1 Quantitative Evaluation

3.3.1.1 Digits Translation

We examine three digit I2I translation tasks: SVHN→MNIST, MNIST-M→MNIST and MNIST→MNIST-M [2]. The models are trained on the training split with images size 32×32, and λ_{SCC} is set to 20. We adopt the classification accuracy as the evaluation

[2] refer to S→M, M-M→M and M→M-M

metric, and design two evaluation methods: (1) we train a classifier on the target dataset's training split. The fake images translated from the source dataset's test images are used to compute the classification accuracy. This evaluation method can only measure the quality of translated images. (2) a classifier is trained on the translated images from the source dataset's training images, and test the performance of this classifier on the target dataset's test split. This evaluation method can measure both the quality and diversity of translation images, but it is unstable [3].

We conduct each experiment five times to reduce the randomness of GAN-based approaches. The scores are reported in Table 3.1. Generally, by incorporating our SCC, all the baselines show promising improvements in both accuracy and stability, especially for the challenging task S→M. Some qualitative results are shown in Figure 3.5. More details and results are given in Appendix A.6.1 and A.7.1, respectively.

3.3.1.2 Segmentation in Cityscapes

Following [51, 321], we train the models using the unaligned 3975 images of Cityscapes [36] with 128×128 resolution. We evaluate the domain mappers using FCN scores and scene parsing metrics as previously done in [321]. Specifically, for parsing→image, we use the pre-trained FCN-8s [153] provided by pix2pix [90] to predict segmentation label maps from translated images, then compare them with true labels using parsing metrics including pixel accuracy, class accuracy, and mean IoU. We do not report the score of DRIT++, because its network size is too big to perform experiments with 128×128 resolution, resulting in the unfair comparison with other methods, but the results of other datasets can still show the superiority of our method over DRIT++.

As reported in Table 3.2, the results of all the image translation methods are improved if further constrained by our SCC, which shows the effectiveness of our method on reducing the semantics distortion problem. In particular, GcGAN coupled with SCC yields a promising improvement compared with GcGAN in the parsing → image task.

[3] Domain adaptation. has access to the labels of source domain images while I2I translation does not.

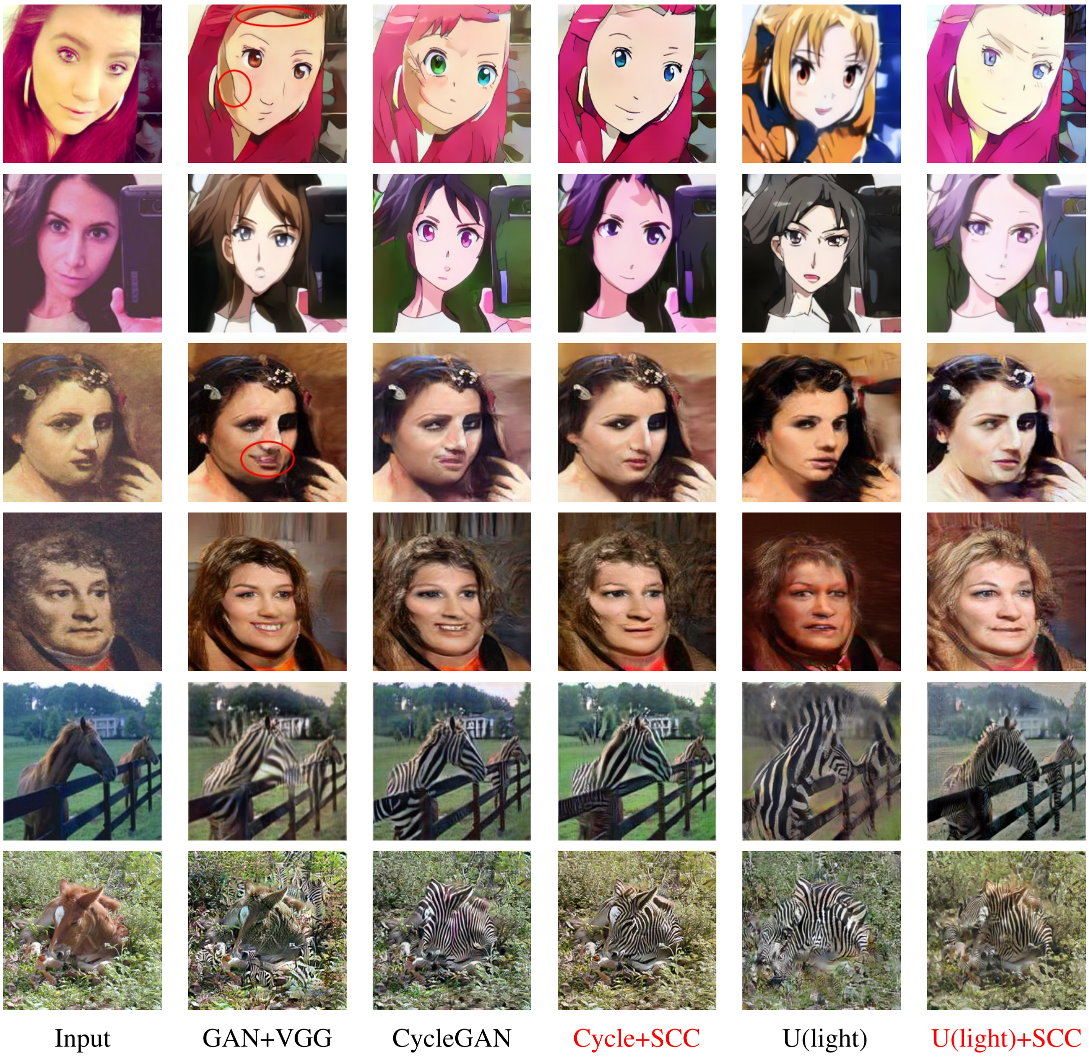


FIGURE 3.7. Qualitative results on Selfie → Anime, Portrait → Photo, Horse → Zebra datasets. More qualitative results are given in A.7.3. We can see that the no matter personal identification or horse shape is better preserved by the translation model empowered by our SCC.

3.3.1.3 Maps

The Maps dataset [90] contains 2194 aerial photo-map image pairs, with 1096 pairs for training and 1098 pairs for evaluation. For evaluation, we employ the metrics including RMSE and pixel accuracy with threshold δ ($\delta_1 = 5$ and $\delta_2 = 10$) suggested by GcGAN [51]. All images are resized to 256×256 resolution. Following [321, 51], the network details are similar to the details of Cityscape, but the generator contains 9 res-blocks for images with 256×256 resolution.

The scores are reported in Table 3.2. Compared with the vanilla GAN, our SCC can significantly improve translation accuracy to 38.6% and 61.8% from 19.3% and 42.0% with the threshold of δ_1 and δ_2, respectively. Moreover, integrating our SC constraint into CycleGAN and GcGAN can generate better translations than both individual ones. This further demonstrates the compatibility of our SCC. Qualitative results are shown in A.7.1.

3.3.1.4 Simulation to Real: GTA to Cityscapes

To evaluate the effectiveness of our SCC on simulation to real tasks, we use the GTA [208] to cityscapes datasets. Specifically, we use the official training split of GTA dataset the training dataset. All images are resized to 256×256 resolution during training. In the test process, we translate the first 500 images in the GTA test set to the cityscapes style, and use the pre-trained FCN-8s [153] provided by pix2pix [90] to predict the segmentation label maps from translated images, and calculate the scores with the true label in the GTA.

The results are give as Table 3.2, and the sample translated images are given as Figure 3.6. Our SCC can consistently alleviate the semantic distortion problem in GTA2cityscape task, as Figure 3.6 shows, all other translation models tend to translate sky to vegetation to align the distribution, but the translation model with SCC can maintain sky during translation, and thus we can consistently improve the segmentation score when coupling SCC with other models.

3.3.2 Qualitative Evaluation

We implement the qualitative evaluation on anime2selfie [114], horse2zebra [321], photo2portrait [132]. We choose CycleGAN, GcGAN, AGGAN, DRIT, UNIT, MUNIT, and CUT as baselines. All images are resized to 256×256 resolution. More experimental details are given in A.4.4.

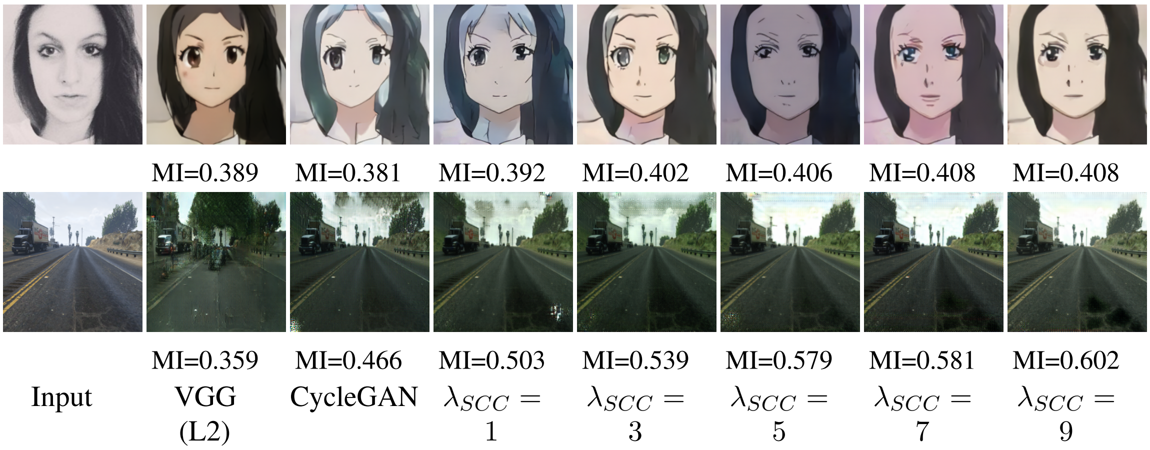


FIGURE 3.8. Sensitivity analysis examples on Selfie → Anime and GTA → Cityscapes. Obviously, the semantics distortion problem in CycleGAN is alleviated after incorporating with our SCC.

TABLE 3.3. The results of User Study: the percentage of users prefer a particular model. To avoid the concern of cherry-picking, qualitative results of U-GAT-IT and our results are used in the user study. Sample images are given in Appendix A.7.3.

	hor2zeb	sel2ani	pho2por	Paramaters
Cyc+Gc+SCC	**33.20**	**47.85**	**56.89**	45.2MB
U-GAT-IT	32.22	37.22	19.00	134.0MB
MUNIT	1.25	1.67	8.44	46.6MB
DRIT	5.28	2.94	3.00	65.0MB
CycleGAN	28.05	10.32	12.67	28.3MB

Following [114], we use KID score [19] as the evaluation metric. The results are reported in Appendix A.3.1 because the pages are limited, and we can see that the method coupled with our SCC can even achieve better results than those methods with larger model sizes. As the qualitative results are shown in Figure 3.7, after adding our SCC, the translated images retain more geometric structure than the original images, and are consistent with the style of the target images. Specifically, the light version of U-GAT-IT with our SCC can achieve better performance than the full version of U-GAT-IT, even with a half size of parameters. Then we conducted a user study, in which 180 participants were asked to choose the best-translated image given the domain names *e.g.*, selfie → anime, exemplar images in the source and target domains, and the corresponding translated images from different methods. The results shown in Table 3.3 demonstrate that most users choose the outputs of our method, which shows that preserving the structure of the image can significantly improve the appearance attraction of the translated images. More qualitative results are given in appendix A.7.4 .

3.3.3 Sensitivity Analysis

We study the influence of SCC by performing experiments with different λ_{SCC}. As shown in Table 3.4 and Figure 3.8, the performance of translation models are all improved to some extent after incorporating our SCC. However, when λ_{SCC} becomes too large, the improvement with our SCC is limited as the model focuses on reducing geometry distortion and ignores the style information learned from GAN. More examples are given in Appendix A.7.5. A practical strategy of choosing λ_{SCC} is to find the largest λ_{SCC} with normal style information using binary search. Specifically, the first value of λ_{SCC} can be set to 5, which can promote the structure consistency of most translation models.

TABLE 3.4. The segmentation scores for different λ_{SCC} of the model CycleGAN + SCC in the datasets GTA2cityscapes.

λ_{SCC}	0	1	3	5	7	9
pixel acc ↑	0.232	0.292	0.322	0.360	0.382	0.386
class acc ↑	0.127	0.136	0.143	0.160	0.160	0.161
mean IoU ↑	0.0432	0.055	0.059	0.070	0.075	0.076

3.4 Related Work

Unsupervised Image-to-Image Translation. Although unsupervised image-to-image (I2I) translation has obtained some promising progress in recent years, several works study it from an optimization perspective. Specifically, Cyclic consistency based GAN, *e.g.,* CycleGAN [321], DualGAN [296] and DiscoGAN [115], is a general approach for this problem. DistanceGAN [17] and GcGAN [51] further introduced distance and geometry transformation consistency to constraint the search space of mapping functions. Instead of exploiting general constraints for the subject, more works developed novel frameworks to investigate special settings of unsupervised I2I translation. Several other works [88, 33, 133, 132, 220] mapped the content and style information of images into disentangled spaces for multi-modal translations. However, we find that the complex neural networks and many hyper-parameters make the optimization process unstable [114]. [45, 99, 57, 175, 107] tried to reduce the perceptual loss or content loss based on a pre-trained VGG model to reduce the content of two domain image, which is computationally cost and cannot be easily adapted to the data on hand. Moreover, [266, 222, 21, 246, 176, 250, 291, 315] use the attention-based/ pretrained model or pre-define functions to preserve the semantics during translation. SRUNIT [95] promote the robustness of feature translation, but SRUNIT is mainly incorporated into CUT [193]. However, how to preserve the semantics via low-level information is under explored.

Mutual Information (MI). Mutual information is the measure of dependency between two random variables, and it is widely used in machine learning and particularly suitable for canonical tasks, *e.g.*, multi-modalities images registration [322, 170, 163]. Since computing MI is difficult [192], researchers have taken much effort to improve the estimation of MI. For example, early works studied Non-parametric models based on Kernel Density Estimator (KDE) [100, 124, 106, 230, 231], K-nearest Neighbor Method (KNN) [123, 122], and likelihood-ratio estimator [243] for MI estimation. Subsequent works improved the performance in more complicated cases such as discrete-continuous mixtures [184, 55], segmentation [318, 314, 287] and continue learning [270, 268]. Recently, MINE [16, 82] showed that the mutual information between high dimensional continuous random variables can be estimated by gradient descent over neural networks.

3.5 Conclusion

In this Chapter, we propose the structure consistency constraint (SCC) to improve the structure consistency in pixel-wise level for unsupervised image-to-image translation. To enable efficient estimation of our constraint, we propose an expression of mutual information called relative Squared-loss Mutual Information(rSMI) with an analytical estimation method. We evaluate our model quantitatively in a wide range of applications.

The experimental results demonstrate that SCC can achieve high-quality translation to maintain images' geometry in the original domain.

CHAPTER 4

A Relational Intervention Approach for Unsupervised Dynamics Generalization in Model-Based Reinforcement Learning

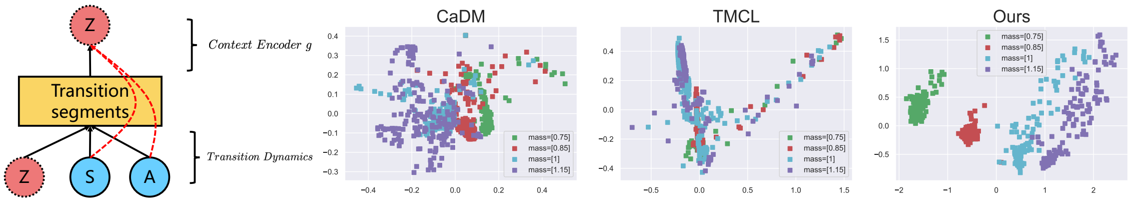
The generalization of model-based reinforcement learning (MBRL) methods to environments with unseen transition dynamics is an important yet challenging problem. Existing methods try to extract environment-specified information Z from past transition segments to make the dynamics prediction model generalizable to different dynamics. However, because environments are not labelled, the extracted information inevitably contains redundant information unrelated to the dynamics in transition segments and thus fails to maintain a crucial property of Z: Z should be similar in the same environment and dissimilar in different ones. As a result, the learned dynamics prediction function will deviate from the true one, which undermines the generalization ability. To tackle this problem, we introduce an interventional prediction module to estimate the probability of two estimated $\hat{z}_i, \hat{z}_j$ belonging to the same environment. Furthermore, by utilizing the Z's invariance within a single environment, a relational head is proposed to enforce the similarity between $\hat{Z}$ from the same environment. As a result, the redundant information will be reduced in $\hat{Z}$. We empirically show that $\hat{Z}$ estimated by our method enjoy less redundant information than previous methods, and such $\hat{Z}$ can significantly reduce dynamics prediction errors and improve the performance of model-based RL methods on zero-shot new environments with unseen dynamics. The codes of this method are available at `https://github.com/CR-Gjx/RIA`.

4.1 Introduction

Reinforcement learning (RL) has shown great success in solving sequential decision-making problems, such as board games [226, 227, 215], computer games (*e.g.* Atari, StarCraft II) [182, 225, 260], and robotics [138, 22]. However, solving real-world problems with RL is still a challenging problem because the sample efficiency of RL is low while the data in many applications is limited or expensive to obtain [63, 158, 159, 117]. Therefore, model-based reinforcement learning (MBRL) [94, 103, 215, 312, 78, 76, 75, 137], which explicitly builds a predictive model to generate samples for learning RL policy, has been widely applied to a variety of limited data sequential decision-making problems.

However, the performance of MBRL methods highly relies on the prediction accuracy of the learned environmental model [94]. Therefore, a slight change of environmental dynamics may cause a significant performance decline of MBRL methods [135, 187, 219]. The vulnerability of MBRL to the change of environmental dynamics makes them unreliable in real world applications. Taking the robotic control as an example [187, 292, 206, 68, 22, 204, 289], dynamics change caused by parts damages could easily lead to the failure of MBRL algorithms. This problem is called the *dynamics generalization* problem in MBRL, where the training environments and test environments share the same state $\mathcal{S}$ and action space $\mathcal{A}$ but the transition dynamics between states $p(s_{t+1}|s_t, a_t)$ varies across different environments. Following previous works [200, 63], we focus on the **unsupervised dynamics generalization** setting, *i.e.* the id or label information of dynamics function in training MDPs is not available. This setting appears in a wide range of applications where the information of dynamics function is difficult to obtain. For example, in healthcare, patients may respond differently to the same treatment, *i.e.*, $p(s_{t+1}|s_t, a_t)$ varies across patients. However, it is difficult to label which patients share similar dynamics.

To build a generalized dynamics prediction function that can generalize to different transition dynamics, the shift of the transition dynamics can be modelled as the change



(a) (b) (c) (d)

FIGURE 4.1. (a) The illustration of why historical states and actions are encoded in environment-specified factor Z, (b)(c)(d) The PCA visualizations of estimated context (environmental-specific) vectors in **Pendulum** task, where the dots with different colors denote the context vector (after PCA) estimated from different environments. More visualization results are given at Appendix C0.13.

of unobserved factors across different environments, *i.e.* there are hidden environment-specified factors $Z \in \mathcal{Z}$ which can affect the environmental dynamics. This is analogous to the human intuition to understand the change of dynamics, *e.g.* patients may show different responses to identical treatments because the differences of their gene sequences can affect how well they absorb drugs [278]. It is natural to assume that Z is the same in a single environment but varies across different environments. As such, these unobserved environment-specified factors do not violate the nature of MDP in a single environment, but their changes can affect the dynamics functions across environments. Therefore, the dynamics function $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ can naturally be augmented by incorporating Z to be $f : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \rightarrow \mathcal{S}$ [206, 320, 135, 219].

Learning the augmented dynamics function is difficult because the environment-specified factor Z is unobservable. Previous methods [206, 320, 135] try to extract information from historical transition segments and use it as a surrogate for Z (Figure 4.1a) . However, in the unsupervised dynamics generalization setting, the extracted information from historical transition segments inevitably contains redundant information unrelated to the dynamics. The redundant information would cause the surrogate for Z to lose a crucial property that characterizes Z: Z should be similar in the same environment and dissimilar in different environments. As shown in Figure 4.1b, the environment-specified information $\hat{Z}$ learned by CaDM [135] does not form clear clusters for different environments. Because the learned $\hat{Z}$ fails to represent environmental information, the learned dynamics function will deviate from the true one, which undermines the generalization ability. To alleviate this problem, TMCL [219] directly clusters the environments by introducing multiple prediction heads, *i.e.* multiple prediction functions. However, TMCL needs to choose the proper prediction head for each new environment, making it hard to be deployed into the scenario with consistently changing environments, *e.g.* robots walk in the terrain which is constantly changing. To avoid adaptation at the deployment time, we thus need to learn a single generalized prediction function $\hat{f}$. To ensure that $\hat{f}$ can learn modals of transition dynamics in different environments, we need to cluster Z according to their belonging environments.

In this Chapter, we provide an explicit and interpretable description to learn Z as a vector $\hat{Z}$ (*i.e.* the estimation of Z) from the history transition segments. To cluster $\hat{Z}$ from the same environment, we introduce a relational head module as a learnable function to enforce the similarity between $\hat{Z}$s learned from the same environments. However, because environment label is not available, we can only cluster the $\hat{Z}$s from the same trajectory, so we then propose an interventional prediction module to identify the probability of a pair of $\hat{z}_i, \hat{z}_j$ belonging to the same environment through estimating $\hat{Z}$'s direct causal effect on next states prediction by do-calculus [196]. Because Zs from the same environment surely have the same causal effect, we can directly maximize the similarity of $\hat{Z}$s with the similar causal effect using the relational head, and thus can cluster $\hat{Z}$ according to the estimated environmental similarity and alleviate the redundant information that varies in an environment, *e.g.* historical states and actions. In the experiments, we evaluate our method on a range of tasks in OpenAI gym [24] and Mujoco [254], and empirically show that $\hat{Z}$ estimated by our method enjoy less redundant information than baselines. The experimental results show that our method significantly reduces the model prediction errors and outperforms the state-of-art model-based RL methods **without any adaptation step** on a new environment, and even achieve comparable results with the method directly cluster $\hat{Z}$ with the true environment label.

4.2 Related Work

Dynamics Generalization in MBRL Several meta-learning-based MBRL methods are proposed [187, 186, 210, 86] to adapt the MBRL into environments with unseen dynamics by updating model parameters via a small number of gradient updates [49] or hidden representations of a recurrent model [44], and then [265] proposes a graph-structured model to improve dynamics forecasting. [10] focuses on the offline setting and proposes an augmented model method to achieve zero-shot generalization. [135, 219] try to learn a generalized dynamics model by incorporating context information or clustering dynamics implicitly using multi-choice learning, aiming to adapt any dynamics without training. However, how to explicitly learn the meaningful dynamics change information remains a big challenge.

Relational Learning Reasoning relations between different entities is an important way to build knowledge of this world for intelligent agents [109]. In the past decades, relational paradigm have been applied to a wide range of deep learning-based application, *e.g.*, reinforcement learning [304], question-answer [211, 207], graph neural network [15], sequential streams [212], few-shot learning [241], object detection [85] and self-supervised learning [194]. Different from previous methods that perform binary relational reasoning on entities, our method can also perform multiplies relations between entities through the learned similarity of entities, and thus can learn more compact and meaningful entity representation.

Causality in Reinforcement Learning Many works focus on the intersection area of reinforcement learning and causal inference. For example, some works aims to alleviate the causal confusion problem in the imitation learning [74, 309, 126], batch learning [12], and partial observability settings [50, 104, 306, 158] in the online environment [166, 308], [264] also try to apply causal inference in the offline setting, where the observational data is always confounded. [136, 13, 129, 185, 261] also explore how to design an optimal intervention policy in bandits or RL settings. In addition, [305, 307] improve the generalization ability of state abstraction. Different from these methods, we focus on the setting of unsupervised dynamics generalization, and measure the direct causal effect

[197] between $\hat{Z}$ and the next state to estimate the probability of them belonging to the same environment.

4.3 Method

4.3.1 Problem Setup

The standard reinforcement learning task can be formalized as a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, p, \gamma, \rho_0)$ over discrete time [201, 242], where $\mathcal{S}$, $\mathcal{A}$, $\gamma \in (0, 1]$, ρ_0 are state space, action space, the reward discount factor, and the initial state distribution, respectively. The reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ specifies the reward at each timestep t given s_t and a_t, and transition dynamics $p(s_{t+1}|s_t, a_t)$ gives the next state distribution conditioned on the current state s_t and action a_t. The goal of RL is to learn a policy $\pi(\cdot|s)$ mapping from state $s \in \mathcal{S}$ over the action distribution to maximize the cumulative expected return over timesteps $\mathbb{E}_{s_t \in \mathcal{S}, a_t \in \mathcal{A}}[\sum_{t=0}^{\infty} \gamma^t \ r(s_t, a_t)]$. In model-based RL, a model f is used to approximate the transition dynamics p, and then f can provide training data to train policy π or predict the future sequences for planning. Benefiting from data provided by learned dynamics model f, model-based RL has higher data efficiency and better planing ability compared with model-free RL.

Here we consider the unsupervised *dynamics generalization* problem in model-based RL, where we are given K training MDPs $\{\mathcal{M}_i^{tr}\}_{i=1}^{K}$ and L test MDPs $\{\mathcal{M}_j^{te}\}_{j=1}^{L}$ that have the same state and action space but disjoint dynamics functions, and we randomly sample several MDPs from training MDPs in each training iteration. We assume that all these MDPs have a finite number of dynamics functions, meaning that the MDPs can be categorized into a finite number of environments and the MDPs in each environment share the same dynamics function but the environment id of MDPs is unavailable in the training process. In the context of model-based RL, how to learn a generalized dynamics model f is the key challenge to solve unsupervised *dynamics generalization* problem.

4.3.2 Overview

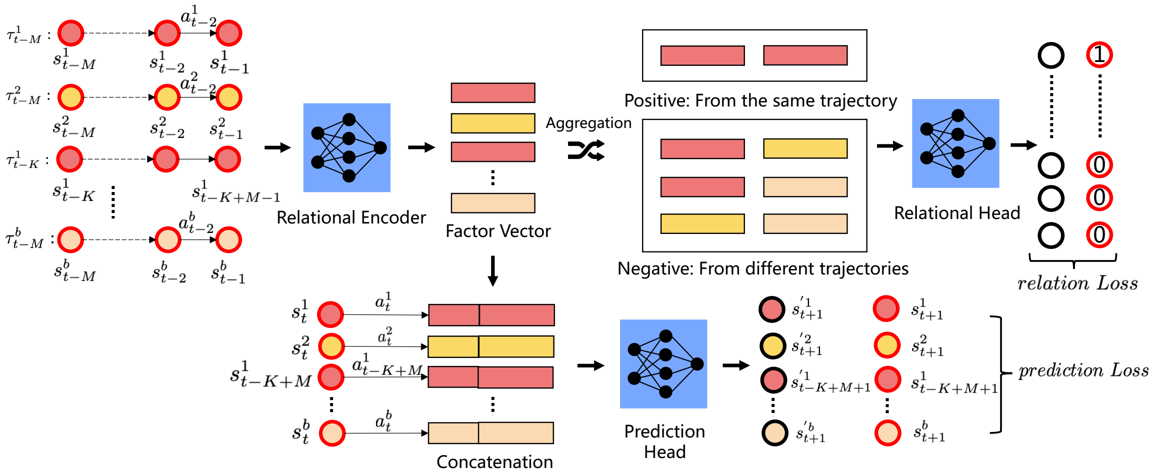


FIGURE 4.2. An overview of our Relational Intervention approach, where Relational Encoder, Prediction Head and Relational Head are three learnable functions, and the circles denote states (Ground-Truths are with red boundary, and estimated states are with black boundary), and the rectangles denote the estimated vectors. Specifically, $prediction\ Loss$ enables the estimated environmental-specified factor can help the Prediction head to predict the next states, and the $relation\ Loss$ aims to enforce the similarity between factors estimated from the same trajectory or environments.

As analyzed in Section 4.1, we can incorporate the environment-specified factors $Z \in \mathcal{Z}$ into the dynamics prediction process to generalize the dynamic functions on different environments, *i.e.* extending the dynamics function from $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ to $f : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to \mathcal{S}$. Because Z is the same within an environment, we expect estimated $\hat{Z}$s from the same environment are similar while those from different environments are dissimilar. Therefore, f models the commonalities of the transition dynamics in different environments and Z models the differences. In the supervised dynamics generalization setting, where the environment id is given, one can easily learn Z by using metric losses, *e.g.*, CPC [189] and relation loss [194] to enforce that the estimated $\hat{Z}$s are similar in the same environment and dissimilar in different environments. However, since the environment label is unavailable in the unsupervised setting, we have to simultaneously learn Z and discover the cluster structures. To this end, we propose an intervention module to measure the similarities between each pair of $\hat{z}_i$ and $\hat{z}_j$ as the probability of them belonging to the same environment. Furthermore, we then introduce a relational head to aggregate $\hat{Z}$s with high probability using relation loss. By simultaneously updating the dynamics prediction and the relation loss, we can cluster $\hat{Z}$s from the same environment, and learn an augmented dynamics prediction model $\hat{f}$. Next, we will give details about our relational intervention approach.

4.3.3 Relational Context Encoder

To learn the environment-specified factor Z of each environment, we firstly introduce a relational encoder g parameterized by ϕ. Similar to previous methods [187, 206, 320, 135, 219], we use the past transition segments $\tau_{t-k:t-1} = \{(s_{t-k}, a_{t-k}), ..., (s_{t-1}, a_{t-1})\}$ as the input of g to estimate its corresponding $\hat{z}_{t-k:t-1}$:

$$\hat{z}_{t-k:t-1} = g(\tau^i_{t-k:t-1}; \phi).$$

After obtaining environment-specified $\hat{z}_{t-k:t-1}$ at timestep t, we incorporate it into the dynamics prediction model $\hat{f}$ to improve its generalization ability on different dynamics by optimizing the objective function following [135, 219, 94]:

$$\mathcal{L}^{pred}_{\theta,\phi} = -\frac{1}{N}\sum_{i=1}^{N} \log\ \hat{f}(s^i_{t+1}|s^i_t, a^i_t, g(\tau^i_{t-k:t-1}; \phi); \theta), \tag{4.1}$$

where k is the length of transition segments, t is the current timestep and N is the sample size. In practice, we sub-sample a mini-batch of data from the whole dataset to estimate (4.1) and use stochastic gradient descent to update the model parameters.

However, as analyzed in Section 4.3.2, the vanilla prediction error (4.1) is not sufficient to capture environment-specified Z of each environment, and even introduce redundant information into it. In order to eliminate the redundant information within transition segments and preserve the trajectory invariant information, we introduce a relational head [194] as a learnable function h to pull factors $\hat{Z}$ from the same trajectory together and push away those from different trajectories. Concretely, the estimated $\hat{z}^i_{t-k:t-1}$ in a mini-batch will be firstly aggregated as pairs, *e.g.* concatenate two factors as $[\hat{z}^i, \hat{z}^j]$, and the pairs having two factors from the same trajectory are seen as positives, and vice versa. Then the relational head h parameterized by φ takes a pair of aggregated factors as input to quantify the similarity of given two factors and returns a similarity score $\hat{y}$. To increase the similarity score $\hat{y}$ of positive pairs and decrease those negatives, we minimize the following objective:

$$\mathcal{L}^{relation}_{\varphi,\phi} = -\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}\Big[\, y^{i,j}\cdot\log\, h([\hat{z}^i,\hat{z}^j];\varphi)+(1-y^{i,j})\cdot\log\,(1-h([\hat{z}^i,\hat{z}^j];\varphi))\Big], \tag{4.2}$$

where $y^{i,j} = 1$ stands for positive pairs, and $y^{i,j} = 0$ stands for negatives. Because the positive pairs have two factors belonging to the same trajectory, optimizing (2) can increase the similarity of $\hat{Z}$s estimated from the same trajectory, and push away those factors estimated from different trajectories in their semantical space. Therefore, by optimizing (4.2), the information that is invariant within a trajectory will be encoded into $\hat{Z}$ and the redundant information in transition segments will be reduced. (4.2) can also be interpreted from the perspective of mutual information, if we regard the $\hat{Z}$s from the same trajectory as the positive pairs, optimizing (4.2) can be seen as maximizing the mutual information between $\hat{Z}$s from the same trajectory (Please refer to [256] and Appendix D1.7), and thus preserve the invariant information with the same trajectory. However, estimating trajectory invariant information is insufficient because the estimated $\hat{Z}$s in the same environment will also be pushed away, which may undermine the cluster compactness for estimated $\hat{Z}$s.

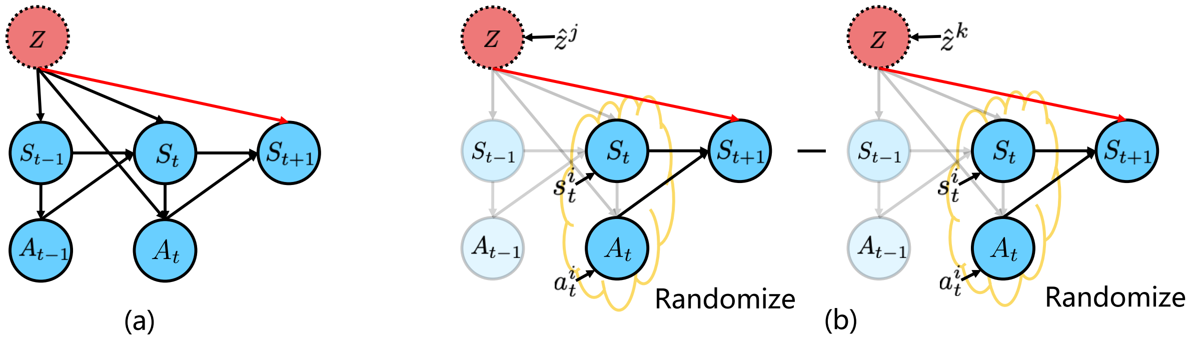


FIGURE 4.3. (a) The illustration of causal graph, and the red line denotes the direct causal effect from Z to S_{t+1}. (b) The illustration of estimating the controlled causal effect.

4.3.4 Interventional Prediction

Because the environment id of a trajectory is unknown, we cannot directly optimize relational loss (4.2) to cluster $\hat{Z}$ within an environment. We propose an interventional prediction method to find the trajectories belonging to the same environment. Here we formalize the dynamics prediction model using a graphical causal model, and the causal graph is illustrated as Figure 4.3 (a), where the next state S_{t+1} is caused by the current state S_t, action A_t and $\hat{Z}$, and the dynamics prediction model f represents the causal mechanism between them. Because Z from the same environment should have the same influence on the states, and thus they should have the same causal effect on the next state S_{t+1} if given S_t and A_t under the causal framework. As such, we can find estimated $\hat{Z}$ belonging to the same environment by measuring the similarity of their causal effect on S_{t+1}. As Figure 4.3 (a) shows, there are multiple paths from Z to S_{t+1} and we roughly categorize them into two categories. The first category is the direct path between Z and S_{t+1} (shown as Figure 4.3). The second category contains all the indirect paths where Z influences S_{t+1} via previous states and actions. However, because the mediator in other paths *e.g.* S_t, A_t, may amplify or reduce the causal effect of Z, we only consider the direct path from Z to the next state(denote by the red line at Figure 4.3 (a)), which means that we need to block all paths with meditors from $\hat{Z}$ to S_{t+1}. By means of do-calculus [196], we can estimate the direct causal effect of changing $Z = \hat{z}^j$ to $Z = \hat{z}^k$ on S_{t+1} through calculating the controlled direct effect (CDE) [197] by intervening mediators and $\hat{Z}$:

$$CDE_{\hat{z}^j,\hat{z}^k}(s_t, a_t) = \mathbb{E}[S_{t+1}|do(S_t = s_t, A_t = a_t), do(Z = \hat{z}^j)] \tag{4.3}$$

$$- \mathbb{E}[S_{t+1}|do(S_t = s_t, A_t = a_t), do(Z = \hat{z}^k)] \tag{4.4}$$

$$= \mathbb{E}[S_{t+1}|S_t = s_t, A_t = a_t, Z = \hat{z}^j] - \mathbb{E}[S_{t+1}|S_t = s_t, A_t = a_t, Z = \hat{z}^k], \tag{4.5}$$

where do is the do-calculus [196]. There is no arrow entering $\hat{Z}$, so the do operator on $\hat{Z}$ can be removed. Also, since there is no confounder between the mediators (S_t, A_t) and S_{t+1}, so we can remove the do operator of them as well, and the equation become as (D.3). Because the direct causal effects may differ for different values of S_t and A_t, we should sample S_t and A_t independently of Z, *i.e.* sampling S_t and A_t [199] uniformly to get the average controlled direct causal effect from $\hat{Z}$ to S_{t+1}. However, if we use the uniformly generated S_t and A_t, the sampled distribution may differ from the training distribution, resulting in inaccurate the next state prediction. As such, we directly sample S_t and A_t from the observational data. For the convenience of optimization, we only use a mini-batch of S_t and A_t pairs (s_t^i, a_t^i), and concatenate them with $\hat{z}^j$ and $\hat{z}^k$ to calculate the average controlled direct effect under $\hat{f}$:

$$ACDE_{\hat{z}^j,\hat{z}^k} = \frac{1}{N}\sum_{i=1}^{N}|CDE_{\hat{z}^j,\hat{z}^k}(s_t^i, a_t^i)|, \tag{4.6}$$

where N is the batch size, j and k are the id of $\hat{Z}$ estimated from two transition segments. Specifically, because the factors $\hat{z}$ estimated from the same trajectory should be the same, and thus we minimize their controlled direct effect D.4 as $\mathcal{L}^{dist}$ between them in the optimization process. Now we can use the calculated $ACDE_{\hat{z}^j,\hat{z}^k}$ as the semantic distance $d^{j,k}$ between estimated $\hat{z}^i$ and $\hat{z}^j$, and thus we can aggregate factors $\hat{Z}$ estimated from similar trajectories by the proposed relational head h. As such, we apply a transformation to convert distance metric $d^{j,k}$ to a similarity metric $w \in (0, 1]$, which is $w^{j,k} = exp(\frac{-d^{j,k}}{\beta})$, where β is a factor controlling the sensitivity of the distance metric. Specifically, because the scale and size of state varies in different tasks, *e.g.* 3 dims in Pendulum but 20 dims in Half-Cheetah, the optimal β may vary in different task. As such, we apply the normalization in the distance metric d, *i.e.*, normalize d with batch variance, to convert it as a relative distance within a single task, thus making the optimal β stable in different tasks. Then we can directly aggregate similar trajectories by extending the loss function (4.2) as follows:

$$\begin{aligned}\mathcal{L}_{\varphi,\phi}^{i-relation} = & -\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}\Big[\, [y^{i,j} + (1-y^{i,j})\cdot w^{i,j}]\cdot\log\ h([\hat{z}^i, \hat{z}^j]; \varphi) \\ & + (1-y^{i,j})\cdot(1-w^{i,j})\cdot\log\,(1-h([\hat{z}^i, \hat{z}^j]; \varphi))\Big],\end{aligned} \tag{4.7}$$

where the first term indicates that the factors from the different trajectories can be aggregated with the similarity weight w and 1 those from the same trajectory, and the second term means that factors from different trajectories should be pushed with each other with weight $1 - w$. Similar to the analysis in section 3.3, optimizing the loss function (6) can increase the similarity between $\hat{Z}$ with weight w, and push away from them with the weight $1 - w$. Because $\hat{Z}$s estimated from the same environment have similar effects, these factors will be assigned with high similarities (estimated by the intervention operation of the Chapter). By simultaneously updating the prediction loss (5.4) and intervention relation loss 5.3, estimated $\hat{Z}$s within the same environment will be aggregated, and the learned dynamics function $\hat{f}$ can learn the modals of transition dynamics according to the $\hat{Z}$ in different clusters. The training procedure of our approach can refer to Algorithm process in Appendix D1.2.

4.4 Experiments

In this section, we conduct experiment to evaluate the performance of our approach by answering the following questions:

- Can our approach reduce the dynamics prediction errors in model-based RL? (Section 4.4.2.1)
- Can our approach promote the performance of model-based RL on environments with unseen dynamics? (Section 4.4.2.2)
- Can our approach learn the semantic meaningful dynamics change? (Figuer 4.1 and AppendixC0.13)
- Is the similarity of w measured by the intervention module reasonable? (Appendix C0.7)
- Can solely relational learning improve the performance of model-based RL? (Section 4.4.3)

4.4.1 Enviromental Setup

Implementation Details Our approach includes three learnable functions, including relational encoder, relational head and prediction head. All three functions are constructed with MLP and optimized by Adam [116] with the learning rate 1e-3. During the training procedure, the trajectory segments are randomly sampled from the same trajectory to break the temporal correlations of the training data, which was also adopted by [219, 269, 271]. Specifically, the length of the transition segments, *i.e.*, k, is 10. All implementation details can be found in Appendix D1.1.

Datasets Following the previous methods [135, 219], we perform experiments on a classic control task (Pendulum) from OpenAI gym [24] and simulated robotics control tasks (HalfCheetah, Cripple-HalfCheetah, Ant, Hopper, Slim-Humanoid) from Mujoco physics engine [254].

Dynamics Settings To change the dynamics of each environment, we follow previous methods [320, 190, 135, 219] to change the environmental parameters (*e.g.* length and mass of Pendulum) and predefine them in the training and test environmental parameters lists. At the training time, we randomly sample the parameters from the training parameter list to train our relational context encoder and dynamics prediction model. Then we test our model on the environments with unseen dynamics sampled from the test parameter list. Specifically, the predefined parameters in the test parameter list are outside the training range. The predefined training and test parameter lists for each task are the same with [135], and all details are given in Appendix D1.1.

Planning Following [135, 219], we use the model predictive model (MPC) [167] to select actions based on learned dynamics prediction model, and assume that the reward functions of environments are known. Also, the cross-entropy method (CEM) [39] is used to optimize action sequences for finding the best performing action sequences.

Baselines We compare our approach with following state-of-the-art model-based RL methods on dynamics generalization. Also, to show the performance gap between our method and supervised dynamics generalization, we perform the method using true environment label to cluster Z.

- Probabilistic ensemble dynamics model (PETS) [127]: PETS employs an probabilistic dynamics models to capture the uncertainty in modeling and planning.
- Meta learning based model-based RL (ReBAL and GrBAL) [186, 187]: These methods train a dynamics model by optimizing a meta-objective [49], and

update the model parameters by updating a hidden with a recurrent model or by updating gradient updates at the test time.

- Context-aware dynamics model (CaDM) [135]: This method design several auxiliary loss including backward and future states prediction to learn the context from transition segments.
- Trajectory-wise Multiple Choice Learning (TMCL) [219]: This method is the state-of-the-art model-based RL method on dynamics generalization, which introduces the multi-choice learning to cluster environments. TMCL needs the adaptation in the test procedure, while our method does not, so we also report the performance of TMCL without adaptation in Figure C.1 for the fair comparison.
- True Label: The method uses our relational head to cluster $\hat{Z}$ with the true environment label (not the ground-truth of Z). All hyperparameters are same with our method for the fair comparison.

4.4.2 Performance Comparisons with Baselines

4.4.2.1 Prediction Error Comparisons

We first evaluate whether the dynamics model trained by our methods can predict next-states more accurately or not. Figure 4.4 shows that the average dynamics prediction error of dynamics prediction models trained by three methods (CaDM [135], TMCL [219] and ours). We can see that the dynamics model trained by our relational intervention method has superior prediction performance over other state-of-the-art methods, achieving the lowest prediction errors on almost all six tasks. Specifically, the prediction errors of our model are lower than others by a large margin in Hopper and Pendulum, outperforming the state-of-the-art methods by approximately 10%.

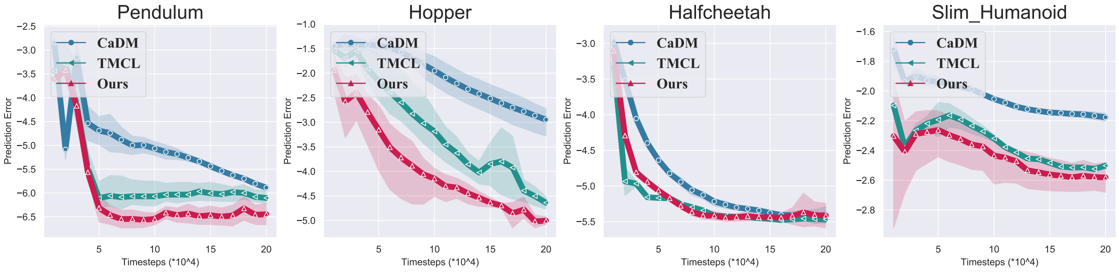


FIGURE 4.4. The average prediction errors of dynamics models on training environments during training process (over three times). Specifically, the x axis is the training timesteps and y axis is the *log* value of average prediction prediction errors. More figures are given at Appendix C0.8.

TABLE 4.1. The average rewards of baseline model-based RL methods and ours on test environments with unseen dynamics. Here we report the average rewards over three runs (ours is ten). Specifically, the results of methods with $*$ are from the paper [135].

	PETS*	ReBAL*	GrBAL*	CaDM	TMCL	Ours	↑ Ratio
Pendulum	-1103	-943.6	-1137.9	-713.95±21.1	-691.2±93.4	**-587.5**±64.4	15.0%
Ant	965.883.5	63.0	44.7	1660±57.8	2994.9±243.8	**3297.9**±159.7	10.1%
Hopper	821.2	846.2	621	845.2±20.41	999.35±22.8	**1057.4**±37.2	5.8%
HalfCheetah	1720.9	52	-69.1	5876.6±799.0	9039.6±1065	**10859.2**±465.1	20.1%
C_HalfCheetah	1572	868.7	2814	3656.4±856.2	3998.8±856.2	**4819.3**±409.3	20.5%
Slim_Humanoid	784.5	97.25	-480.7	859.1±24.01	2098.7±109.1	**2432.6**±465.1	15.9%

4.4.2.2 Performance Comparisons

Then we evaluate the generalization of model-based RL agents trained by our methods and baselines on test environments with unseen dynamics. Following the setting of [219], we perform experiments three runs (ours with 10 runs to reduce random errors), and give the mean of rewards at Table 4.1. We can see that the meta-learning based methods [186, 187] do not perform better than vanilla PETS [127], while methods [135, 219] that aim to learn a generalized dynamics prediction model are superior to others significantly. Among which our approach achieves the highest rewards on all six tasks among all methods. Figure D.1 shows the mean and standard deviation of average rewards during the training procedure, indicating that the performance of our methods is better than the other two methods consistently at the training time, which is sufficient to show the superiority of our method over other methods. A fair comparison between TMCL (no adaptation) and our method can be found at Appendix C0.6. In addition, we observe that our method achieves comparable results with the method directly cluster $\hat{Z}$ using the truth environment label, which indicates that our intervention module actually can assign high similarities into $\hat{Z}$s estimated from the same environment in an unsupervised manner. We also observe the same results in the similarity visualization in the Appendix C0.7, where we find that $\hat{Z}$s from the same environment are assigned significant higher similarities than those pairs from different environments.

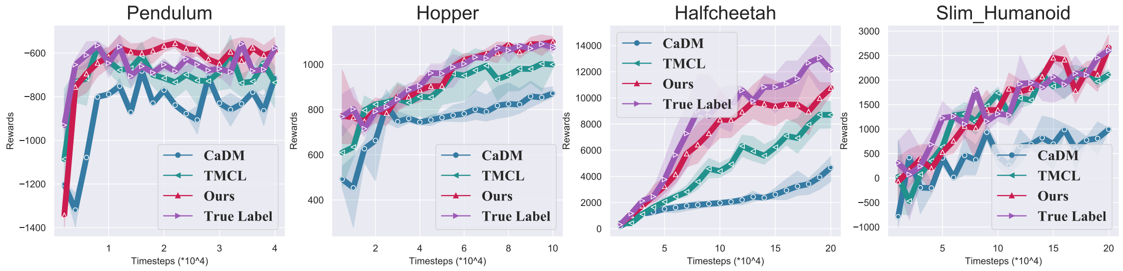
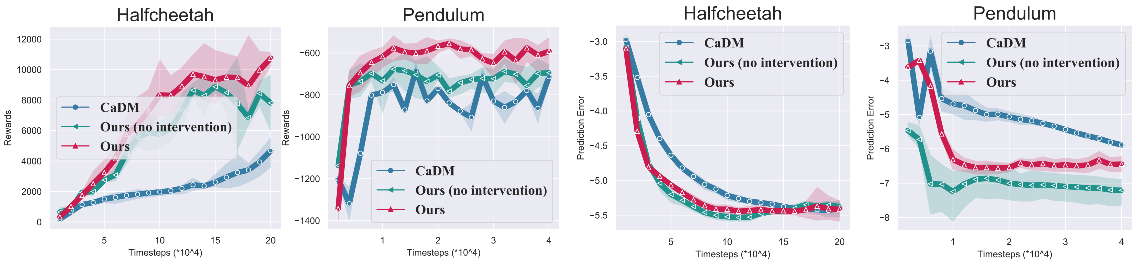


FIGURE 4.5. The average rewards of trained model-based RL agents on unseen test environments. The results show the mean and standard deviation of returns averaged over three runs. The fair comparison between TMCL (no adaptation) and our method can be found in Appendix C0.6

4.4.3 Ablation Study

In this section, we evaluate the effect of the proposed relation head and intervention prediction on the generalization improvement, respectively. Because the intervention prediction is based on the relational head, we compare the performance of our approach with and without the intervention. As Figure 4.6a and 4.6b show, after incorporating the relational head and intervention prediction, the performance of model-based agents and the generalization of the dynamics prediction model are both improved. However, although the model without the intervention module has lower prediction errors in the Pendulum task, it also has lower rewards than the whole model. One possible reason is that the Pendulum is simple for the dynamics prediction model to learn, and thus the dynamics prediction model with the vanilla relational head is a little over-fitting on the training environments (Please refer to Appendix C0.9), limiting the performance improvement. This phenomenon confirms the importance of our intervention prediction on reducing the trajectory-specified redundant information.



(a) **(b)**

FIGURE 4.6. (a) The average rewards of trained model-based RL agents on unseen environments. The results show the mean and standard deviation of returns averaged over three runs. (b) The average prediction errors over the training procedure. Prediction errors on test environments are given in Appendix C0.9

4.5 Conclusion

In this Chapter, we propose a relational intervention approach to learn a generalized dynamics prediction model for dynamics generalization in model-based reinforcement learning. Our approach models the dynamics change as the variation of environment-specified factor $\mathcal{Z}$ and explicitly estimates $\mathcal{Z}$ from past transition segments. Because environment label is not available, it is challenging to extract $\mathcal{Z}$ from transition segments without introducing additional redundant information. We propose an intervention module to identify the probability of two estimated factors belonging to the same environment, and a relational head to cluster those estimated $\hat{Z}$s are from the same environments with high probability, thus reducing the redundant information unrelated to the environment. By incorporating the estimated $\hat{Z}$ into the dynamics prediction process, the dynamics prediction model has a stronger generalization ability against the change of dynamics. The experiments demonstrate that our approach can significantly reduce the dynamics prediction error and improve the performance of model-based agents on new environments with unseen dynamics.

CHAPTER 5

Hierarchical Prototypes for Unsupervised Dynamics Generalization in Model-Based Reinforcement Learning

Generalization remains a central challenge in model-based reinforcement learning. Recent works attempt to model the environment-specific factor and incorporate it as part of the dynamic prediction to enable generalization to different contexts. By estimating environment-specific factors from historical transitions, earlier research was unable to clearly distinguish environment-specific factors from different environments, resulting in poor performance. To address this issue, we introduce a set of environment prototypes to represent the environmental-specified representation for each environment. By encouraging learned environment-specific factors to resemble their assigned environmental prototypes more closely, the discrimination of factors between different environments will be enhanced. To learn such prototypes in the unsupervised manner, we propose a hierarchical prototypical method which first builds trajectory embeddings according to the trajectory label information, and then hierarchically constructs environmental prototypes from trajectory prototypes sharing similar semantics. Experiments demonstrate that environment-specific factors estimated by our method have superior clustering performance and can improve MBRL's generalisation performance in six environments consistently.

5.1 Introduction

Reinforcement learning (RL) has achieved great success in solving sequential decision-making problems, *e.g.*, board games [226, 227, 215], computer games [182, 225, 260], and robotics [138, 22], but it still suffers from the low sample efficiency problem, making it challenging to solve real-world problems, especially for those with limited or expensive data [63, 158, 159, 117].In contrast, model-based reinforcement learning (MBRL) [94, 103, 215, 312, 78, 76, 75, 137] has recently received wider attention, because it explicitly builds a predictive model and can generate samples for learning RL policy to alleviate the sample inefficiency problem.

As a sample-efficient alternative, the model-based RL method derives a policy from the learned environmental dynamics prediction model. Therefore, the dynamics model's prediction accuracy is highly correlated with policy quality [94]. However, it has been evidenced that the learned dynamics prediction model is not robust to the change of environmental dynamics [135, 219, 70], and thus the agent in model-based RL algorithms has a poor generalization ability on the environments with different dynamics. Such a vulnerability to the change in environmental dynamics makes model-based RL methods unreliable in real-world applications where the factors that can affect dynamics are partially observed. For example, the friction coefficient of the ground is usually difficult to measure, while the changes in it can largely affect the dynamics when controlling a robot walking on the grounds, leading to the performance degradation of an agent trained by model-based RL methods [289, 68, 186].

Recent Studies [219, 187, 135, 70] have demonstrated that incorporating environmental factor Z into dynamics prediction facilitates the generalisation of model-based RL methods to unseen environments. However, environmental factors are unobservable in the majority of applications; for instance, the friction coefficient is not available for robots. Therefore, estimating semantical meaningful Z for each environments is the first step for generalization of model-based RL. However, it is not easy to implement, because the environment is hard to label. For example, it is impractical to measure the friction coefficient of every road. Without the label information of environments, Zs estimated from previous methods [219, 187, 135, 70] cannot form clear clusters for different environments as Figure 5.3 shows. These entangled Zs cannot represent the distinct environmental specific information, and thus may deviate the learned dynamics prediction function from the true one, resulting in the poor generalization ability.

In this paper, we propose a hierarchical prototypical method (HPM) with the objective of learning an environment-specific representation with distinct clusters. By representing environment-specific information semantically meaningfully, HPM learns more generalizable dynamics prediction function. To achieve this, our method propose to construct a set of environmental prototypes to capture environment-specific information for each environment. By enforcing the estimated $\hat{Z}$ to be more similar to its respective environmental prototypes and dissimilar to other prototypes, the estimated $\hat{Z}$s can form compact clusters for the purpose of learning a generalizable dynamics prediction function. Because environmental labels are not available, we cannot construct environmental prototypes directly. To address this issue, we begin by developing easily-learned trajectory prototypes based on the trajectory label. Then, environmental prototypes can be created by merging trajectory prototypes with similar semantics, as suggested by the natural hierarchical relationship between trajectory and environment.

With the built hierarchical prototypical structure, we further propose a prototypical relational loss to learn Z from past transitions. Specifically, we not only aggregate the $\hat{Z}$s with similar causal effects by optimizing the relational loss [70] but also aggregate $\hat{Z}$ with its corresponding trajectory and environmental prototypes via the relational loss. In addition, to alleviate the over-penalization of semantically similar prototypes, we propose to penalize prototypes adaptively with the intervention similarity. In the experiments, we evaluate our method on a range of tasks in OpenAI gym [24] and Mujoco [254]. The experimental results show that our method can form more clear and tighter clusters for $\hat{Z}$s, and such $\hat{Z}$s can improve the generalization ability of model-based RL methods and achieve state-of-art performance in new environments with different dynamics without any adaptation step.

5.2 Related Work

Model-based reinforcement learning With the learned dynamics prediction model, Model-based Reinforcement Learning (MBRL) takes advantage of high data efficiency. The learned prediction model can generate samples for training policy [46, 277] or planning ahead in the inference [7, 137, 253]. Therefore, the performance of MBRL highly relies on the prediction accuracy of the dynamics predictive model. To improve the predictive model's accuracy of MBRL, several methods were proposed, such as ensemble methods [35], latent dynamics model [76, 75, 215], and bidirectional prediction [128]. However, current predictive methods are still hard to generalize well on unseen dynamics, which hinders the application of MBRL methods in the real-world problems.

Dynamics generalization in model-based reinforcement learning To adapt the MBRL to unknown dynamics, meta-learning methods [187, 186, 210] attempted to update model parameters by updating a small number of gradient updates [49] or hidden representations of a recurrent model [44]. Then, using multi-choice learning, [135, 219] attempted to learn a generalised dynamics model by incorporating environmental-specified information or clustering dynamics implicitly, with the goal of adapting any dynamics without training. Through relational learning and causal effect estimation, RIA [70] aims to explicitly learn meaningful environmental-specific information. However, the dynamics change learned by RIA still suffer from a high variance issue.

Prototypical methods By learning an encoder to embed data in a low-dimensional representation space, prototypical methods gain a set of prototypical embeddings, which are referred to as prototypes [6, 27] that form the basis of this representation space. Prototypical methods aim to derive compact data representations gathering around corresponding prototypes [144, 189, 267], which captures some basic semantic structures. Therefore, prototypical methods have been applied into many areas, *e.g.* self-supervised learning [142, 28], few-shot learning [232, 14, 228], domain adaptation [251] and continue learning [40, 299]. In the RL area, [295] ties representation learning with exploration through prototypical representations for image-based RL, while our method focuses

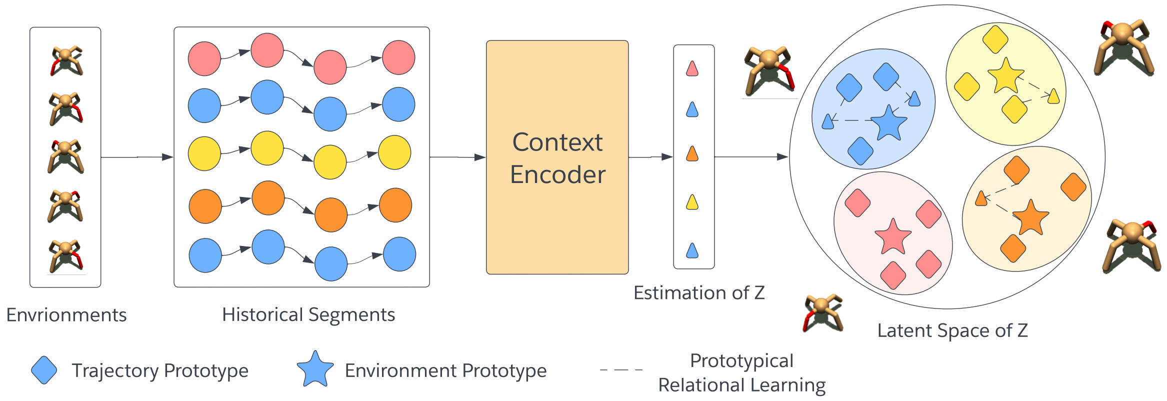


FIGURE 5.1. An overview of our Hierarchical Prototypical Method, where the context encoder estimates the environmental-specific factor $\hat{z}$ and environments includes four ants with different destroyed leg with red color. Items extracted from different environments are different colors. We construct prototypes for each trajectory and environment, and denote them as diamond and star, respectively. Each estimated $\hat{z}$ are optimized with its corresponding trajectory and environment prototype using our prototypical relational learning as dotted line shows.

on the unsupervised dynamics generalization problem in model-based RL, aiming to learn semantical meaningful dynamics change using prototypical method. Specifically, our method propose a hierarchical method to construct environmental prototypes from trajectory prototypes.

5.3 Method

In this section, we first introduce the formulation of the unsupervised dynamic generalization problem in model-based reinforcement learning. Then we present the details of how our hierarchical prototype method learns the environment-specific factors.

5.3.1 Problem setup

We formulate the standard reinforcement learning as a markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, f, \gamma, \rho_0)$ over discrete time [201, 242], where $\mathcal{S}$, $\mathcal{A}$, $\gamma \in (0, 1]$ and ρ_0 are state space, action space, the reward discount factor, and the initial state distribution, respectively. Dynamics function $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ gives the next state s_{t+1} conditioned on the current state s_t and action a_t, and reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ specifies the reward at each timestep t given s_t and a_t. The goal of RL is to learn a policy $\pi(\cdot|s)$ mapping from state $s \in \mathcal{S}$ over the action distribution to maximize the cumulative expected return $\mathbb{E}_{s_t \in \mathcal{S}, a_t \in \mathcal{A}}[\sum_{t=0}^{\infty} \gamma^t \ r(s_t, a_t)]$ over timesteps. In model-based RL, we aim to learn a prediction model $\hat{f}$to approximate the dynamics function f, and then $\hat{f}$ can generate training data to train policy π or predict the future sequences for planning. With the data provided by learned dynamics model $\hat{f}$, model-based RL has higher data efficiency and better planing ability compared with model-free RL.

In this Chapter, we consider the unsupervised *dynamics generalization* problem in model-based RL. Different from the standard reinforcement learning, there exists an unobserved variable Z that can affect the dynamics prediction function f in the *dynamics generalization* problem. The goal of *dynamics generalization* is to derive a generalizable policy from given K training MDPs $\{\mathcal{M}_i^{tr}\}_{i=0}^{K}$, and expect the policy can generalize well on L test MDPs $\{\mathcal{M}_j^{te}\}_{j=0}^{L}$. Without losing generality, we assume all MDPs share the same state and action space but preserve different factor Z.

In the context of model-based reinforcement learning, we need to learn the dynamics function before learning policy. In order to generalize the dynamic functions on different environment, we need to incorporate the unobserved variable Z into dynamics prediction process, *i.e.*, extending the dynamics function from $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ to $f : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \rightarrow \mathcal{S}$. Since Z is not available, we should estimate it from past transition segments $\tau_{t-k:t-1} = \{(s_{t-k}, a_{t-k}), ..., (s_{t-1}, a_{t-1})\}$ [219, 135, 70].

Next, we will present how our hierarchical prototypes method estimates Z, and enable it to learn the dynamics function f that can generalize to environments with unseen dynamics. In Section 5.3.2, we present how our method hierarchically constructs prototypes as a representative embedding to represent environmental-specific information for each environment. In Section 5.3.3, we describe how we update prototypes dynamically and how to estimate environmental-specific factors Z from past transition segments using prototypes. Once Z are estimated, we describe how they enable dynamics function f to generalize well environments with different dynamics.

5.3.2 Hierarchical environment prototypes construction

The objective of our method is to construct a set of prototypes to represent the environmental-specific information for each environment, and guide the context encoder to estimate environmental-specific variable Z from historical transition segments. In each training iteration, we randomly sample a trajectory from a subset of MDPs in the training MDPs. Because labels of MDPs are not available, we cannot estimate environmental prototypes directly. Furtunately, we still have the trajectory label information, and thus we can construct the prototypes for each sampled trajectory first. Specifically, we denote the prototype for j-th trajectory as c^j_{tra}. Because different trajectories may be sampled from a single environment, the trajectory prototypes from the same environment should share similar semantics for dynamics prediction. Therefore, we can construct environmental prototypes hierarchically from trajectory prototypes sharing similar semantics. In this way, environmental prototypes and trajectory prototypes form a natural hierarchical structure, and environmental prototypes can be constructed utilising trajectory label information even if no environmental label is available.

If we denote the $w^{i,j}_{tra}$ as the semantical similarity between the trajectory prototypes c^i_{tra} and c^j_{tra}, we can construct a trajectory similarity matrix w as Figure 5.2 (b) shows, where each row of w, such as w^i represents the similarity between c^i_{tra} and all other trajectory prototypes. Because it is unknown how many environments are in the sampled trajectories, we directly construct environmental prototypes c^i_{env} for each trajectory prototype c^i_{tra}. Specifically, each environmental prototype c^i_{env} is the mean of its corresponding trajectory prototype c^i_{tra} and c^i_{tra}'s top k similar trajectory prototypes.

$$c^i_{env} = \frac{1}{K} \sum_{k \in \{T^i\}} c^k_{tra}, \tag{5.1}$$

where T_i denotes the index set of the top-K similar trajectory prototypes with c^i_{tra}. In this way, we can obtain the i-th environmental prototypes, but before that, we need to calculate the semantic similarity matrix w.

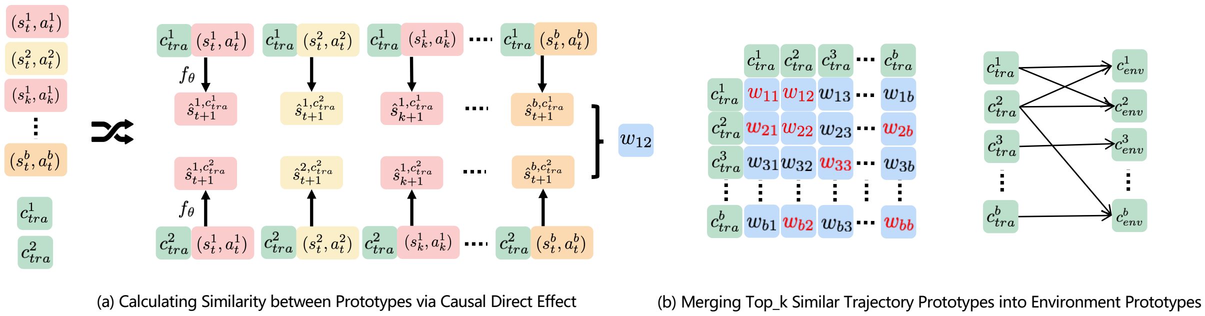

(a) Calculating Similarity between Prototypes via Causal Direct Effect
(b) Merging Top_k Similar Trajectory Prototypes into Environment Prototypes
f_θ
w_{12}

FIGURE 5.2. (a) The illustration of causal direct effect estimation between two trajectory prototypes, where we calculate the mean difference over a batch of predicted next states as the similarity w of the two prototypes. (b) The illustration constructed the similarity matrix between trajectory prototypes. We use the mean of top_k (denoted by the red color) similar trajectory prototypes as the corresponding environmental prototypes.

Normally, we can directly use the euclidean distance to discriminate the similarity between different trajectory prototypes. However, this ignores the semantic effect of trajectory prototypes on dynamics prediction. If two trajectories prototypes are from a single environment, their trajectory prototypes should share the same semantics, *i.e.*, and their effects on the dynamics function should be the same. Therefore, we consider take account the semantic effect on the dynamics prediction into similarity estimation. However, it is challenging to estimate the effects of the trajectory prototype on the dynamics function because Z is not the only factor that can influence the dynamics function. To remove the effects of other factors, *e.g.* states and actions, on the dynamics function, our method draws inspiration from the recently proposed RIA method [70] to calculate the direct causal effects (CDE) of trajectory prototypes. By controlling all factors that have effects on the dynamics function over a mini-batch, we can solely estimate average CDE between different trajectory prototypes as their semantic difference d. Concretely, we compute d between two trajectory prototypes using a mini-batch of S_t and A_t pairs (s_t^i, a_t^i) as Figure 5.2 (a) shows:

$$d_{ij} = \frac{1}{N}\sum_{k=1}^{N}|CDE_{c_{tra}^i, c_{tra}^j}(s_t^k, a_t^k)|, \tag{5.2}$$

where N is the batch size, i and j are the id of trajectory prototypes. Please refer to Appendix D1.6 for the details of CDE. With semantic difference d, we can convert it as the semantic difference w via $w = exp(\frac{-d}{\beta})$, where β is a factor that controls the sensitivity of w. With the calculated similarity w, we can construct environmental prototypes via equation 5.1.

Next, we will describe how to update the built trajectory and environmental prototypes to ensure that hierarchical prototypes are representative for each trajectory and environment, and how they help learn the context encoder.

5.3.3 Prototypical relational learning

As Figure 5.1 shows, we introduce a context encoder g parameterized by ϕ to estimate environmental-specific factor $\hat{z}_t^i$ from the past transition segments $\tau_{t-k:t-1} = \{(s_{t-k}, a_{t-k}), ..., (s_{t-1}, a_{t-1})\}$ following previous methods:

$$\hat{z}_t^i = g(\tau_{t-k:t-1}^i; \phi).$$

In order to learn the context encoder and encourage the estimated environmental-specific factor $\hat{z}_t^i$ to be semantically meaningful, we optimize g via the proposed prototypical relational loss to form a clear cluster for Zs from the same environments. Concretely, we introduce a relational head [194] as a learnable function h to derive the environmental-specific estimation $\hat{z}_t^i$ closely surrounded its associated cluster prototypes. To achieve this, we concatenate the $\hat{z}_t^i$ and its assigned prototypes, *e.g.*, c_{tra}^i as the positive pair, and the concatenation of other prototypes are negative pairs. Then we use the relational head h parameterized by φ to quantify the similarity score of $\hat{y}$. To increase the similarity score $\hat{y}$ of positive pairs and decrease those of negatives, we can regard it as a simple binary classification problem to distinguish positive and negative pairs. This can be regarded as maximizing the mutual information between Zs and its corresponding prototypes (Please refer to [256, 70] and Appendix A.3). However, it neglects the semantic correlation among different prototypes, and so it may excessively penalize some semantically relevant prototypes. To alleviate such over-penalization, we propose to penalize prototypes adaptively with the intervention similarity [70] through the following objective:

$$\begin{aligned}\mathcal{L}^{i-p-relation}_{\varphi,\phi} &= -\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N}\Big[\,[y^{i,j}+(1-y^{i,j})\cdot w^{i,j}]\cdot\log\ h([\hat{z}^{i},c^{j}];\varphi)\\ &\qquad +(1-y^{i,j})\cdot(1-w^{i,j})\cdot\log\,(1-h([\hat{z}^{i},c^{j}];\varphi))\Big],\end{aligned} \tag{5.3}$$

where w ranges from 0 to 1, and we use it as the similarity between different prototypes. In addition, the first term of equation 5.3 clusters z_t^i with prototypes c^j with the similarity weight $w^{i,j}$, and the second term push them away with weight $1 - w^{i,j}$. To maintain the hierarchical prototypes structure [142, 73], we simultaneously update the context encoder by optimizing the objective equation 5.3 between z with trajectory and environmental prototypes. Specifically, the calculation of similarity w_{env} between environmental prototypes and z is same with Section 5.3.2 as . In addition, we also optimize the relation loss among different Zs following [70, 142] because Z itself can be regarded as an instance prototype, and thus can retain the property of local smoothness and help bootstrap clustering.

In order to improve its generalization ability on different dynamics, we incorporate the estimated environment-specific $\hat{z}_t$ into the dynamics prediction model $\hat{f}$ and optimize the objective function following [135, 219, 94]:

$$\mathcal{L}^{pred}_{\theta,\phi} = -\frac{1}{N}\sum_{i=1}^{N}\log\ \hat{f}(s_{t+1}^{i}|s_t^i, a_t^i, g(\tau_{t-k:t-1}^{i};\phi);\theta), \tag{5.4}$$

where k is the length of transition segments, t is the current timestep, and N is the sample size. In addition, we also enable the built prototypes to optimize equation 5.4 to ensure that the learned prototypes are semantically meaningful. Overall, our method simultaneously optimize the prediction loss equation 5.4 and prototypical relational loss equation 5.3 with prototypes in different levels to learn context encoder g and semantic meaningful prototypes, which encourage the estimated environmental-specific $\hat{Z}$ can form clear clusters, and thus can learn a generalizable prediction function f.

5.3.4 Difference to RIA

This chapter refers the idea of RIA [70] to estimate semantic similarities between different prototypes. However, our method differs RIA from three aspects: 1) RIA estimates the semantic similarities between different instance estimation $\hat{z}$ while our method estimates the semantic similarities between different prototypes. Considering the number of prototypes are limited, the training procedure are faster and more stable than RIA. 2) Our method fully takes the advantage the hierarchy between trajectory and environments, and construct environmental prototype based on trajectory label information while RIA ignores it. Thus our method can achieve better performance than RIA. 3) RIA only pulls $\hat{z}$ and other estimations with similar semantics, but our prototypical relational learning further pulls the $\hat{z}$ and its corresponding trajectory c_{tra} and environmental prototypes c_{env}.

5.4 Experiment

In this section, we perform experiments to evaluate the effectiveness of our approach by answering the following questions: 1) Can our method encourages the learned Z to form a clear cluster? (Section C0.13); 2) Can the learned $\hat{Z}$ with the clear cluster reduce the dynamics prediction errors in model-based RL? (Supplementary Material D1.2); 3) Can the learned $\hat{Z}$ with the clear cluster promote the performance of model-based RL in environments with unseen dynamics? (Section 5.4.3); 4) Is our method sensitive to hyperparameters? (Section 5.4.4)

5.4.1 Environmental setup

Implementation details Our method includes three learnable functions and a set of learnable trajectory prototypes. The learnable functions are context encoder, relational head and prediction head, and they all are constructed with MLP and optimized by Adam [116] with 1e-3 learning rate. During the training procedure, the trajectory segments are randomly sampled from the same trajectory to break the temporal correlations of the training data, which was also adopted by [219, 70]. Specifically, we combine $k = 3$ similar trajectory embedding into environmental embedding, and the length of the transition segments is 10, and the hyper-parameters are the same for all experiments, and details can be found in supplementary material D1.1.

Tasks Following the previous methods [135, 219], we perform experiments on the classic control algorithm (Pendulum) from OpenAI gym [24] and simulated robotics control tasks (HalfCheetah, Swimmer, Ant, Hopper, Slim-Humanoid) from Mujoco physical engine [254].

Dynamics settings To construct different dynamics of environments, we change the environmental parameters (*e.g.,* length and mass of Pendulum) and predefine them in the training and test environmental parameters lists following previous methods [320, 190, 135, 219, 70]. Specifically, for the convthe training environmental parameters lists for all tasks are $\{0.75, 0.8, 0.85, 0.90, 0.95, 1, 1.05, 1.1, 1.15, 1.2, 1.25\}$, and test environmental parameters lists are $\{0.2, 0.4, 0.5, 0.7, 1.3, 1.5, 1.6, 1.8\}$. We can see that the parameters in test list are out of range of the parameters in the training set. At the training time, we randomly sample the parameters from the training parameter list to train our context encoder and dynamics prediction model. Then we test our model on the environments with unseen dynamics sampled from the test parameter list. All details are given in supplementary material D1.1.

Planning Following [135, 219], we use the model predictive model (MPC) [167] to select actions based on learned dynamics prediction model, and assume that reward functions are known. In addition, we use the cross-entropy method (CEM) [39] to find the best action sequences.

Baselines In this Chapter, we compare our approach with the following state-of-the-art model-based RL methods on dynamics generalization:

- Context-aware dynamics model (CaDM) [135]: This method design several auxiliary loss, including backward and future states prediction to learn the context from transition segments.

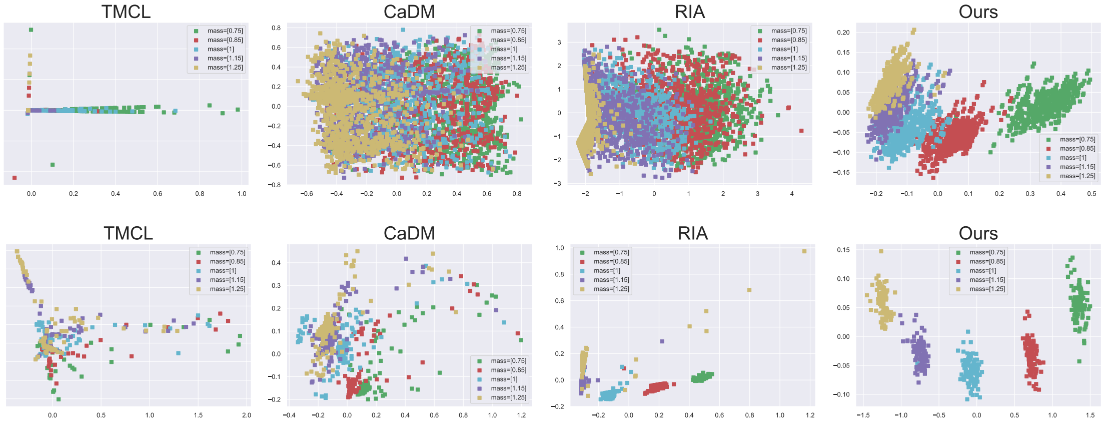


FIGURE 5.3. The PCA visualization of environmental-specific factors estimated by TMCL [219], CaDM [135], RIA [70] and ours on the Half-cheetah (upper part) and Pendulum (lower part) task.

- Trajectory-wise Multiple Choice Learning (TMCL) [219]: TMCL introduces multi-choice learning to adapt to different environments. For a fair comparison, we use the no adaptation version of this method.
- Relation Intervention Approach (RIA) [70]: This method proposes to use relational intervention loss to cluster Zs from the same environments.

It has been clearly evidenced that Probabilistic ensemble dynamics model (PETS) [127] and Meta learning based model-based RL methods, *e.g.* Recurrent model ReBAL and hidden-parameter model GrBAL [186, 187], perform worse than CaDM [135],TMCL [219] and RIA [70], so we do not consider them as baselines in this Chapter.

5.4.2 Cluster visualization and analysis

TABLE 5.1. The quantitative evaluation results of estimated environmental-specific factors.

	ARI				AMI				V-means			
	TMCL	CaDM	RIA	Ours	TMCL	CaDM	RIA	Ours	TMCL	CaDM	RIA	Ours
HalfCheetah	0.006	0.128	0.212	**0.570**	0.058	0.175	0.333	**0.681**	0.06	0.176	0.314	**0.680**
Pendulum	0.060	0.471	0.754	**0.971**	0.054	0.529	0.838	**0.967**	0.051	0.531	724	**0.975**

We perform PCA visualization of estimated $\hat{Z}$s from baselines and our method as Figure 5.3 to evaluate the cluster performance of estimated $\hat{Z}$s. We can see that our method can achieve better cluster performance qualitatively. Specifically, most $\hat{Z}$s estimated by RIA [70] have good cluster performance in general, but the outliers decrease the cluster performance. By contrast, we can see that there are fewer outliers in our method than them in RIA because the built prototypes and the proposed prototypical relational loss can enforce constraints into estimated $\hat{Z}$s. More qualitatively cluster comparisons can be found in Supplementary Material D1.8.

We also quantitatively evaluate the cluster performance of $\hat{Z}$s estimated by baselines and our method. Here we firstly perform k-means [168] on the estimated $\hat{Z}$s, and then use the ground-truth environmental label to calculate the cluster performance. Here we use the popular mutual information-based metric AMI [259], random-index metric ARI[89] and V-means [209] as the evaluation metrics. The results are shown in Table D.2, we can see that $\hat{Z}$s estimated by our method achieves the highest cluster performance. More quantitatively cluster comparisons can be found in Supplementary Material D1.8.

5.4.3 Performance comparisons

Then, we evaluate the generalization of model-based RL agents trained by our methods and baselines on test environments with unseen dynamics. Following the setting of [219], we perform experiments across five runs, and show the test returns on the test environments in Figure D.1. Note that the results are slightly different from the results in RIA and TMCL paper since we change the parameter lists that change the environmental dynamics. Specifically, we change the parameter lists of all environments to the same for the convenience of performing environments.

As Figure D.1 shows, we can see that our method can achieve significantly better performance than baselines in Ant, Halfcheetah, and Pendulum. Specifically, we can see that our method outperforms the second-best method RIA by 20% in Ant and Halfcheetah environments, which indicates that the changing parameter can largely

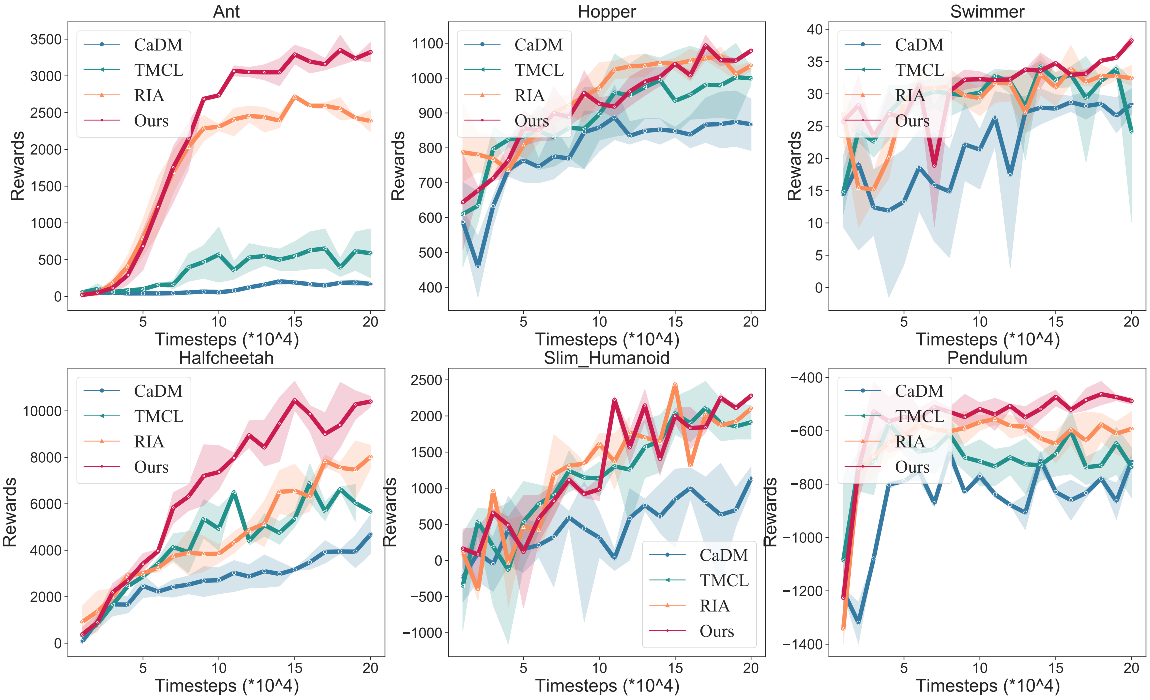


FIGURE 5.4. The average returns of model-based RL agents on unseen test environments. The results show the mean and standard deviation of returns averaged over five runs. Specifically, we use the no adaptation version of TMCL for a fair comparison. The performance comparisons on dynamics prediction errors are given at Appendix D1.4.

change their dynamics. In addition, we can see that our method achieves only slightly better performance than baselines in Hopper, Swimmer, and Slim_Humanoid problems. For Hopper and Slim_Humanoid environment, we observe that both RIA and our method can achieve comparable results in all test environments, which indicates that the change of dynamics for Hopper is easy to model and solve. For the Swimmer environment, we observe that TMCL [219] sometimes may have a significant performance decline at the final training iteration. This may be because that TMCL may fail to learn the modalities of dynamics function in the no adaptation version. Also, our method still achieves better performance than RIA at the Swimmer task.

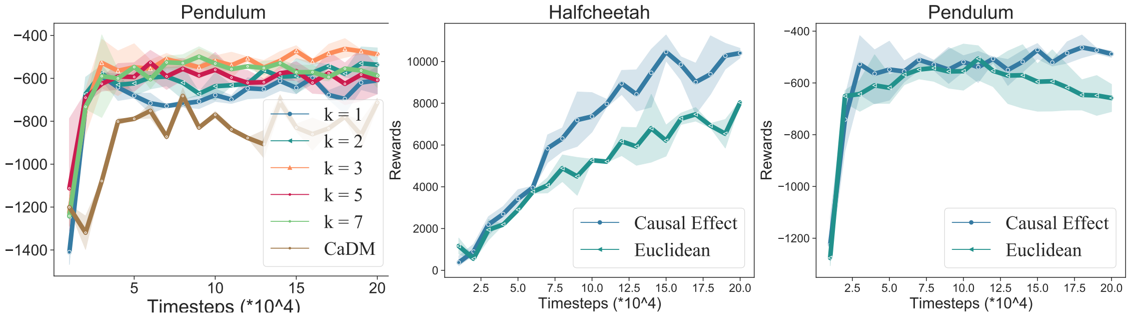


FIGURE 5.5. Left Image: The sensitivity analysis about how many trajectory prototypes should be combined into environmental prototypes. Right Two Images: The similarity metrics used in combining trajectory prototypes into environmental prototypes.

5.4.4 Ablation study

In this section, we first perform a sensitive analysis of how many trajectory prototypes should be combined into environmental prototypes. The experiments are conducted at the Pendulum task, and the results are shown as the left image of Figure 5.5, we can see that no matter what k it is, our method consistently outperforms the baseline CaDM [135], which indicates that our method is robust to the selection of k value. Specifically, $k = 1$ means that there are no hierarchical prototypes because one trajectory prototype can decide one environmental prototype, and thus environmental prototypes are the same as trajectory prototypes. We can see that all experimental results with $k > 1$ are better than the experimental result with $k = 1$, which shows the effectiveness of our proposed hierarchical prototypes method and the necessity of the built environmental prototypes. The results of $k = 1$ achieve the best performance on the Pendulum task, so we use it as the default parameter in all experiments.

We also perform an ablation study about the similarity metric used to calculate the similarity among trajectory prototypes. For most cluster methods, *e.g.* k-means [168], they usually calculate the similarity among entities using the Euclidean distance, while our method uses the direct causal effect as the similarity metric. To evaluate the effectiveness of the similarity metrics based on direct causal effect [197], we perform experiments on the Halfcheetah and Pendulum tasks, and we can see that using the causal effect to calculate the similarities among trajectory prototypes can achieve better performance than using Euclidean distance on both tasks.

5.5 Limitation

Our paper only considers the unsupervised dynamics generalization in model-based reinforcement learning, but model-free RL also suffers from this problem, and we will apply our method to model-free RL in future work. In addition, there are many other generalization problems in reinforcement learning area, *e.g.* observation generalization [263, 118, 59] and action generalization [92], and it would be interesting to extend our method into other generalization settings and train generalizable agents.

5.6 Conclusion

In this paper, we focus on the unsupervised dynamics generalization problem in model-based reinforcement learning, and propose a hierarchical prototypical method to construct environmental prototypes in an unsupervised manner. With the learned environmental prototypes, we further propose a prototypical relational loss to learn a context encoder to estimate environmental-specific factors from past transition segments, which enables the dynamics prediction function in model-based reinforcement learning to generalize well on environments with unseen dynamics. The experiments demonstrate that our method can form clearer and tighter clusters for $\hat{Z}$s from the same environment and improve the performance of model-based agents in new environments with unseen dynamics.

CHAPTER 6

From Images to Textual Prompts: Zero-shot VQA with Frozen Large Language Models

Large language models (LLMs) have demonstrated excellent zero-shot generalization to new language tasks. However, effective utilization of LLMs for zero-shot visual question-answering (VQA) remains challenging, primarily due to the modality disconnection and task disconnection between LLM and VQA task. End-to-end training on vision and language data may bridge the disconnections, but is inflexible and computationally expensive. To address this issue, we propose *Img2Prompt*, a plug-and-play module that provides the prompts that can bridge the aforementioned modality and task disconnections, so that LLMs can perform zero-shot VQA tasks without end-to-end training. In order to provide such prompts, we further employ LLM-agnostic models to provide prompts that can describe image content and self-constructed question-answer pairs, which can effectively guide LLM to perform zero-shot VQA tasks. Img2Prompt offers the following benefits: 1) It can flexibly work with various LLMs to perform VQA. 2) Without the needing of end-to-end training, it significantly reduces the cost of deploying LLM for zero-shot VQA tasks. 3) It achieves comparable or better performance than methods relying on end-to-end training. For example, we outperform Flamingo [3] by 5.6% on VQAv2. On the challenging A-OKVQA dataset, our method even outperforms few-shot methods by as much as 20%.

6.1 Introduction

Visual question answering (VQA) [5] is a prominent vision-language task that finds a broad range of real-world applications, such as assisting blind individuals in understanding their environments. A diverse set of VQA datasets have been proposed, some focusing on image recognition [65, 5] and others on logical reasoning [172]. However, human annotations are expensive to obtain and may introduce a variety of human biases [30, 11, 301], making the VQA system brittle towards new answer styles and question types[1, 101]. This has led researchers to zero-shot VQA methods [30, 11, 101] that do not require ground-truth question-answer annotations, thereby facilitating more generalizable VQA systems.

Recently, large language models (LLMs) (e.g., [25, 316]) have demonstrated excellent capabilities to perform tasks with zero in-domain data, conduct logical reasoning, and apply commonsense knowledge in NLP tasks [119, 274, 273]. As a result, recent approaches [3, 293, 257] have resorted to leverage LLMs in zero-shot VQA.

However, applying LLMs to VQA tasks is less than straightforward, due to (1) the modality disconnect between vision and language and (2) the task disconnect between language modeling and question answering. A common technique is to finetune a vision encoder jointly with the LLM [257, 3, 98] to align the vision and language representation spaces, but this can incur prohibitive computational and data cost. For example, Flamingo [3] finetunes on billions of image-text pairs with thousands of TPUs. Further, the finetuning specializes and introduces strong interdependence between the vision encoder and the LLM. If we need to upgrade the LLM as new versions emerge, the entire model needs to undergo expensive re-training.

In contrast to the end-to-end integration of LLM into a VQA system, this Chapter proposes a modular VQA system built on top of frozen off-the-shelf LLMs. This brings two benefits. First, it can reduce the deployment cost and simplify the deployment. Second, upgrading the LLM is straightforward. However, it is challenging to bridge the modality disconnect and task disconnect without end-to-end training. PICa [293] converts images into captions, and provides exemplar QA pairs from training data as prompt to the LLM. However, doing so assumes the existence of annotated training data and the performance is sensitive to the selection of few-shot exemplars.

We propose *Img2Prompt*, a plug-and-play module that enables off-the-shelf LLMs to perform zero-shot VQA. The central insight of Img2Prompt is that we can utilize a vision-language model (*e.g.* BLIP [141]) and a question-generation model to translate the image content into synthetic question-answer (QA) pairs, which are fed to the LLM as part of the prompt. These exemplar QA pairs tackle the modality disconnect by describing the image content verbally, and tackle the task disconnect by demonstrating the QA task to the LLM. Notably, the exemplar QA pairs are constructed entirely based on the test image and question, obviating the need for similar few-shot examples as required by PICa [293], which are not always available in practical zero-shot scenarios. When applied to the open-source OPT language models [316], Img2LLM achieves comparable or superior zero-shot VQA performance to methods that perform costly end-to-end training.

With this Chapter, we make the following contributions.

- We propose Img2LLM, a plug-and-play module that converts an image into synthetic question-answer pairs based solely on the current image of the question. Img2LLM bridges the modality disconnect between language and vision as well as the task disconnect between language modeling and visual question-answering.
- Img2LLM enables off-the-shelf LLMs to perform zero-shot VQA without costly end-to-end training or specialized textual QA networks [177], thereby allowing low-cost and flexible model deployment and painless LLM upgrades (Table 6.3).

- Our experimental results show that the OPT models equipped with Img2LLM achieve zero-shot VQA performance that is competitive or superior to the end-to-end trained models. For example, we outperform Flamingo [3] by 5.6% on VQAv2. We even outperform many few-shot VQA methods.

6.2 Related Work

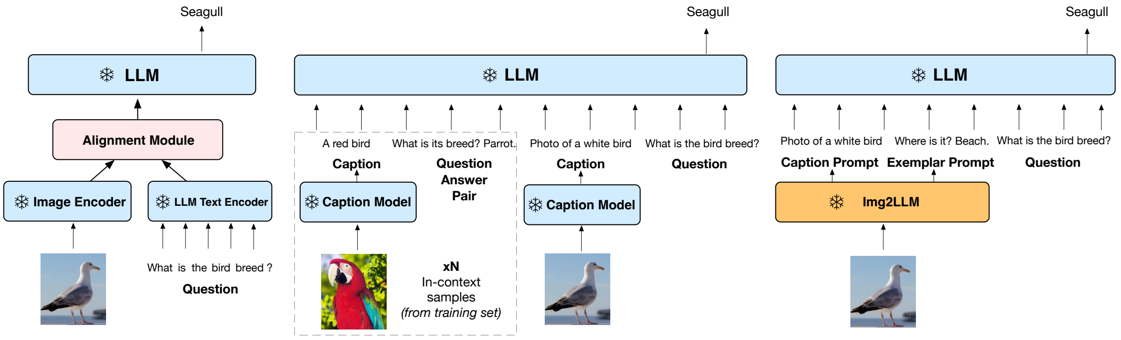
6.2.1 Recent Advances in VQA Methods

As a multi-modal evaluation benchmark, Visual Question Answering (VQA) that requires the model to answer a natural language question according to the image, has been the focus of active research [294, 4, 5, 216, 2]. The past few years witnessed rapid performance advances with large-scale image-text pretraining [96, 302, 161, 140, 143, 313, 272, 229, 141, 98, 38] followed byfine-tuning on VQA datasets. To tackle knowledge-based VQA [216, 172], recent works [69, 145, 283, 165, 164, 171, 56, 139] incorporate external knowledge, such as ConceptNet [233] or Wikipedia, but experimental results in [216] show that these methods still struggle to answer questions requiring complex reasoning.

6.2.2 LLM for Zero/Few-Shot VQA Tasks

Large language models (LLMs) [26, 316, 34] trained on web-scale corpus are powerful in natural language understanding and reasoning [319, 25]. To infer on task data, LLMs typically generate target tokens autoregressively. In specific, given prompt C and task input x, an LLM generates target tokens $Y = \{y_i\}_{i=1}^{n}$, with y_i = $\arg\max p_\theta(y_i|y_{<i}, C, x)$ and θ the model parameters. Prior VQA methods using LLMs mainly fall into two categories: multi-modal pretraining and language-mediated VQA.

Multi-modal pretraining. These approaches align vision and language embeddings by training additional alignment modules, as shown in Figure 6.1(a). Considering that LLMs are too large to finetune efficiently, [257] opt to fine-tune only the visual encoder while Flamingo [3] trains extra cross-attention layers to model cross-modality interactions. However, this paradigm suffers from two drawbacks: 1) Highly computationally inefficient. Jointly aligning vision backbones and LLMs requires large compute resources. For example, training Flamingo requires 1536 TPUv4 over two weeks. Hence, it becomes prohibitively expensive to switch to a different LLM. 2) Catastrophic forgetting. The



(a) Multi-modal pretraining (b) In-context few-shot learning (c) Img2LLM (Ours)

FIGURE 6.1. The illustrative comparison of three types of methods that enable LLM to perform VQA tasks, where blue block denotes that the inner parameters are frozen while pink block indicates the inner parameters are trainable.

alignment step may be detrimental to LLMs' reasoning ability, if the LLMs are jointly trained with the visual model [3].

Language-mediated VQA. Instead of vectorized representations, this VQA paradigm directly resorts to natural language as the intermediate representation of the image and no longer requires expensive pretraining. As depicted by Figure 6.1(b), it first converts the current image to language descriptions and feeds the descriptions, possibly accompanied by in-context exemplars, to a frozen LLM. In a few-shot setting, PICa [293] generates captions for the image and selects training data samples as in-context exemplars, but its performance degrades substantially when the exemplars are omitted. As a concurrent zero-shot approach, [177] generates question-relevant captions. Due to the zero-shot requirement, it is unable to provide in-context exemplars and does not reap the benefits of in-context learning. As a result, it has to rely on a QA-specific LLM, UnifiedQAv2 [111], to achieve high performance.

6.3 Method

Difficulties in utilizing LLMs effectively in zero-shot VQA stem mainly from two obstacles: (i) *The modality disconnection*: LLMs do not natively process images and encoding visual information into a format that LLMs can process can be a challenge. (ii)

The task disconnection: LLMs are usually pretrained using generative [25] or denoising objectives [42] on language modeling tasks. As the LLMs are unaware of the tasks of question answering or VQA, they often fail to fully utilize contextual information in generating the answers.

In language-mediated VQA [293, 177], the modality disconnection is addressed by converting the image to intermediate language descriptions instead of dense vectors (§6.2.2). The task disconnection must be addressed using either few-shot in-context exemplars [293] or an LLM directly finetuned on textual QA [177]. It is not clear how to tackle the task disconnection on generic LLMs under zero-shot settings.

We propose a new zero-shot technique to address the task disconnection on generic LLMs, Img2Prompt (Figure 6.1c), which generates image-relevant exemplar prompts for the LLM. Given a question Q and an image, our key insight is that we can generate synthetic question-answer pairs as in-context exemplars from the *current* image. The exemplars not only demonstrate the QA task but also communicate the content of the image to the LLM for answering the question Q, thereby hitting two birds with one stone. Img2Prompt is LLM-agnostic; it unlocks the knowledge and the reasoning capacity of off-the-shelf LLMs, offering a powerful yet flexible solution for zero-shot VQA.

6.3.1 Answer Extraction

In order to incorporate the image content into the exemplars for in-context learning, from the current VQA image, we first seek words that could serve as answers to synthetic questions. We generate a number of captions using an off-the-shelf question-relevant caption generation module (§6.3.3). Following recent papers [30, 134], we extract noun phrases (including named entities), verb phrases, adjective phrases, numbers, and boolean-typed words like "yes" and "no" as potential answers[1]. We show some extracted answer candidates in Figure 6.2 and Appendix A.3.

[1] We use the spaCy parser at https://spacy.io/

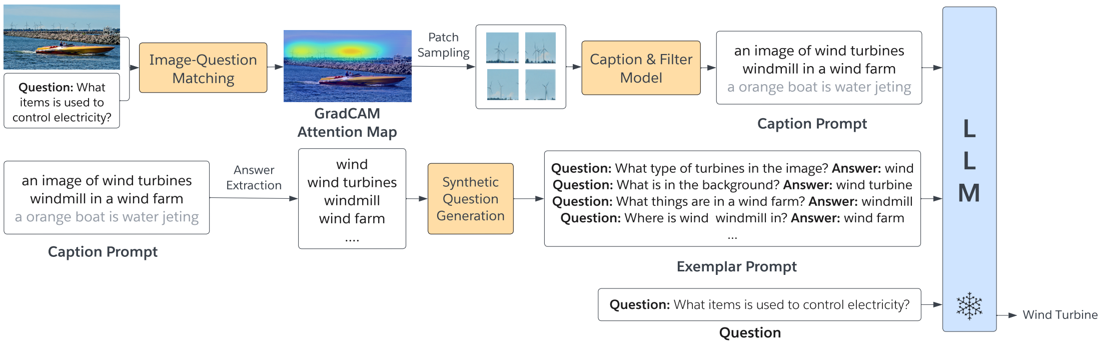

Question: What items is used to control electricity?
Image-Question Matching
GradCAM Attention Map
Patch Sampling
Caption & Filter Model
an image of wind turbines
windmill in a wind farm
a orange boat is water jeting
Caption Prompt
an image of wind turbines
windmill in a wind farm
a orange boat is water jeting
Caption Prompt
Answer Extraction
wind
wind turbines
windmill
wind farm
....
Synthetic Question Generation
Question: What type of turbines in the image? Answer: wind
Question: What is in the background? Answer: wind turbine
Question: What things are in a wind farm? Answer: windmill
Question: Where is wind windmill in? Answer: wind farm
...
Exemplar Prompt
Question: What items is used to control electricity?
Question
L L M
Wind Turbine

FIGURE 6.2. The overall pipeline of Img2Prompt, including Caption Prompt and Exemplar Prompt generation.

TABLE 6.1. Results from mixing captions and exemplar prompts on 30B OPT [316].

Prompt Template	Caption Prompt	Exemplar Prompt	VQAv2 val	OK-VQA
Instruction	✗	✗	18.1	3.3
Instruction + Captions	✓	✗	46.1	23.5
Instruction + Question-Answer Pairs	✗	✓	57.9	41.1
Instruction + Captions + Question-Answer Pairs	✓	✓	59.5	41.8

6.3.2 Question Generation

With the extracted answer candidate set $\{\hat{a}_j\}_{j=1}^{U}$, we can directly use any question generation network [102, 160, 286, 113, 2] to generate specific questions for each answer candidate. In this Chapter, we experiment with both template-based and neural question-generation methods. Note that to avoid violating the zero-shot requirements, our method is purely textual-based without access to any VQA data.

Template-based Question Generation. Using an off-the-shelf parser, we obtain the part-of-speech for each answer, and design specific question templates for each POS type. For example, for answers that are nouns, we use the question "What object is in this image?" For verb answers, we use the question "What action is being taken in this image?" Due to space constraints, we put the complete list of templates in Appendix A.5.

Neural Question Generation. Inspired by [30], we train a neural question generation model on textual QA datasets. Specifically, we finetune a pretrained T5-large model [203] to generate questions from answers. The input to the model contains the prompt "Answer:

`[answer]`. Context: `[context]`", where `[answer]` denotes the answer text and `[context]` denotes the context text from textual QA datasets. During inference, we replace `[answer]` with an extracted answer candidate and `[context]` with the generated caption from which the answer was extracted. The model is finetuned on five textual QA datasets including SQuAD2.0 [205], MultiRC [110], BookQA [178], CommonsenseQA [247] and Social IQA[213].

With the above question generation methods, we acquire a set of synthetic question-answer pairs $\{\hat{q}_j, \hat{a}_j\}_{j=1}^{U}$. We use these question-answer pairs as exemplars of LLM in-context learning [25], which guides the LLM to perform QA task given the image content and bridges the task disconnect between language modelling and VQA.

As a sneak preview, we show effects of exemplar QA pairs in Table 6.1. The details of the instructions are explained in §6.3.4. We observe that exemplar QA prompts perform considerably better than caption prompts (detailed in §6.3.3) only, demonstrating their efficacy in bridging the task disconnection between LLM pre-training and VQA tasks. Moreover, since the exemplar prompts already describe much content of the image, which helps to bridge the modality disconnection, adding captions on top does not provide much new information and brings only limited performance gains.

6.3.3 Question-relevant Caption Prompt

In addition to the synthetic exemplar QA pairs, we also supply question-relevant image captions to the LLM. We observe that the question may ask about specific objects or regions in the image [282] but generic captions generated by existing networks may not contain relevant information. In Figure 6.2, the question *"What items are spinning in the background which can be used to control electricity?"* is relevant only to the wind turbines. However, captions generated from the whole image are likely to focus on the salient orange boat, leaving LLM with no information to answer the question. To address this issue, we generate captions about the question-relevant portion of the image and include them in the prompt to the LLM.

To achieve this, we first determine the regions of the image that are relevant to the question, by using the Image-grounded Text Encoder (ITE) in BLIP [141], as which assigns a similarity score sim(v, q) to any pair of image v and textual question q. With ITE, we use GradCAM [218], a feature-attribution interpretability technique, to generate a coarse localisation map highlighting matching image regions given a question [141]. Briefly, GradCam qualifies the cross-attention scores from the Transformer network by the gradient of ITE simlarity function sim(v, q) with respect to the cross-attention scores. As this technique was proposed in [177], we leave the details to Appendix A.1.

Having obtained the patch relevance r, we sample a subset of image patches with probability proportional to patch relevance r. After that, we generate captions from the sampled image patches using top-k sampling [47]. To generate semantically meaningful captions, a short prompt, “a picture of," is also fed into the text decoder. We repeat this M times for each image to generate M diverse captions, and keep only captions that are not exact substrings of others.

However, due to the non-deterministic nature of top-k sampling, the caption model may generate noisy captions that have a negative impact on performance. To remove noisy captions, we use ITE to calculate the similarity score between the generated caption and sampled question-relevant image patches, and filter captions with less than 0.5 matching scores. Overall, this process yields synthetic captions that are question-relevant, diverse, and clean, providing a bridge between visual and language information.

6.3.4 Prompt Design

With synthetic question-relevant captions and question-answer pairs, we construct complete prompts for LLM by concantenating the instruction, captions, and QA exemplars. The instruction text is “Please reason the answers of question according to the contexts." The caption prompt is formatted as “Contexts: `[all captions]`". Individual QA exemplars are formatted as “Question: `[question]` Answer: `[answer]`” and concatenated. We position the current question as the last portion of the prompt, formatted as “Question: `[question]`. Answer: ”. Finally, to get the answer, we perform greedy decoding on the LLM and remove meaningless tokens as in Flamingo.

Furthermore, as the input to LLMs has maximum lengths, *e.g.* 2048 in OPT and GPT3, it is necessary to select a subset of question-relevant captions and question-answer pairs to construct the prompt. To select the most informative prompt, we first count the frequency of the synthetic answer candidates in 100 generated captions. We then select 30 answer candidates with highest frequencies and generate one question for each. Also, we include 30 answers with the lowest frequency and one caption containing each answer. See §6.4.5 for analysis of caption selection strategies.

6.4 Experiment

In this section, we first validate the efficacy of Img2Prompt by comparing it with other zero-shot and few-shot VQA methods. Then, we perform ablation studies on important design choices, such as prompt patterns and caption selection strategies, to understand their effect. We also show qualitative examples and include discussion on observed failure cases.

6.4.1 Environment Setup

Datasets. We validate our method on VQAv2 [65], OK-VQA [172] and A-OKVQA [216] datasets, which contain questions requiring perception, reasoning and commonsense to answer. Specifically, VQAv2 [65] contains 214,354 questions in the validation set and 107,394 in the test-dev dataset. OK-VQA [172] and A-OK-VQA [216] emphasize on commonsense reasoning, among which OK-VQA contains 5,046 test questions and A-OKVQA [216] contains 1,100 validation questions and 6,700 test questions.

Implementation details. To obtain question-relevant caption prompt, we use BLIP [141] to generate captions and perform image-question matching. To localize the image regions relevant to the question, we generate GradCam from the cross-attention layer of

BLIP image-grounded text encoder. We then sample $K' = 20$ image patches based on GradCam, and use them to obtain 100 question-relevant captions. For the LLMs, our main result uses the open-source OPT model with multiple different sizes. Our ablation study also experiments with various other LLMs to show the generalization ability of our method. We use LLMs to generate answers auto-regressively, without access to either answer list or training samples, thereby facilitating zero-shot VQA. We follow official evaluation protocols and report VQA scores on each dataset.

Competing methods. We compare with prior VQA methods, which rougly fall into three categories: (i) *Zero-shot methods with frozen LLMs*, such as PICa [293]. Our method also belongs to this category, yet unlike PICa, Img2Prompt requires no training samples to compose the prompts. (ii) *Zero-shot methods with extra multi-modal pre-training*, such as Flamingo [3], Frozen [257], VL-T5 [32], FewVLM [98] and VLKD [38]. These methods require large-scale vision-language datasets and are costly to update. We also include results from VQ^{2}A [30] and WeaQA [11] in this category, with *caveats* that they assume access to answer candidates which may not be available in practice. Therefore, their results should be interpreted with caution. (iii) For reference purposes, we also include available results from *few-shot methods*. These include few-shot results of PICa [293], FewVLM [98] and ClipCap [183].

6.4.2 Main Results

Main quantitative results are shown in Table 6.2. We summarize our findings as follows.

State-of-the-art results on zero-shot evaluation with plug-in frozen LLMs. Img2Prompt surpasses PICa, the best prior zero-shot model with frozen LLMs, by a significant margin (45.6 *versus* 17.7 on OK-VQA), thereby establishing a new state-of-the-art. In addition, we remark that despite PICa uses frozen LLMs, it requires training samples to build prompts. In contrast, our method generates question-answers with no access to VQA samples, thus fully fulfilling the zero-shot requirements.

TABLE 6.2. Performance on VQAv2, OK-VQA, and A-OKVQA. A few methods do not strictly satisfy the zero/few-shot requirements: methods without end-to-end training but assumes access to training samples are labeled with †; methods that answer from a predefined list of candidates are in grey. Further, ✗ annotates methods requiring no end-to-end training, which is desirable, and ✓ otherwise.

Methods	End-to-End	Shot	VQAv2		OK-VQA	A-OKVQA	
	Training?	Number	val	test	test	val	test
Zero-Shot Evaluation with Frozen Large Language Model							
PICa$_{\text{175B}}$†	✗	0	-	-	17.7	-	-
Img2LLM$_{\text{6.7B}}$	✗	0	57.6	57.0	38.2	33.3	32.2
Img2LLM$_{\text{13B}}$	✗	0	57.1	57.3	39.9	33.3	33.0
Img2LLM$_{\text{30B}}$	✗	0	59.5	60.4	41.8	36.9	36.0
Img2LLM$_{\text{66B}}$	✗	0	59.9	60.3	43.2	38.7	38.2
Img2LLM$_{\text{175B}}$	✗	0	**60.6**	**61.9**	45.6	**42.9**	**40.7**
Zero-Shot Evaluation with Extra End-to-End Training							
VL-T5$_{\text{no-vqa}}$	✓	0	13.5	-	5.8	-	-
FewVLM$_{\text{base}}$	✓	0	43.4	-	11.6	-	-
FewVLM$_{\text{large}}$	✓	0	47.7	-	16.5	-	-
VLKD $_{\text{ViT-B/16}}$	✓	0	38.6	39.7	10.5	-	-
VLKD $_{\text{ViT-L/14}}$	✓	0	42.6	44.5	13.3	-	-
Frozen$_{\text{7B}}$	✓	0	29.5	-	5.9	-	
Flamingo$_{\text{3B}}$	✓	0	-	49.2	41.2	-	-
Flamingo$_{\text{9B}}$	✓	0	-	51.8	44.7	-	-
Flamingo$_{\text{80B}}$	✓	0	-	56.3	**50.6**	-	-
Zero-shot Evaluation with Access to Answer Candidates							
WeaQA ZSL	✓	0	46.8	-	-	-	-
VQ^{2}A	✓	0	61.1	-	19.8	-	-
Few-Shot Evaluation							
ClipCap→Cap→GPT$_{\text{175B}}$	✗	10	-	-	-	16.6	15.8
ClipCap→Rel→GPT$_{\text{175B}}$	✗	10	-	-	-	18.1	15.8
FewVLM$_{\text{base}}$	✓	16	48.2	-	15.0	-	
FewVLM$_{\text{large}}$	✓	16	51.1	-	23.1	-	-
PICa$_{\text{175B}}$†	✗	1	-	-	36.4	-	-
PICa$_{\text{175B}}$†	✗	4	-	-	43.3	-	-
PICa$_{\text{175B}}$†	✗	16	54.3	-	46.5	-	-
PICa$_{\text{175B}}$-Ensemble	✗	80	56.1	-	48.0	-	-

Scaling effect of LLMs and their emergent capabilities on VQA. When increasing the number of parameters of LLMs from 6.7B to 175B, we see a 3-10 points improvement in VQA across datasets. This shows that stronger language modelling capabilities help better comprehend the question, thus giving more accurate answers. Such a trend is more clear and consistent on OK-VQA and A-OKVQA, whose questions demand commonsense reasoning and external knowledge that LLMs excel at providing. This corroborates our belief that LLMs are beneficial to VQA.

Another intriguing phenomenon we observe is that the effect of scaling LLMs becomes obvious only when the model size becomes sufficiently large, for example, when using 30B or larger models, while not entirely predictable on smaller ones (6.7B and 13B). This echoes with the recent finding on the emergent abilities when using LLMs off-the-shelf [275] for language tasks, while confirming the same trend for the first time when using frozen LLMs for vision(-language) tasks.

Competitive performance with end-to-end pretraining and few-shot models. Img2Prompt obtains superior performance to most models with end-to-end pretraining, as well as those evaluated in few-shot setups. For example, on VQAv2 our method surpasses Flamingo$_{80B}$, which cost over 500K TPU hours and billion-scale datasets to train, by a margin of 5.6 points. On A-OKVQA, Img2Prompt more than doubles the best reported results so far, from ClipClap. The only a few exceptions are on OK-VQA, where our method obtains better results than Flamingo$_{9B}$, yet is not able to stay on par with Flamingo$_{80B}$. Considering that Img2Prompt is flexible to adapt to updated and stronger LLMs with zero extra training cost, we consider it a more approachable solution to practical adoption of VQA systems, than those trained end-to-end. We also include comparisons with supervised models in Appendix A.4. Img2Prompt achieves better performance than most supervised models, despite the fact that it uses zero training data and is evaluated in a zero-shot setup. These results once again validates its effectiveness.

TABLE 6.3. Zero-shot VQA performance with different LLMs.

Methods	VQAv2 val	OK-VQA
PICa $_{\text{GPT-3 175B}}$	-	17.7
Frozen$_{7B}$	29.5	5.9
Ours $_{\text{GPT-Neo 2.7B}}$	50.1	31.5
Ours $_{\text{BLOOM 7.1B}}$	52.4	32.4
Ours $_{\text{GPT-J 6B}}$	56.4	37.4
Ours $_{\text{OPT 6.7B}}$	57.6	38.2
Ours $_{\text{OPT 175B}}$	60.6	45.6

6.4.3 Experimental Results of Different LLMs

In Table 6.3, we evaluate the performance of Img2LLM on various open-sourced LLMs other than OPT, including GPT-J [262], GPT-Neo [20] and BLOOM [214]. The experimental results show that Img2LLM enables various LLMs to perform zero-shot VQA tasks, and that all of them achieve superior performance to zero-shot PICa [293] and Frozen [257]. This is a strong evidence for showing our method's generalization ability with different LLMs.

6.4.4 Analysis on Question Generation Methods

Table 6.4 shows the performance of different question selection strategies described in Section 6.3.2. We compare three question generation techniques, include image-*agnostic*, which uses questions sampled from other images; *template*-based, which uses template questions, and *neural*-based, which uses neural generated questions. Further, we compare two synthetic QA selection strategies. The *random* strategy, which selects QA pairs for prompt randomly; the *max freq.* approach, which selects answer candidates that are most frequent in the captions, and also retrieve the associated synthetic questions to build the prompt.

Among the three question generation techniques, *Agnostic* perform the worst whereas *Neural* performs the best. We attribute the differences to the quality of QA pairs. *Agnostic* QA pairs contain information irrelevant to the current image and may mislead the LLM. *Template* questions feature little linguistic variation and hence cannot demonstrate different QA strategies. *Neural* has the most relevant information and the most linguistic diversity. QA pair with maximum answer frequency outperform random questions. We hypothesize that the most frequent answers describe the most salient or important aspects of the image, thereby providing more information than random questions.

In addition, we evaluate visual information quality encoded in the exemplar prompts using the answer hit rate and the answer noise rate. Answer hit rate (AHR) is defined as the proportion of QA pairs containing the ground-truth answer. Answer noise rate (ANR)

TABLE 6.4. Effect of question selection strategies.

		OK-VQA	VQAv2
$\text{PICa}_{175\text{B}}$		17.7	-
Agnostic	Random	35.9	52.9
Template	Random	40.2	53.0
	Max Freq.	41.5	55.8
Neural	Random	40.5	57.0
	Max Freq.	**41.8**	**59.5**

TABLE 6.5. Ablations on prompts designs.

Methods	OK-VQA	VQAv2 val
CQA-CQA-CQA	37.8	52.1
CCC-QAQAQA	41.8	59.5

TABLE 6.6. Ablation on caption selection methods.

Caption Selection	Random	Max Frequency	Min Frequency
OK-VQA Acc	41.3	41.1	**41.8**

is defined as the ratio of ground-truth answers to the total number tokens in the exemplar prompts. Table 6.7 indicates that exemplar prompts generated from question-relevant captions have a higher AHR, hence enhancing the VQA performance. In addition, the caption filter procedure can remove some noisy captions, allowing it to achieve a higher ANR than its competitors. The experimental results demonstrate that improving both the AHR and the ANR can improve the quality of prompts and VQA performance.

TABLE 6.7. The experimental results on QA pairs generated from different captions. The results are run with OPT 30B.

Exemplar Prompts Generation Source	OK-VQA			VQAv2 val		
	VQA Score	Answer Noise Rate	Answer Hit Rate	VQA Score	Answer Noise Rate	Answer Hit Rate
Caption from Complete Image	39.8	0.018	0.480	57.1	0.0290	0.725
Question-relevant Caption	40.6	0.022	**0.581**	58.1	0.0303	**0.821**
Question-relevant Caption with Filter	**41.8**	**0.025**	0.566	**59.5**	**0.0313**	0.804

6.4.5 Ablation on Caption Selection

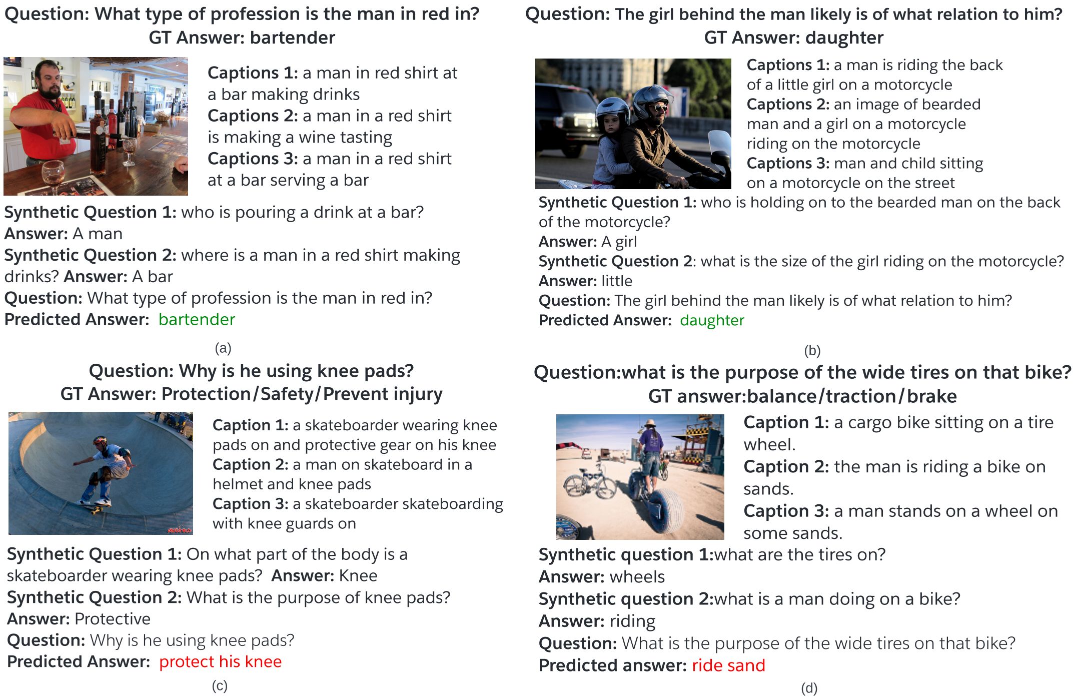
As Table 6.6 shows, we evaluate the performance different caption selection strategies, where Max Frequency selects captions containing 30 answers with highest frequencies and Min Frequency selects answers with the lowest frequencies. As the exemplar prompts are produced with answers with the highest frequencies, the Max Frequency strategy does not provide more information than exemplar prompts. In contrast, the Min Frequency strategy chooses captions that can provide some information not in the QA pairs, providing a performance boost.

6.4.6 Ablation Study on Prompt Design

We have two options to construct LLM's prompt. The first option is to append a syntheic QA pair after the caption that the QA pair is generated from. This can be described as CQA-CQA-CQA, where C, Q, A stand for caption, synthetic question, and synthetic answer respectively. Alternatively, we can present all captions at once, followed by all question-answer pairs, which we denote as CCC-QAQAQA. Experimentally (Table 6.5), the second design performs significantly better than the first. We hypothesize that the first design may induce the LLM to read only one caption before answering, since in the prompt this caption contains all the information needed for the question. While it is hard to pinpoint the actual mechanism, the results highlight the importance of QA prompts and their positions.

6.4.7 Examples and Failure Case Analysis

In Figure 6.3, we show four examples of caption and exemplar prompts and the predictions, including cases of success and failure. In Figure 6.3(a), the captions and the synthetic QA pairs provide the information that a man is making drinks at a bar. The LLM draws on background knowledge and correctly infers that his job is bartender. In Figure 6.3(c), while the prediction is understandable (even if not strictly grammatical), the LLM is unable to make inferences based on qualitative physics and predict the right answer.



Question: What type of profession is the man in red in?
GT Answer: bartender

Captions 1: a man in red shirt at a bar making drinks
Captions 2: a man in a red shirt is making a wine tasting
Captions 3: a man in a red shirt at a bar serving a bar

Synthetic Question 1: who is pouring a drink at a bar?
Answer: A man
Synthetic Question 2: where is a man in a red shirt making drinks? **Answer:** A bar
Question: What type of profession is the man in red in?
Predicted Answer: bartender

(a)

Question: **The girl behind the man likely is of what relation to him?**
GT Answer: daughter

Captions 1: a man is riding the back of a little girl on a motorcycle
Captions 2: an image of bearded man and a girl on a motorcycle riding on the motorcycle
Captions 3: man and child sitting on a motorcycle on the street

Synthetic Question 1: who is holding on to the bearded man on the back of the motorcycle?
Answer: A girl
Synthetic Question 2: what is the size of the girl riding on the motorcycle?
Answer: little
Question: The girl behind the man likely is of what relation to him?
Predicted Answer: daughter

(b)

Question: Why is he using knee pads?
GT Answer: Protection/Safety/Prevent injury

Caption 1: a skateboarder wearing knee pads on and protective gear on his knee
Caption 2: a man on skateboard in a helmet and knee pads
Caption 3: a skateboarder skateboarding with knee guards on

Synthetic Question 1: On what part of the body is a skateboarder wearing knee pads? **Answer:** Knee
Synthetic Question 2: What is the purpose of knee pads?
Answer: Protective
Question: Why is he using knee pads?
Predicted Answer: protect his knee

(c)

Question:what is the purpose of the wide tires on that bike?
GT answer:balance/traction/brake

Caption 1: a cargo bike sitting on a tire wheel.
Caption 2: the man is riding a bike on sands.
Caption 3: a man stands on a wheel on some sands.

Synthetic question 1:what are the tires on?
Answer: wheels
Synthetic question 2:what is a man doing on a bike?
Answer: riding
Question: What is the purpose of the wide tires on that bike?
Predicted answer: ride sand

(d)

FIGURE 6.3. Example predictions made by Img2LLM. Specifically, (a) and (b) are successful cases, while (c) and (d) are failure cases. See more examples at Appendix A.5.

These results highlight the importance to apply appropriate commonsense knowledge in open-ended VQA.

6.5 Limitation

One limitation of the proposed approach is that generating image captions and question-answer pairs incurs extra inference overhead. On an 8×A100 machine, our current implementation brings about 24.4% additional computational time on top of the inference time of 175B OPT. We note that further reduction of the overhead can be obtained by shortening the prompt, trading accuracy for speed. Additionally, our method avoids expensive end-to-end multimodal representation alignment, which, in the case of Flamingo, took more than 500K TPU hours.

6.6 Conclusion

In this Chapter, we propose Img2LLM, a plug-and-play module designed to exploit the knowledge and reasoning power of large language models (LLMs) off-the-shelf for zero-shot VQA tasks. Concretely, Img2LLM provides visual information and task guidance to LLMs in the format of easily-digestible prompts. This eliminates the requirement for the expensive end-to-end vision-language alignment, increasing model deployment flexibility while decreasing model deployment cost. The experiments show that Img2Prompt enables different LLMs to achieve comparable or even superior zero-shot VQA performance to other methods that require costly end-to-end training.

Bibliography

[1] Aishwarya Agrawal et al. ‘Don’t just assume; look and answer: Overcoming priors for visual question answering’. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4971–4980.

[2] Arjun Akula et al. ‘Crossvqa: Scalably generating benchmarks for systematically testing vqa generalization’. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 2148–2166.

[3] Jean-Baptiste Alayrac et al. ‘Flamingo: a Visual Language Model for Few-Shot Learning’. In: *arXiv Preprint 2204.14198* (2022). URL: https://arxiv.org/abs/2204.14198.

[4] Peter Anderson et al. ‘Bottom-up and top-down attention for image captioning and visual question answering’. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6077–6086.

[5] Stanislaw Antol et al. ‘Vqa: Visual question answering’. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2425–2433.

[6] Yuki M. Asano, Christian Rupprecht and Andrea Vedaldi. ‘Self-labelling via simultaneous clustering and representation learning’. In: *International Conference on Learning Representations (ICLR)*. 2020.

[7] Christopher G Atkeson and Juan Carlos Santamaria. ‘A comparison of direct and model-based reinforcement learning’. In: *Proceedings of international conference on robotics and automation*. Vol. 4. IEEE. 1997, pp. 3557–3564.

[8] Kamyar Azizzadenesheli et al. ‘Regularized Learning for Domain Adaptation under Label Shifts’. In: *arXiv preprint arXiv:1903.09734* (2019).

[9] Mahsa Baktashmotlagh et al. ‘Unsupervised domain adaptation by domain invariant projection’. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 769–776.

[10] Philip J Ball et al. ‘Augmented World Models Facilitate Zero-Shot Dynamics Generalization From a Single Offline Environment’. In: *arXiv preprint arXiv:2104.05632* (2021).

[11] Pratyay Banerjee et al. ‘WeaQA: Weak Supervision via Captions for Visual Question Answering’. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 3420–3435. DOI: 10.18653/v1/2021.findings-acl.302. URL: https://aclanthology.org/2021.findings-acl.302.

[12] James Bannon et al. ‘Causality and Batch Reinforcement Learning: Complementary Approaches To Planning In Unknown Domains’. In: *arXiv preprint arXiv:2006.02579* (2020).

[13] Elias Bareinboim, Andrew Forney and Judea Pearl. ‘Bandits with unobserved confounders: A causal approach’. In: *Advances in Neural Information Processing Systems* 28 (2015), pp. 1342–1350.

[14] Peyman Bateni et al. ‘Improved few-shot visual classification’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 14493–14502.

[15] Peter W Battaglia et al. ‘Relational inductive biases, deep learning, and graph networks’. In: *arXiv preprint arXiv:1806.01261* (2018).

[16] Mohamed Ishmael Belghazi et al. ‘Mine: mutual information neural estimation’. In: *arXiv preprint arXiv:1801.04062* (2018).

[17] Sagie Benaim and Lior Wolf. ‘One-sided unsupervised domain mapping’. In: *Advances in neural information processing systems*. 2017, pp. 752–762.

[18] Yuchen Bian et al. ‘On Attention Redundancy: A Comprehensive Study’. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 930–945. DOI: 10.

18653/v1/2021.naacl-main.72. URL: https://aclanthology.org/2021.naacl-main.72.

[19] Mikołaj Bińkowski et al. 'Demystifying mmd gans'. In: *arXiv preprint arXiv:1801.01401* (2018).

[20] Sid Black et al. *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*. Version 1.0. If you use this software, please cite it using these metadata. Mar. 2021. DOI: 10.5281/zenodo.5297715. URL: https://doi.org/10.5281/zenodo.5297715.

[21] Konstantinos Bousmalis et al. 'Unsupervised pixel-level domain adaptation with generative adversarial networks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3722–3731.

[22] Konstantinos Bousmalis et al. 'Using simulation and domain adaptation to improve efficiency of deep robotic grasping'. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 4243–4250.

[23] Andrew Brock, Jeff Donahue and Karen Simonyan. 'Large scale gan training for high fidelity natural image synthesis'. In: *arXiv preprint arXiv:1809.11096* (2018).

[24] Greg Brockman et al. 'Openai gym'. In: *arXiv preprint arXiv:1606.01540* (2016).

[25] Tom Brown et al. 'Language Models are Few-Shot Learners'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[26] Tom Brown et al. 'Language models are few-shot learners'. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[27] Mathilde Caron et al. 'Unsupervised Learning of Visual Features by Contrasting Cluster Assignments'. In: (2020).

[28] Mathilde Caron et al. 'Unsupervised learning of visual features by contrasting cluster assignments'. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9912–9924.

[29] Yee Seng Chan and Hwee Tou Ng. 'Word Sense Disambiguation with Distribution Estimation.' In: *IJCAI*. Vol. 5. 2005, pp. 1010–5.

[30] Soravit Changpinyo et al. 'All You May Need for VQA are Image Captions'. In: *North American Chapter of the Association for Computational Linguistics*. 2022. DOI: 10.48550/ARXIV.2205.01883. URL: https://arxiv.org/abs/2205.01883.

[31] Jiacheng Cheng et al. 'Learning with bounded instance-and label-dependent label noise'. In: *arXiv preprint arXiv:1709.03768* (2017).

[32] Jaemin Cho et al. 'Unifying Vision-and-Language Tasks via Text Generation'. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 1931–1942. URL: https://proceedings.mlr.press/v139/cho21a.html.

[33] Yunjey Choi et al. 'Stargan: Unified generative adversarial networks for multi-domain image-to-image translation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8789–8797.

[34] Aakanksha Chowdhery et al. 'Palm: Scaling language modeling with pathways'. In: *arXiv preprint arXiv:2204.02311* (2022).

[35] Kurtland Chua et al. 'Deep reinforcement learning in a handful of trials using probabilistic dynamics models'. In: *arXiv preprint arXiv:1805.12114* (2018).

[36] Marius Cordts et al. 'The cityscapes dataset for semantic urban scene understanding'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.

[37] Gabriela Csurka. 'Domain adaptation for visual applications: A comprehensive survey'. In: *arXiv preprint arXiv:1702.05374* (2017).

[38] Wenliang Dai et al. 'Enabling Multimodal Generation on CLIP via Vision-Language Knowledge Distillation'. In: *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2383–2395. DOI: 10.18653/v1/2022.findings-

acl.187. URL: https://aclanthology.org/2022.findings-acl.187.

[39] Pieter-Tjerk De Boer et al. ‘A tutorial on the cross-entropy method’. In: *Annals of operations research* 134.1 (2005), pp. 19–67.

[40] Matthias De Lange and Tinne Tuytelaars. ‘Continual prototype evolution: Learning online from non-stationary data streams’. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8250–8259.

[41] Weijian Deng, Stephen Gould and Liang Zheng. ‘On the Strong Correlation Between Model Invariance and Generalization’. In: *arXiv preprint arXiv:2207.07065* (2022).

[42] Jacob Devlin et al. ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[43] Jeff Donahue, Philipp Krähenbühl and Trevor Darrell. ‘Adversarial feature learning’. In: *arXiv preprint arXiv:1605.09782* (2016).

[44] Finale Doshi-Velez and George Konidaris. ‘Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations’. In: *IJCAI: proceedings of the conference*. Vol. 2016. NIH Public Access. 2016, p. 1432.

[45] Alexey Dosovitskiy and Thomas Brox. ‘Generating images with perceptual similarity metrics based on deep networks’. In: *arXiv preprint arXiv:1602.02644* (2016).

[46] Yilun Du and Karthic Narasimhan. ‘Task-agnostic dynamics priors for deep reinforcement learning’. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 1696–1705.

[47] Angela Fan, Mike Lewis and Yann Dauphin. ‘Hierarchical Neural Story Generation’. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 889–898. DOI: 10.18653/v1/P18-1082. URL: https://aclanthology.org/P18-1082.

[48] Tongtong Fang et al. ‘Rethinking Importance Weighting for Deep Learning under Distribution Shift’. In: *arXiv preprint arXiv:2006.04662* (2020).

[49] Chelsea Finn, Pieter Abbeel and Sergey Levine. ‘Model-agnostic meta-learning for fast adaptation of deep networks’. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1126–1135.

[50] Andrew Forney, Judea Pearl and Elias Bareinboim. ‘Counterfactual data-fusion for online reinforcement learners’. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1156–1164.

[51] Huan Fu et al. ‘Geometry-Consistent Generative Adversarial Networks for One-Sided Unsupervised Domain Mapping’. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2427–2436.

[52] Kenji Fukumizu, Francis R Bach and Michael I Jordan. ‘Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces’. In: *Journal of Machine Learning Research* 5.Jan (2004), pp. 73–99.

[53] Yaroslav Ganin et al. ‘Domain-adversarial training of neural networks’. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 2096–2030.

[54] Yaroslav Ganin et al. ‘Domain-adversarial training of neural networks’. In: *Domain Adaptation in Computer Vision Applications*. Springer, 2017, pp. 189–209.

[55] Weihao Gao et al. ‘Estimating mutual information for discrete-continuous mixtures’. In: *Advances in neural information processing systems*. 2017, pp. 5986–5997.

[56] François Gardères et al. ‘Conceptbert: Concept-aware representation for visual question answering’. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020, pp. 489–498.

[57] Leon A Gatys, Alexander S Ecker and Matthias Bethge. ‘Image style transfer using convolutional neural networks’. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423.

[58] Arnab Ghosh et al. ‘Interactive Sketch & Fill: Multiclass Sketch-to-Image Translation’. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 1171–1180.

[59] Dibya Ghosh et al. ‘Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability’. In: *Advances in Neural Information Processing Systems* 34 (2021).

[60] Mingming Gong et al. ‘Domain adaptation with conditional transferable components’. In: *International conference on machine learning*. 2016, pp. 2839–2848.

[61] Mingming Gong et al. ‘Twin Auxiliary Classifiers GAN’. In: *arXiv preprint arXiv:1907.02690* (2019).

[62] Ian Goodfellow et al. ‘Generative adversarial nets’. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[63] Omer Gottesman et al. ‘Evaluating reinforcement learning algorithms in observational health settings’. In: *arXiv preprint arXiv:1805.12298* (2018).

[64] Cyril Goutte and Eric Gaussier. ‘A probabilistic interpretation of precision, recall and F-score, with implication for evaluation’. In: *European Conference on Information Retrieval*. Springer. 2005, pp. 345–359.

[65] Yash Goyal et al. ‘Making the v in vqa matter: Elevating the role of image understanding in visual question answering’. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2017, pp. 6904–6913. URL: https://openaccess.thecvf.com/content_cvpr_2017/html/Goyal_Making_the_v_CVPR_2017_paper.html.

[66] Arthur Gretton et al. ‘A kernel two-sample test’. In: *Journal of Machine Learning Research* 13.Mar (2012), pp. 723–773.

[67] Arthur Gretton et al. ‘Covariate shift by kernel mean matching’. In: *Dataset shift in machine learning* 3.4 (2009), p. 5.

[68] Shixiang Gu et al. ‘Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates’. In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 3389–3396.

[69] Liangke Gui et al. ‘Kat: A knowledge augmented transformer for vision-and-language’. In: *arXiv preprint arXiv:2112.08614* (2021).

[70] Jiaxian Guo, Mingming Gong and Dacheng Tao. ‘A Relational Intervention Approach for Unsupervised Dynamics Generalization in Model-Based Reinforcement Learning’. In: *International Conference on Learning Representations*. 2021.

[71] Jiaxian Guo et al. ‘Long text generation via adversarial training with leaked information’. In: *arXiv preprint arXiv:1709.08624* (2017).

[72] Jiaxian Guo et al. ‘Ltf: A label transformation framework for correcting label shift’. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3843–3853.

[73] Yuanfan Guo et al. ‘HCSC: Hierarchical Contrastive Selective Coding’. In: *arXiv preprint arXiv:2202.00455* (2022).

[74] Pim de Haan, Dinesh Jayaraman and Sergey Levine. ‘Causal confusion in imitation learning’. In: *arXiv preprint arXiv:1905.11979* (2019).

[75] Danijar Hafner et al. ‘Dream to control: Learning behaviors by latent imagination’. In: *arXiv preprint arXiv:1912.01603* (2019).

[76] Danijar Hafner et al. ‘Learning latent dynamics for planning from pixels’. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2555–2565.

[77] Boris Hanin and Mark Sellke. ‘Approximating continuous functions by relu nets of minimal width’. In: *arXiv preprint arXiv:1710.11278* (2017).

[78] Hado P van Hasselt, Matteo Hessel and John Aslanides. ‘When to use parametric models in reinforcement learning?’ In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 14322–14333.

[79] Chunmei He et al. ‘Multi-attention representation network partial domain adaptation for COVID-19 diagnosis’. In: *Applied Soft Computing* 125 (2022), p. 109205.

[80] Kaiming He et al. ‘Deep Residual Learning for Image Recognition’. In: *CVPR*. 2016, pp. 770–778.

[81] Kaiming He et al. ‘Deep residual learning for image recognition’. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[82] R Devon Hjelm et al. ‘Learning deep representations by mutual information estimation and maximization’. In: *arXiv preprint arXiv:1808.06670* (2018).

[83] Judy Hoffman et al. ‘Cycada: Cycle-consistent adversarial domain adaptation’. In: *arXiv preprint arXiv:1711.03213* (2017).

[84] Kurt Hornik, Maxwell Stinchcombe and Halbert White. ‘Multilayer feedforward networks are universal approximators’. In: *Neural networks* 2.5 (1989), pp. 359–366.

[85] Han Hu et al. ‘Relation networks for object detection’. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3588–3597.

[86] Biwei Huang et al. ‘AdaRL: What, Where, and How to Adapt in Transfer Reinforcement Learning’. In: *arXiv preprint arXiv:2107.02729* (2021).

[87] Jiayuan Huang et al. ‘Correcting sample selection bias by unlabeled data’. In: *Advances in neural information processing systems*. 2007, pp. 601–608.

[88] Xun Huang et al. ‘Multimodal unsupervised image-to-image translation’. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 172–189.

[89] Lawrence Hubert and Phipps Arabie. ‘Comparing partitions’. In: *Journal of classification* 2.1 (1985), pp. 193–218.

[90] Phillip Isola et al. ‘Image-to-image translation with conditional adversarial networks’. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[91] Arun Iyer, Saketha Nath and Sunita Sarawagi. ‘Maximum mean discrepancy for class ratio estimation: Convergence bounds and kernel selection’. In: *International Conference on Machine Learning*. 2014, pp. 530–538.

[92] Ayush Jain, Andrew Szot and Joseph J Lim. ‘Generalization to new actions in reinforcement learning’. In: *arXiv preprint arXiv:2011.01928* (2020).

[93] Eric Jang, Shixiang Gu and Ben Poole. ‘Categorical reparameterization with gumbel-softmax’. In: *arXiv preprint arXiv:1611.01144* (2016).

[94] Michael Janner et al. ‘When to trust your model: Model-based policy optimization’. In: *arXiv preprint arXiv:1906.08253* (2019).

[95] Zhiwei Jia et al. ‘Semantically Robust Unpaired Image Translation for Data With Unmatched Semantics Statistics’. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 14273–14283.

[96] Haojun Jiang et al. ‘Pseudo-q: Generating pseudo language queries for visual grounding’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 15513–15523.

[97] Yu Jiang et al. ‘Pythia v0. 1: the winning entry to the vqa challenge 2018’. In: *arXiv preprint arXiv:1807.09956* (2018).

[98] Woojeong Jin et al. ‘A Good Prompt Is Worth Millions of Parameters: Low-resource Prompt-based Learning for Vision-Language Models’. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2763–2775. DOI: 10.18653/v1/2022.acl-long.197. URL: https://aclanthology.org/2022.acl-long.197.

[99] Justin Johnson, Alexandre Alahi and Li Fei-Fei. ‘Perceptual losses for real-time style transfer and super-resolution’. In: *European conference on computer vision*. Springer. 2016, pp. 694–711.

[100] Erik Jonsson and Michael Felsberg. ‘Soft Histograms for Belief Propagation’. In: (2006).

[101] Kushal Kafle and Christopher Kanan. ‘An analysis of visual question answering algorithms’. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1965–1973.

[102] Kushal Kafle, Mohammed Yousefhussien and Christopher Kanan. 'Data augmentation for visual question answering'. In: *Proceedings of the 10th International Conference on Natural Language Generation*. 2017, pp. 198–202.

[103] Lukasz Kaiser et al. 'Model-based reinforcement learning for atari'. In: *arXiv preprint arXiv:1903.00374* (2019).

[104] Nathan Kallus and Angela Zhou. 'Confounding-robust policy improvement'. In: *arXiv preprint arXiv:1805.08593* (2018).

[105] Amita Kamath et al. 'Webly Supervised Concept Expansion for General Purpose Vision Models'. In: *arXiv preprint arXiv:2202.02317* (2022).

[106] Kirthevasan Kandasamy et al. 'Nonparametric von mises estimators for entropies, divergences and mutual informations'. In: *Advances in Neural Information Processing Systems*. 2015, pp. 397–405.

[107] Oren Katzir, Dani Lischinski and Daniel Cohen-Or. 'Cross-Domain Cascaded Deep Feature Translation'. In: *arXiv* (2019), arXiv–1906.

[108] Kenji Kawaguchi, Leslie Pack Kaelbling and Yoshua Bengio. 'Generalization in deep learning'. In: *arXiv preprint arXiv:1710.05468* (2017).

[109] Charles Kemp and Joshua B Tenenbaum. 'The discovery of structural form'. In: *Proceedings of the National Academy of Sciences* 105.31 (2008), pp. 10687–10692.

[110] Daniel Khashabi et al. 'Looking beyond the surface: A challenge set for reading comprehension over multiple sentences'. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2018, pp. 252–262.

[111] Daniel Khashabi et al. 'Unifiedqa: Crossing format boundaries with a single qa system'. In: *arXiv preprint arXiv:2005.00700* (2020). URL: https://arxiv.org/abs/2202.12359.

[112] Patrick Kidger and Terry Lyons. 'Universal approximation with deep narrow networks'. In: *Conference on learning theory*. PMLR. 2020, pp. 2306–2327.

[113] Jihyung Kil et al. ‘Discovering the unknown knowns: Turning implicit knowledge in the dataset into explicit training examples for visual question answering’. In: *arXiv preprint arXiv:2109.06122* (2021).

[114] Junho Kim et al. ‘U-GAT-IT: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation’. In: *arXiv preprint arXiv:1907.10830* (2019).

[115] Taeksoo Kim et al. ‘Learning to discover cross-domain relations with generative adversarial networks’. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1857–1865.

[116] Diederik P Kingma and Jimmy Ba. ‘Adam: A method for stochastic optimization’. In: *arXiv preprint arXiv:1412.6980* (2014).

[117] B Ravi Kiran et al. ‘Deep reinforcement learning for autonomous driving: A survey’. In: *arXiv preprint arXiv:2002.00444* (2020).

[118] Robert Kirk et al. ‘A survey of generalisation in deep reinforcement learning’. In: *arXiv preprint arXiv:2111.09794* (2021).

[119] Takeshi Kojima et al. ‘Large Language Models are Zero-Shot Reasoners’. In: *arXiv preprint arXiv:2205.11916* (2022).

[120] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[121] Ilya Kostrikov, Denis Yarats and Rob Fergus. ‘Image augmentation is all you need: Regularizing deep reinforcement learning from pixels’. In: *arXiv preprint arXiv:2004.13649* (2020).

[122] LF Kozachenko and Nikolai N Leonenko. ‘Sample estimate of the entropy of a random vector’. In: *Problemy Peredachi Informatsii* 23.2 (1987), pp. 9–16.

[123] Alexander Kraskov, Harald Stögbauer and Peter Grassberger. ‘Estimating mutual information’. In: *Physical review E* 69.6 (2004), p. 066138.

[124] Akshay Krishnamurthy et al. ‘Nonparametric estimation of renyi divergence and friends’. In: *International Conference on Machine Learning*. 2014, pp. 919–927.

[125] Alex Krizhevsky and Geoffrey Hinton. *Learning multiple layers of features from tiny images*. Tech. rep. Citeseer, 2009.

[126] Daniel Kumor, Junzhe Zhang and Elias Bareinboim. ‘Sequential Causal Imitation Learning with Unobserved Confounders’. In: (2021).

[127] Thanard Kurutach et al. ‘Model-ensemble trust-region policy optimization’. In: *arXiv preprint arXiv:1802.10592* (2018).

[128] Hang Lai et al. ‘Bidirectional Model-based Policy Optimization’. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5618–5627.

[129] Finnian Lattimore, Tor Lattimore and Mark D Reid. ‘Causal bandits: Learning good interventions via causal inference’. In: *arXiv preprint arXiv:1606.03203* (2016).

[130] Tor Lattimore and Marcus Hutter. ‘No free lunch versus Occam’s razor in supervised learning’. In: *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*. Springer, 2013, pp. 223–235.

[131] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. ‘Deep learning’. In: *nature* 521.7553 (2015), pp. 436–444.

[132] Hsin-Ying Lee et al. ‘Diverse image-to-image translation via disentangled representations’. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 35–51.

[133] Hsin-Ying Lee et al. ‘DRIT++: Diverse Image-to-Image Translation via Disentangled Representations’. In: *arXiv preprint arXiv:1905.01270* (2019).

[134] Hwanhee Lee et al. ‘QACE: Asking questions to evaluate an image caption’. In: *arXiv preprint arXiv:2108.12560* (2021).

[135] Kimin Lee et al. ‘Context-aware dynamics model for generalization in model-based reinforcement learning’. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5757–5766.

[136] Sanghack Lee and Elias Bareinboim. ‘Structural causal bandits: where to intervene?’ In: *Advances in Neural Information Processing Systems 31* 31 (2018).

[137] Ian Lenz, Ross A Knepper and Ashutosh Saxena. ‘DeepMPC: Learning deep latent features for model predictive control.’ In: *Robotics: Science and Systems*. Rome, Italy. 2015.

[138] Sergey Levine and Pieter Abbeel. 'Learning Neural Network Policies with Guided Policy Search under Unknown Dynamics.' In: *NIPS*. Vol. 27. Citeseer. 2014, pp. 1071–1079.

[139] Guohao Li, Xin Wang and Wenwu Zhu. 'Boosting visual question answering with context-aware knowledge aggregation'. In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 1227–1235.

[140] Junnan Li et al. 'Align before Fuse: Vision and Language Representation Learning with Momentum Distillation'. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 9694–9705. URL: https://proceedings.neurips.cc/paper/2021/file/505259756244493872b7709a8a01b536-Paper.pdf.

[141] Junnan Li et al. 'BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation'. In: *International Conference on Machine Learning*. 2022. URL: https://arxiv.org/abs/2201.12086.

[142] Junnan Li et al. 'Prototypical contrastive learning of unsupervised representations'. In: *arXiv preprint arXiv:2005.04966* (2020).

[143] Xiujun Li et al. 'Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks'. In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX*. Ed. by Andrea Vedaldi et al. Vol. 12375. Lecture Notes in Computer Science. Springer, 2020, pp. 121–137. DOI: 10.1007/978-3-030-58577-8_8. URL: https://doi.org/10.1007/978-3-030-58577-8%5C_8.

[144] Yunfan Li et al. 'Contrastive clustering'. In: *2021 AAAI Conference on Artificial Intelligence (AAAI)*. 2021.

[145] Yuanze Lin et al. 'REVIVE: Regional Visual Representation Matters in Knowledge-Based Visual Question Answering'. In: *arXiv preprint arXiv:2206.01201* (2022).

[146] Zachary C. Lipton, Yu-Xiang Wang and Alex Smola. *Detecting and Correcting for Label Shift with Black Box Predictors*. 2018. arXiv: 1802.03916 [cs.LG].

[147] Hong Liu et al. ‘Transferable Adversarial Training: A General Approach to Adapting Deep Classifiers’. In: *International Conference on Machine Learning*. 2019, pp. 4013–4022.

[148] Ming-Yu Liu, Thomas Breuel and Jan Kautz. ‘Unsupervised image-to-image translation networks’. In: *Advances in neural information processing systems*. 2017, pp. 700–708.

[149] Ming-Yu Liu and Oncel Tuzel. ‘Coupled generative adversarial networks’. In: *Advances in neural information processing systems*. 2016, pp. 469–477.

[150] Ming-Yu Liu et al. ‘Few-shot unsupervised image-to-image translation’. In: *arXiv preprint arXiv:1905.01723* (2019).

[151] Song Liu et al. ‘Trimmed density ratio estimation’. In: *Advances in Neural Information Processing Systems*. 2017, pp. 4518–4528.

[152] Tongliang Liu and Dacheng Tao. ‘Classification with noisy labels by importance reweighting’. In: *IEEE Transactions on pattern analysis and machine intelligence* 38.3 (2015), pp. 447–461.

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

[154] Mingsheng Long et al. ‘Conditional adversarial domain adaptation’. In: *Advances in Neural Information Processing Systems*. 2018, pp. 1640–1650.

[155] Mingsheng Long et al. ‘Deep transfer learning with joint adaptation networks’. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 2208–2217.

[156] Mingsheng Long et al. ‘Learning transferable features with deep adaptation networks’. In: *International conference on machine learning*. 2015, pp. 97–105.

[157] David Lopez-Paz and Maxime Oquab. ‘Revisiting classifier two-sample tests’. In: *arXiv preprint arXiv:1610.06545* (2016).

[158] Chaochao Lu, Bernhard Schölkopf and José Miguel Hernández-Lobato. ‘Deconfounding reinforcement learning in observational settings’. In: *arXiv preprint arXiv:1812.10576* (2018).

[159] Chaochao Lu et al. 'Sample-Efficient Reinforcement Learning via Counterfactual-Based Data Augmentation'. In: *arXiv preprint arXiv:2012.09092* (2020).

[160] Jiasen Lu et al. '12-in-1: Multi-task vision and language representation learning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10437–10446.

[161] Jiasen Lu et al. 'ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/c74d97b01eae257e44aa9d5bade97baf-Paper.pdf.

[162] Zhou Lu et al. 'The expressive power of neural networks: A view from the width'. In: *Advances in neural information processing systems* 30 (2017).

[163] Hongxia Luan et al. 'Multimodality image registration by maximization of quantitative–qualitative measure of mutual information'. In: *Pattern Recognition* 41.1 (2008), pp. 285–298.

[164] Man Luo et al. 'Weakly-supervised visual-retriever-reader for knowledge-based question answering'. In: *arXiv preprint arXiv:2109.04014* (2021).

[165] Ziyang Luo et al. 'VC-GPT: Visual Conditioned GPT for End-to-End Generative Vision-and-Language Pre-training'. In: *arXiv preprint arXiv:2201.12723* (2022).

[166] Clare Lyle et al. 'Resolving Causal Confusion in Reinforcement Learning via Robust Exploration'. In: *Self-Supervision for Reinforcement Learning Workshop-ICLR 2021*. 2021.

[167] Jan Marian Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.

[168] James MacQueen et al. 'Some methods for classification and analysis of multivariate observations'. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[169] Chris J Maddison, Andriy Mnih and Yee Whye Teh. ‘The concrete distribution: A continuous relaxation of discrete random variables’. In: *arXiv preprint arXiv:1611.00712* (2016).

[170] Frederik Maes et al. ‘Multimodality image registration by maximization of mutual information’. In: *IEEE transactions on Medical Imaging* 16.2 (1997), pp. 187–198.

[171] Kenneth Marino et al. ‘Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14111–14121.

[172] Kenneth Marino et al. ‘Ok-vqa: A visual question answering benchmark requiring external knowledge’. In: *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*. 2019, pp. 3195–3204.

[173] Julieta Martinez et al. ‘A simple yet effective baseline for 3d human pose estimation’. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2640–2649.

[174] Loic Matthey et al. *dSprites: Disentanglement testing Sprites dataset*. https://github.com/deepmind/ds dataset/. 2017.

[175] Roey Mechrez, Itamar Talmi and Lihi Zelnik-Manor. ‘The contextual loss for image transformation with non-aligned data’. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 768–783.

[176] Youssef A Mejjati et al. ‘Unsupervised attention-guided image to image translation’. In: *arXiv preprint arXiv:1806.02311* (2018).

[177] Anthony Meng Huat Tiong et al. ‘Plug-and-Play VQA: Zero-shot VQA by Conjoining Large Pretrained Models with Zero Training’. In: *arXiv e-prints* (2022), arXiv–2210.

[178] Todor Mihaylov et al. ‘Can a suit of armor conduct electricity? a new dataset for open book question answering’. In: *arXiv preprint arXiv:1809.02789* (2018).

[179] Mehdi Mirza and Simon Osindero. ‘Conditional generative adversarial nets’. In: *arXiv preprint arXiv:1411.1784* (2014).

[180] Takeru Miyato and Masanori Koyama. ‘cGANs with projection discriminator’. In: *arXiv preprint arXiv:1802.05637* (2018).

[181] Takeru Miyato et al. ‘Spectral normalization for generative adversarial networks’. In: *arXiv preprint arXiv:1802.05957* (2018).

[182] Volodymyr Mnih et al. ‘Playing atari with deep reinforcement learning’. In: *arXiv preprint arXiv:1312.5602* (2013).

[183] Ron Mokady, Amir Hertz and Amit H Bermano. ‘Clipcap: Clip prefix for image captioning’. In: *arXiv preprint arXiv:2111.09734* (2021).

[184] Kevin R Moon, Kumar Sricharan and Alfred O Hero. ‘Ensemble estimation of mutual information’. In: *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE. 2017, pp. 3030–3034.

[185] Melissa Mozifian et al. ‘Intervention Design for Effective Sim2Real Transfer’. In: *arXiv preprint arXiv:2012.02055* (2020).

[186] Anusha Nagabandi, Chelsea Finn and Sergey Levine. ‘Deep online learning via meta-learning: Continual adaptation for model-based rl’. In: *arXiv preprint arXiv:1812.07671* (2018).

[187] Anusha Nagabandi et al. ‘Learning to adapt in dynamic, real-world environments through meta-reinforcement learning’. In: *arXiv preprint arXiv:1803.11347* (2018).

[188] Tuan Duong Nguyen, Marthinus Christoffel and Masashi Sugiyama. ‘Continuous target shift adaptation in supervised learning’. In: *Asian Conference on Machine Learning*. 2016, pp. 285–300.

[189] Aaron van den Oord, Yazhe Li and Oriol Vinyals. ‘Representation learning with contrastive predictive coding’. In: *arXiv preprint arXiv:1807.03748* (2018).

[190] Charles Packer et al. *Assessing Generalization in Deep Reinforcement Learning*. 2019. arXiv: 1810.12282 [cs.LG].

[191] Sinno Jialin Pan et al. ‘Domain adaptation via transfer component analysis’. In: *IEEE Transactions on Neural Networks* 22.2 (2010), pp. 199–210.

[192] Liam Paninski. ‘Estimation of entropy and mutual information’. In: *Neural computation* 15.6 (2003), pp. 1191–1253.

[193] Taesung Park et al. ‘Contrastive Learning for Unpaired Image-to-Image Translation’. In: *arXiv preprint arXiv:2007.15651* (2020).

[194] Massimiliano Patacchiola and Amos Storkey. ‘Self-supervised relational reasoning for representation learning’. In: *arXiv preprint arXiv:2006.05849* (2020).

[195] Deepak Pathak et al. ‘Context encoders: Feature learning by inpainting’. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2536–2544.

[196] J Pearl. ‘Causality: Models, Reasoning, and Inference 47Cambridge University PressCambridge, United Kingdom. Pearl, J. 2000’. In: *Causality: models, reasoning, and inference* 47 (2000).

[197] Judea Pearl. ‘Direct and indirect effects’. In: *arXiv preprint arXiv:1301.2300* (2013).

[198] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[199] Judea Pearl, Madelyn Glymour and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[200] Brenden K Petersen et al. ‘Precision medicine as a control problem: Using simulation and deep reinforcement learning to discover adaptive, personalized multi-cytokine therapy for sepsis’. In: *arXiv preprint arXiv:1802.10440* (2018).

[201] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[202] Alec Radford, Luke Metz and Soumith Chintala. ‘Unsupervised representation learning with deep convolutional generative adversarial networks’. In: *arXiv preprint arXiv:1511.06434* (2015).

[203] Colin Raffel et al. ‘Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer’. In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[204] Roberta Raileanu et al. ‘Fast adaptation to new environments via policy-dynamics value functions’. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7920–7931.

[205] Pranav Rajpurkar, Robin Jia and Percy Liang. ‘Know what you don’t know: Unanswerable questions for SQuAD’. In: *arXiv preprint arXiv:1806.03822* (2018).

[206] Kate Rakelly et al. ‘Efficient off-policy meta-reinforcement learning via probabilistic context variables’. In: *International conference on machine learning*. PMLR. 2019, pp. 5331–5340.

[207] David Raposo et al. ‘Discovering objects and their relations from entangled scene representations’. In: *arXiv preprint arXiv:1702.05068* (2017).

[208] Stephan R Richter et al. ‘Playing for data: Ground truth from computer games’. In: *European conference on computer vision*. Springer. 2016, pp. 102–118.

[209] Andrew Rosenberg and Julia Hirschberg. ‘V-measure: A conditional entropy-based external cluster evaluation measure’. In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*. 2007, pp. 410–420.

[210] Steindór Sæmundsson, Katja Hofmann and Marc Peter Deisenroth. ‘Meta reinforcement learning with latent variable gaussian processes’. In: *arXiv preprint arXiv:1803.07551* (2018).

[211] Adam Santoro et al. ‘A simple neural network module for relational reasoning’. In: *arXiv preprint arXiv:1706.01427* (2017).

[212] Adam Santoro et al. ‘Relational recurrent neural networks’. In: *arXiv preprint arXiv:1806.01822* (2018).

[213] Maarten Sap et al. ‘Socialiqa: Commonsense reasoning about social interactions’. In: *arXiv preprint arXiv:1904.09728* (2019).

[214] Teven Le Scao et al. ‘What Language Model to Train if You Have One Million GPU Hours?’ In: *arXiv preprint arXiv:2210.15424* (2022).

[215] Julian Schrittwieser et al. ‘Mastering atari, go, chess and shogi by planning with a learned model’. In: *Nature* 588.7839 (2020), pp. 604–609.

[216] Dustin Schwenk et al. ‘A-OKVQA: A Benchmark for Visual Question Answering using World Knowledge’. In: *arXiv preprint arXiv:2206.01718* (2022).

[217] Matan Sela, Elad Richardson and Ron Kimmel. ‘Unrestricted facial geometry reconstruction using image-to-image translation’. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1576–1585.

[218] Ramprasaath R. Selvaraju et al. ‘Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization’. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74.

[219] Younggyo Seo et al. ‘Trajectory-wise Multiple Choice Learning for Dynamics Generalization in Reinforcement Learning’. In: *arXiv preprint arXiv:2010.13303* (2020).

[220] Zhiqiang Shen et al. ‘Towards Instance-level Image-to-Image Translation’. In: *arXiv preprint arXiv:1905.01744* (2019).

[221] Hidetoshi Shimodaira. ‘Improving predictive inference under covariate shift by weighting the log-likelihood function’. In: *Journal of statistical planning and inference* 90.2 (2000), pp. 227–244.

[222] Ashish Shrivastava et al. ‘Learning from simulated and unsupervised images through adversarial training’. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2107–2116.

[223] Si Si, Dacheng Tao and Bo Geng. ‘Bregman divergence-based regularization for transfer subspace learning’. In: *IEEE Transactions on Knowledge and Data Engineering* 22.7 (2009), pp. 929–942.

[224] Md Mahfuzur Rahman Siddiquee et al. ‘Learning Fixed Points in Generative Adversarial Networks: From Image-to-Image Translation to Disease Detection and Localization’. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 191–200.

[225] David Silver et al. ‘A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play’. In: *Science* 362.6419 (2018), pp. 1140–1144.

[226] David Silver et al. ‘Mastering the game of Go with deep neural networks and tree search’. In: *nature* 529.7587 (2016), pp. 484–489.

[227] David Silver et al. ‘Mastering the game of go without human knowledge’. In: *nature* 550.7676 (2017), pp. 354–359.

[228] Christian Simon et al. ‘Adaptive subspaces for few-shot learning’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 4136–4145.

[229] Amanpreet Singh et al. ‘FLAVA: A Foundational Language And Vision Alignment Model’. In: *arXiv preprint arXiv:2112.04482* (2021).

[230] Shashank Singh and Barnabás Póczos. ‘Exponential concentration of a density functional estimator’. In: *Advances in Neural Information Processing Systems*. 2014, pp. 3032–3040.

[231] Shashank Singh and Barnabás Póczos. ‘Generalized exponential concentration inequality for Rényi divergence estimation’. In: *International Conference on Machine Learning*. 2014, pp. 333–341.

[232] Jake Snell, Kevin Swersky and Richard Zemel. ‘Prototypical networks for few-shot learning’. In: *Advances in neural information processing systems* 30 (2017).

[233] Robyn Speer, Joshua Chin and Catherine Havasi. ‘Conceptnet 5.5: An open multilingual graph of general knowledge’. In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[234] Petar Stojanov et al. ‘Low-Dimensional Density Ratio Estimation for Covariate Shift Correction’. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 3449–3458.

[235] Masashi Sugiyama. ‘Machine learning with squared-loss mutual information’. In: *Entropy* 15.1 (2013), pp. 80–112.

[236] Masashi Sugiyama, Taiji Suzuki and Takafumi Kanamori. ‘Density-ratio matching under the Bregman divergence: a unified framework of density-ratio estimation’. In: *Annals of the Institute of Statistical Mathematics* 64.5 (2012), pp. 1009–1044.

[237] Masashi Sugiyama, Taiji Suzuki and Takafumi Kanamori. ‘Distribution Comparison’. In: *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012, pp. 140–162. DOI: 10.1017/CBO9781139035613.014.

[238] Masashi Sugiyama et al. ‘Density-difference estimation’. In: *Neural Computation* 25.10 (2013), pp. 2734–2775.

[239] Masashi Sugiyama et al. ‘Direct importance estimation with model selection and its application to covariate shift adaptation’. In: *Advances in neural information processing systems*. 2008, pp. 1433–1440.

[240] Xinwei Sun et al. ‘Recovering latent causal factor for generalization to distributional shifts’. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 16846–16859.

[241] Flood Sung et al. ‘Learning to compare: Relation network for few-shot learning’. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1199–1208.

[242] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[243] Taiji Suzuki et al. ‘Approximating mutual information by maximum likelihood density ratio estimation’. In: *New challenges for feature selection in data mining and knowledge discovery*. 2008, pp. 5–20.

[244] Taiji Suzuki et al. ‘Mutual information estimation reveals global associations between stimuli and biological processes’. In: *BMC bioinformatics* 10.1 (2009), S52.

[245] Siham Tabik et al. ‘COVIDGR dataset and COVID-SDNet methodology for predicting COVID-19 based on chest X-ray images’. In: *IEEE journal of biomedical and health informatics* 24.12 (2020), pp. 3595–3605.

[246] Yaniv Taigman, Adam Polyak and Lior Wolf. ‘Unsupervised cross-domain image generation’. In: *arXiv preprint arXiv:1611.02200* (2016).

[247] Alon Talmor et al. ‘Commonsenseqa: A question answering challenge targeting commonsense knowledge’. In: *arXiv preprint arXiv:1811.00937* (2018).

[248] Hao Tan and Mohit Bansal. ‘LXMERT: Learning Cross-Modality Encoder Representations from Transformers’. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong

Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5100–5111. DOI: 10.18653/v1/D19-1514. URL: https://aclanthology.org/D19-1514.

[249] Hao Tang et al. 'Attention-guided generative adversarial networks for unsupervised image-to-image translation'. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–8.

[250] Hao Tang et al. 'Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks'. In: *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[251] Korawat Tanwisuth et al. 'A Prototype-Oriented Framework for Unsupervised Domain Adaptation'. In: *Advances in Neural Information Processing Systems* 34 (2021).

[252] Dirk Tasche. 'Fisher consistency for prior probability shift'. In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 3338–3369.

[253] Yuval Tassa, Tom Erez and Emanuel Todorov. 'Synthesis and stabilization of complex behaviors through online trajectory optimization'. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 4906–4913.

[254] Emanuel Todorov, Tom Erez and Yuval Tassa. 'Mujoco: A physics engine for model-based control'. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5026–5033.

[255] Matteo Tomei et al. 'Art2Real: Unfolding the Reality of Artworks via Semantically-Aware Image-to-Image Translation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5849–5859.

[256] Yao-Hung Hubert Tsai et al. 'Neural methods for point-wise dependency estimation'. In: *arXiv preprint arXiv:2006.05553* (2020).

[257] Maria Tsimpoukelli et al. 'Multimodal Few-Shot Learning with Frozen Language Models'. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 200–212.

URL: https://proceedings.neurips.cc/paper/2021/file/01b7575c38dac42f3cfb7d500438b875-Paper.pdf.

[258] Eric Tzeng et al. 'Deep domain confusion: Maximizing for domain invariance'. In: *arXiv preprint arXiv:1412.3474* (2014).

[259] Nguyen Xuan Vinh, Julien Epps and James Bailey. 'Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance'. In: *The Journal of Machine Learning Research* 11 (2010), pp. 2837–2854.

[260] Oriol Vinyals et al. 'Grandmaster level in StarCraft II using multi-agent reinforcement learning'. In: *Nature* 575.7782 (2019), pp. 350–354.

[261] Sergei Volodin, Nevan Wichers and Jeremy Nixon. 'Resolving spurious correlations in causal models of environments via interventions'. In: *arXiv preprint arXiv:2002.05217* (2020).

[262] Ben Wang and Aran Komatsuzaki. *GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model*. https://github.com/kingoflolz/mesh-transformer-jax. May 2021.

[263] Kaixin Wang et al. 'Improving generalization in reinforcement learning with mixture regularization'. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7968–7978.

[264] Lingxiao Wang, Zhuoran Yang and Zhaoran Wang. 'Provably efficient causal reinforcement learning with confounded observational data'. In: *arXiv preprint arXiv:2006.12311* (2020).

[265] Qi Wang and Herke van Hoof. 'Model-based Meta Reinforcement Learning using Graph Structured Surrogate Models'. In: *arXiv preprint arXiv:2102.08291* (2021).

[266] Ting-Chun Wang et al. 'High-resolution image synthesis and semantic manipulation with conditional gans'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8798–8807.

[267] Xudong Wang, Ziwei Liu and Stella X Yu. 'Unsupervised feature learning by cross-level instance-group discrimination'. In: *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition. 2021, pp. 12586–12595.

[268] Zhen Wang, Liu Liu and Dacheng Tao. 'Deep Streaming Label Learning'. In: *International Conference on Machine Learning (ICML)*. 2020, pp. 378–387.

[269] Zhen Wang, Liu Liu and Dacheng Tao. 'Deep streaming label learning'. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9963–9972.

[270] Zhen Wang et al. 'Continual Learning Through Retrieval and Imagination'. In: *AAAI Conference on Artificial Intelligence*. 2022.

[271] Zhen Wang et al. 'Dbsvec: Density-based clustering using support vector expansion'. In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE. 2019, pp. 280–291.

[272] Zirui Wang et al. 'SimVLM: Simple visual language model pretraining with weak supervision'. In: *International Conference on Learning Representations, ICLR 2022*. OpenReview.net, 2022. URL: https://openreview.net/forum?id=GUrhfTuf_3.

[273] Jason Wei et al. 'Chain of thought prompting elicits reasoning in large language models'. In: *arXiv preprint arXiv:2201.11903* (2022).

[274] Jason Wei et al. 'Emergent abilities of large language models'. In: *arXiv preprint arXiv:2206.07682* (2022).

[275] Jason Wei et al. 'Emergent abilities of large language models'. In: *arXiv preprint arXiv:2206.07682* (2022).

[276] Darrell Whitley and Jean Paul Watson. 'Complexity theory and the no free lunch theorem'. In: *Search methodologies* (2005), pp. 317–339.

[277] William Whitney et al. 'Dynamics-aware embeddings'. In: *arXiv preprint arXiv:1908.09357* (2019).

[278] Russell A Wilke et al. 'Identifying genetic risk factors for serious adverse drug reactions: current progress and challenges'. In: *Nature reviews Drug discovery* 6.11 (2007), pp. 904–916.

[279] Ronald J Williams. ‘Simple statistical gradient-following algorithms for connectionist reinforcement learning’. In: *Machine learning* 8.3-4 (1992), pp. 229–256.

[280] Garrett Wilson and Diane J Cook. ‘A survey of unsupervised deep domain adaptation’. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 11.5 (2020), pp. 1–46.

[281] David H Wolpert and William G Macready. ‘No free lunch theorems for optimization’. In: *IEEE transactions on evolutionary computation* 1.1 (1997), pp. 67–82.

[282] Jialin Wu, Zeyuan Hu and Raymond J Mooney. ‘Generating question relevant captions to aid visual question answering’. In: *arXiv preprint arXiv:1906.00513* (2019).

[283] Jialin Wu et al. ‘Multi-modal answer validation for knowledge-based vqa’. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 3. 2022, pp. 2712–2721.

[284] Wayne Wu et al. ‘Transgaga: Geometry-aware unsupervised image-to-image translation’. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8012–8021.

[285] Sirui Xie et al. ‘SNAS: stochastic neural architecture search’. In: *International Conference on Learning Representations*. 2018.

[286] Xing Xu et al. ‘Radial graph convolutional network for visual question generation’. In: *IEEE transactions on neural networks and learning systems* 32.4 (2020), pp. 1654–1667.

[287] Yufei Xu et al. ‘Vitae: Vision transformer advanced by exploring intrinsic inductive bias’. In: *Advances in Neural Information Processing Systems* 34 (2021).

[288] Makoto Yamada et al. ‘Relative density-ratio estimation for robust distribution comparison’. In: *Neural computation* 25.5 (2013), pp. 1324–1370.

[289] Jiachen Yang et al. ‘Single episode policy transfer in reinforcement learning’. In: *arXiv preprint arXiv:1910.07719* (2019).

[290] Xue Yang et al. 'Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network'. In: *IEEE Access* 6 (2018), pp. 50839–50849.

[291] Yanchao Yang et al. 'Phase consistent ecological domain adaptation'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9011–9020.

[292] Yuxiang Yang et al. 'Data efficient reinforcement learning for legged robots'. In: *Conference on Robot Learning*. PMLR. 2020, pp. 1–10.

[293] Zhengyuan Yang et al. 'An empirical study of GPT-3 for few-shot knowledge-based VQA'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022. URL: https://arxiv.org/abs/2109.05014.

[294] Zichao Yang et al. 'Stacked attention networks for image question answering'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 21–29.

[295] Denis Yarats et al. 'Reinforcement Learning with Prototypical Representations'. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. 2021, pp. 11920–11931.

[296] Zili Yi et al. 'Dualgan: Unsupervised dual learning for image-to-image translation'. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2849–2857.

[297] Tom Young et al. 'Recent trends in deep learning based natural language processing'. In: *ieee Computational intelligenCe magazine* 13.3 (2018), pp. 55–75.

[298] Lantao Yu et al. 'Seqgan: Sequence generative adversarial nets with policy gradient'. In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[299] Lu Yu et al. 'Semantic drift compensation for class-incremental learning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6982–6991.

[300] Xiyu Yu et al. 'Transfer learning with label noise'. In: *arXiv preprint arXiv:1707.09724* (2017).

[301] Desen Yuan. 'Language bias in visual question answering: A survey and taxonomy'. In: *arXiv preprint arXiv:2111.08531* (2021).

[302] Lu Yuan et al. 'Florence: A new foundation model for computer vision'. In: *arXiv preprint arXiv:2111.11432* (2021).

[303] Bianca Zadrozny. 'Learning and evaluating classifiers under sample selection bias'. In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 114.

[304] Vinicius Zambaldi et al. 'Deep reinforcement learning with relational inductive biases'. In: *International Conference on Learning Representations*. 2018.

[305] Amy Zhang et al. 'Invariant causal prediction for block mdps'. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11214–11224.

[306] Amy Zhang et al. 'Learning causal state representations of partially observable environments'. In: *arXiv preprint arXiv:1906.10437* (2019).

[307] Amy Zhang et al. 'Learning Robust State Abstractions for Hidden-Parameter Block MDPs'. In: *International Conference on Learning Representations*. 2020.

[308] Junzhe Zhang and Elias Bareinboim. 'Fairness in decision-making—the causal explanation formula'. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[309] Junzhe Zhang, Daniel Kumor and Elias Bareinboim. 'Causal imitation learning with unobserved confounders'. In: *Advances in neural information processing systems* 33 (2020).

[310] Kun Zhang et al. 'Domain adaptation as a problem of inference on graphical models'. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 4965–4976.

[311] Kun Zhang et al. 'Domain adaptation under target and conditional shift'. In: *International Conference on Machine Learning*. 2013, pp. 819–827.

[312] Marvin Zhang et al. 'Solar: Deep structured representations for model-based reinforcement learning'. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7444–7453.

[313] Pengchuan Zhang et al. ‘Vinvl: Revisiting visual representations in vision-language models’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 5579–5588.

[314] Qiming Zhang et al. ‘ViTAEv2: Vision Transformer Advanced by Exploring Inductive Bias for Image Recognition and Beyond’. In: *arXiv preprint arXiv:2202.10108* (2022).

[315] Rui Zhang, Tomas Pfister and Jia Li. ‘Harmonic unpaired image-to-image translation’. In: *arXiv preprint arXiv:1902.09727* (2019).

[316] Susan Zhang et al. ‘Opt: Open pre-trained transformer language models’. In: *arXiv preprint arXiv:2205.01068* (2022).

[317] Yifan Zhang et al. ‘COVID-DA: Deep domain adaptation from typical pneumonia to COVID-19’. In: *arXiv preprint arXiv:2005.01577* (2020).

[318] Shuai Zhao et al. ‘Region Mutual Information Loss for Semantic Segmentation’. In: *Advances in Neural Information Processing Systems*. 2019, pp. 11115–11125.

[319] Denny Zhou et al. ‘Least-to-Most Prompting Enables Complex Reasoning in Large Language Models’. In: *arXiv Preprint 2205.10625* (2022). URL: https://arxiv.org/abs/2205.10625.

[320] Wenxuan Zhou, Lerrel Pinto and Abhinav Gupta. ‘Environment probing interaction policies’. In: *arXiv preprint arXiv:1907.11740* (2019).

[321] Jun-Yan Zhu et al. ‘Unpaired image-to-image translation using cycle-consistent adversarial networks’. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

[322] Barbara Zitova and Jan Flusser. ‘Image registration methods: a survey’. In: *Image and vision computing* 21.11 (2003), pp. 977–1000.

APPENDIX A

Appendix of Chapter 2

A0.1 Proof of Proposition 1

PROOF. Let h^c be complementary function of h, such that $X = H(h(X), h^c(X))$. Let $X_1 = h(X)$ and $X_2 = h^c(X)$, then we have $p_{X,Y}(x, y) = p_{X_1,X_2,Y}(x_1, x_2, y)$. Using the decomposition $p_{X_1,X_2,Y}(x_1, x_2, y) = p_{X_1}(x_1)p_{X_2,Y|X_1}(x_2, y|x_1)$, we have

$$I(Y, X) = I(Y, X_1) + E_{X_1}[I(Y|X_1, X_2|X_1)], \tag{A.1}$$

where $I(\cdot, \cdot) \geq 0$ is the mutual information. By maximizing $I(Y, X_1)$, the best solution we can achieve is $I(Y, X_1) = I(Y, X)$, which implies $I(Y|X_1, X_2|X_1) = 0$. This means the conditional independence of Y and X_2 given X_1, i.e., $Y \perp\!\!\!\perp X_2|X_1$, which is equivalent to $Y \perp\!\!\!\perp X|h(X)$. Then it suffices to show that maximizing mutual information $I(Y, X_1)$ is equivalent to minimizing the cross-entropy loss or mean squared loss under some parametric assumptions.

We first expand the mutual information $I(Y, X_1)$ as

$$\begin{aligned} I(Y, X_1) &= H(Y) - H(Y|X_1) \\ &= H(Y) + \int p(y, x_1) \log p(y|x_1) dy dx_1. \end{aligned} \tag{A.2}$$

For regression problems, we use $q(y|x_1) = N(w^T x_1|\sigma^2)$ to approximate $p(y|x_1)$ and write (A.2) as

$$I(Y, X_1) \approx H(Y) - \frac{1}{2\sigma^2} \int p(y, x_1)(y - w^T x_1)^2 dy dx_1. \tag{A.3}$$

It is straightforward to see that maximizing $I(Y, X_1)$ is equivalent minimizing the mean squared loss. For classification, we use $q(y = k|x_1) = \frac{\exp w_k^T x_1}{\sum_{k'=1}^K \exp w_{k'}^T x_1}$ to approximate $p(y|x_1)$ and rewrite (A.2) as

$$I(Y, X_1) \approx H(Y) + \int \sum_{k=1}^{K} p(y = k, x_1) \frac{\exp w_k^T x_1}{\sum_{k'=1}^K \exp w_{k'}^T x_1} dx_1. \tag{A.4}$$

Therefore, maximizing $I(Y, X_1)$ is equivalent minimizing the cross-entropy loss for classification. □

A0.2 Results of Fashion-MNIST and MNIST

A0.2.1 Results of CIFAR-10

The details of experimental settings of CIFAR-10 could be found at the table A.1, and the results of CIFAR-10 could be found at the main paper.

Classifier Details	
Architecture	Resnet-18
Batch Size	128
Training epochs	20
Optimizer	SGD
Learning Rate	1e-2
L2 Penalty Parameter	5e-4
Label Transformation Details	
Architecture	One-Layer Network
Label Influence Recovery Details	
Generator Architecture	BigGAN
Training Method	BigGAN [23]
Distribution Matching	
Optimizer	Adam
Learning Rate	8e-5
Training epochs	1000

TABLE A.1. The experimental details on CIFAR-10 dataset.

A0.2.2 Results of Fashion-MNIST

The details of experimental settings of Fashion-MNIST could be found at the table A.2. The MSE error of the estimated label weights P_Y^T/P_Y^S, accuracy and F1 score of FASHION-MNIST are shown as Figure A.1, A.2, A.3.

Classifier Details	
Architecture	Discriminator of DCGAN
Batch Size	128
Training epochs	20
Optimizer	SGD
Learning Rate	1e-2
L2 Penalty Parameter	5e-4
Label Transformation Details	
Architecture	One-Layer Network
Label Influence Recovery Details	
Generator Architecture	Generator of DCGAN
Training Method	TAC-GAN [61]
Distribution Matching	
Optimizer	Adam
Learning Rate	8e-5
Training epochs	1000

TABLE A.2. The experimental details on FASHION-MNIST dataset.

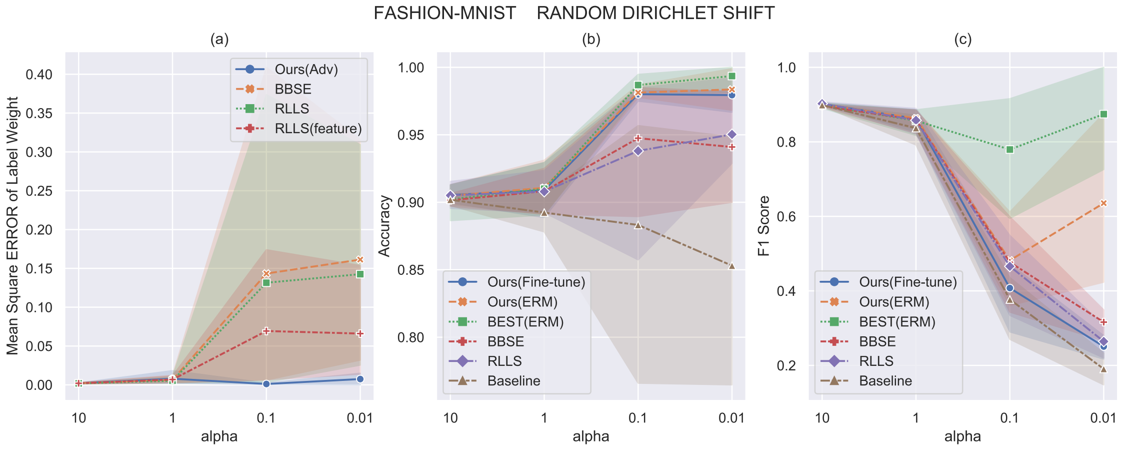


FIGURE A.1. (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score (Higher is better) on FASHION-MNIST for uniform training set and random Dirichlet shifted test set, where smaller *alpha* corresponds to bigger shift.

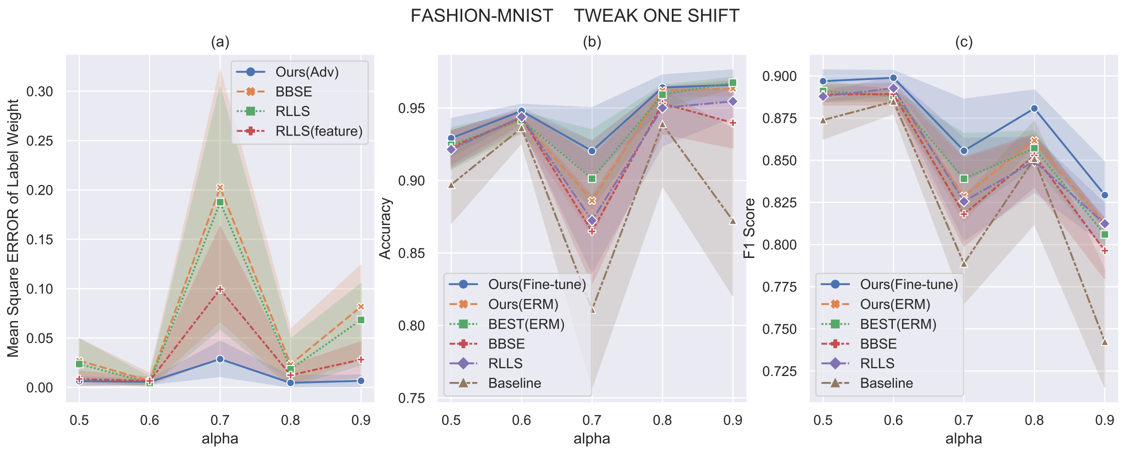


FIGURE A.2. (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score (Higher is better) on FASHION-MNIST for uniform training set and Tweak-One shifted test set, where *alpha* is the probability of tweaked class.

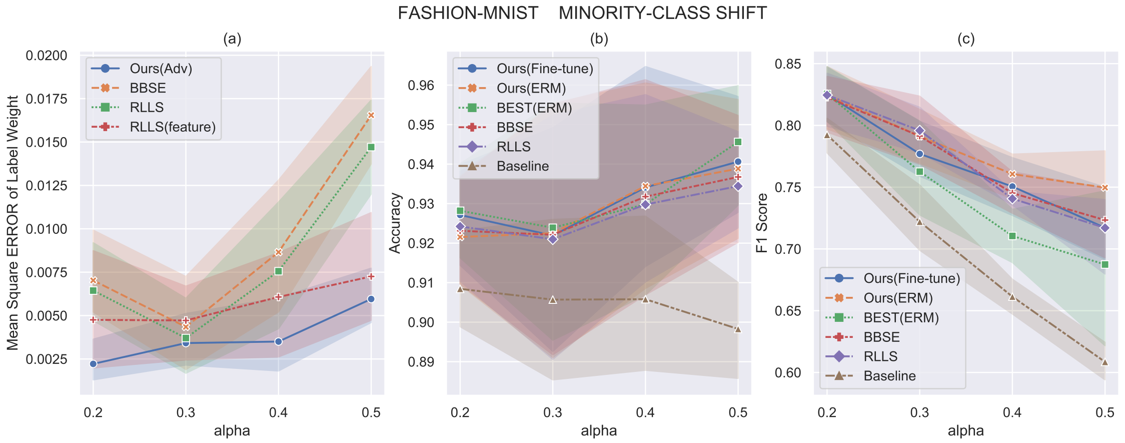


FIGURE A.3. (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score (Higher is better) on FASHION-MNIST for uniform training set and minority-class shifted test set, where *alpha* is the ratio of minority classes.

A0.2.3 Results of MNIST

The details of experimental settings of MNIST could be found at the table A.3. The MSE error of the estimated label weights P_Y^T/P_Y^S, accuracy and F1 score of MNIST are shown as Figure A.4, A.5, A.6.

Classifier Details	
Architecture	Two-layer Network
Batch Size	128
Training epochs	20
Optimizer	SGD
Learning Rate	1e-2
L2 Penalty Parameter	5e-4
Label Transformation Details	
Architecture	One-Layer Network
Label Influence Recovery Details	
Generator Architecture	Four-layer Network
Training Method	TAC-GAN [61]
Distribution Matching	
Optimizer	Adam
Learning Rate	8e-5
Training epochs	1000

TABLE A.3. The experimental details on MNIST dataset.

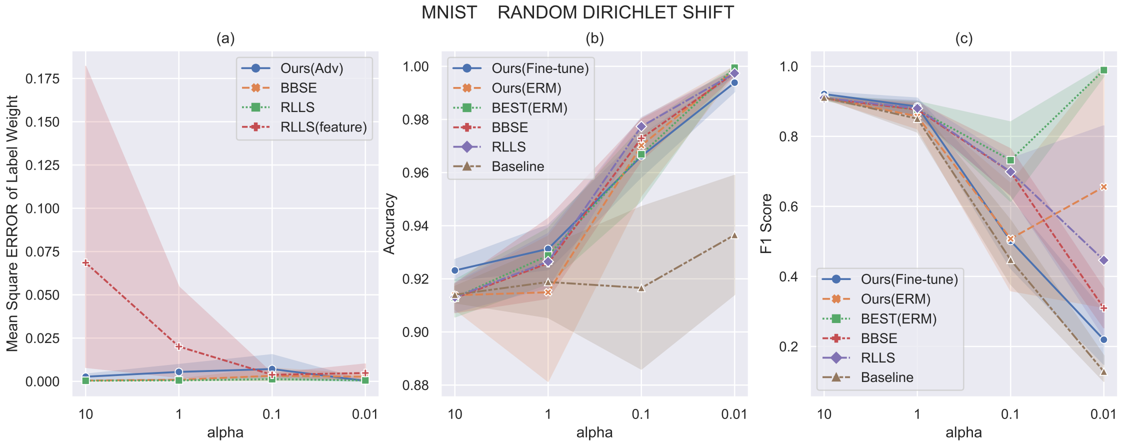


FIGURE A.4. (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score (Higher is better) on MNIST for uniform training set and random Dirichlet shifted test set, where smaller *alpha* corresponds to bigger shift.

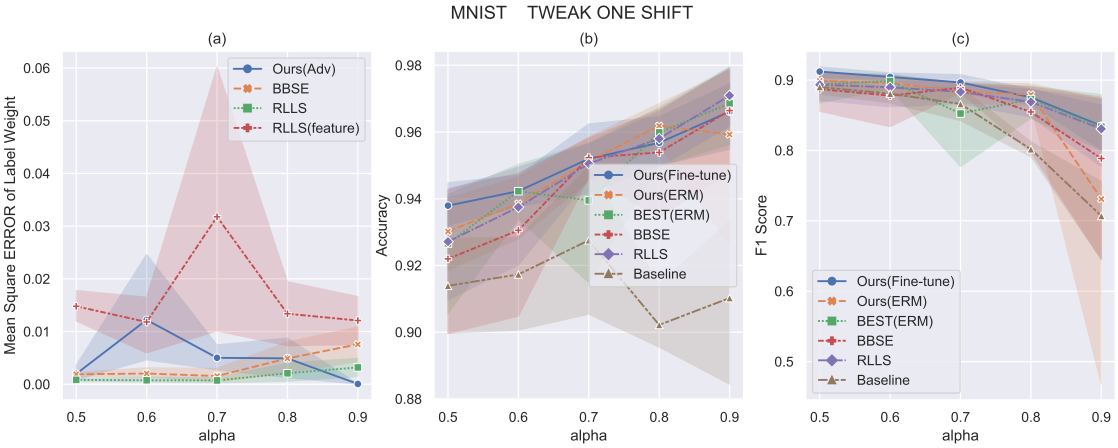


FIGURE A.5. (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score (Higher is better) on MNIST for uniform training set and Tweak-One shifted test set, where *alpha* is the probability of tweaked class.

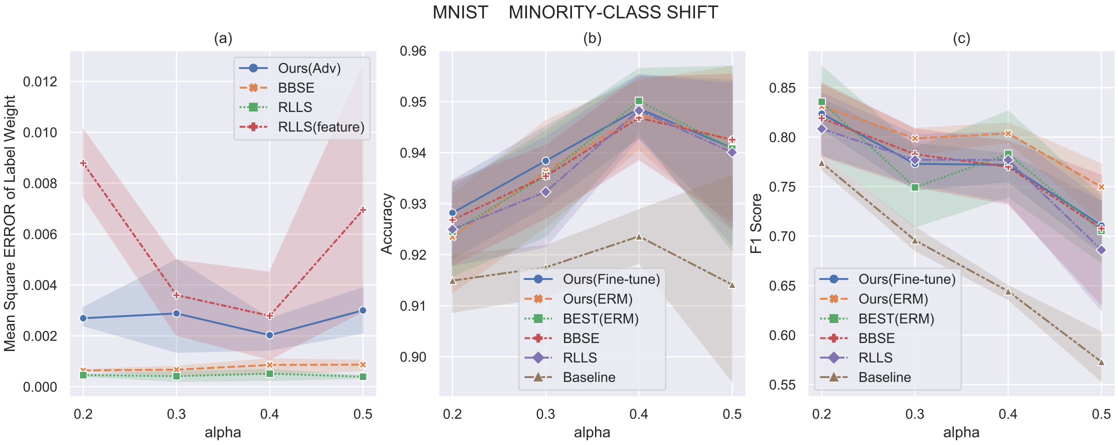


FIGURE A.6. (a) Mean squared error of estimated label weights (Lower is better), (b) accuracy and (c) F-1 score on MNIST for uniform training set and minority-class shifted test set, where *alpha* is the ratio of minority classes.

A0.3 Label Weights Visualization of Continuous Synthetic Data Experimens

Regressor Details	
Architecture	Three-layer Network
Batch Size	64
Training epochs	1000
Optimizer	Adam
Learning Rate	1e-3
Label Transformation Details	
Architecture	Three-layer Network
Label Influence Recovery Details	
Generator Architecture	Three-layer Network
Training Method	TAC-GAN [61]
Distribution Matching	
Optimizer	Adam
Learning Rate	1e-3
Training epochs	10000

TABLE A.4. The experimental details on Moon Synthetic dataset.

A0.3.1 Results of Shift A

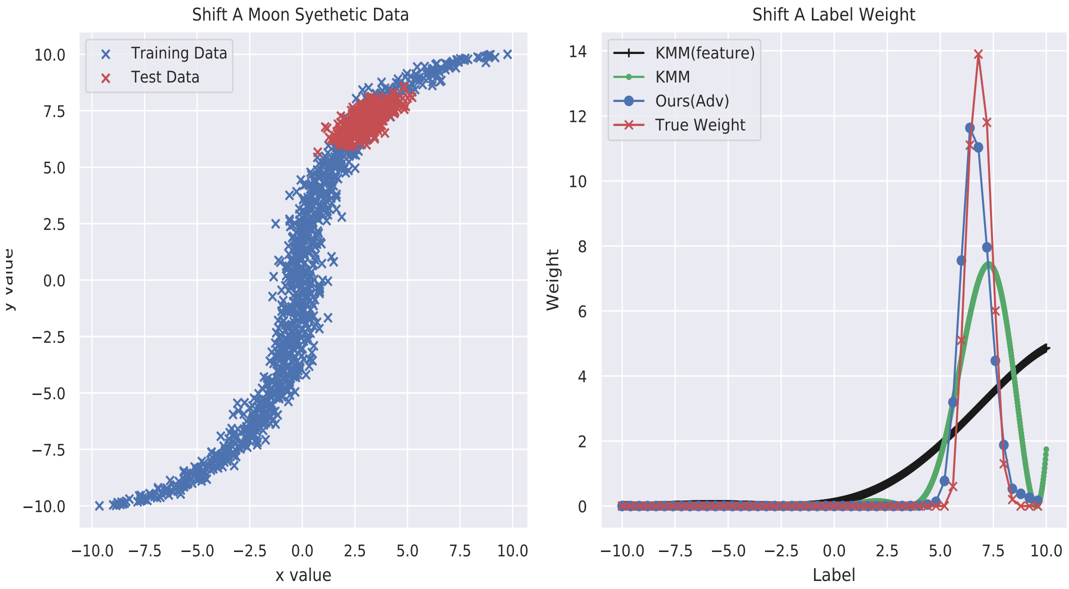


FIGURE A.7. (a) The illustration of Moon Synthetic Data (Shift A, 1st experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

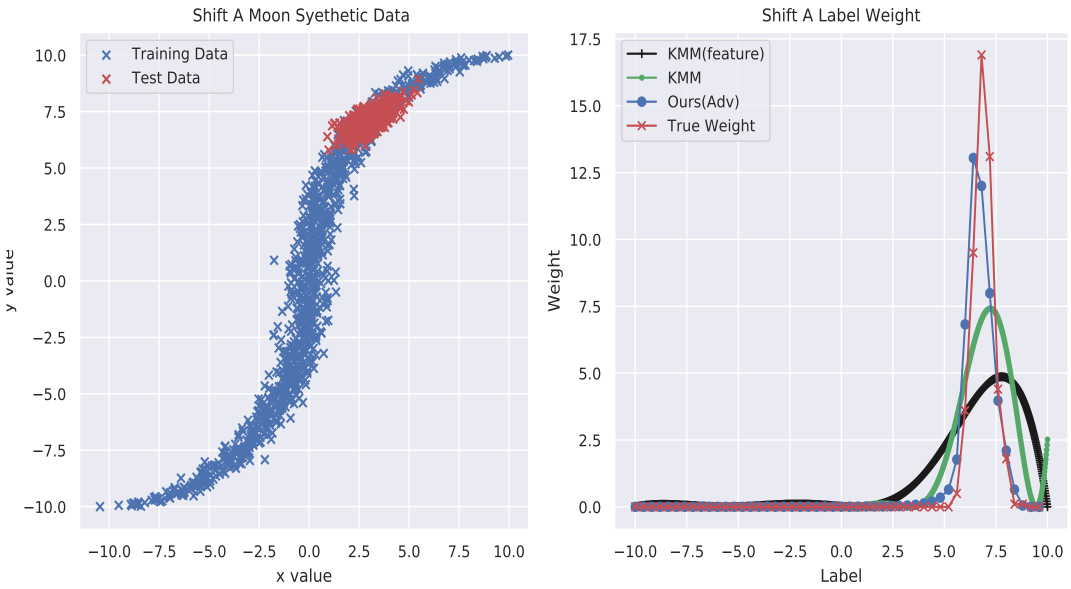


FIGURE A.8. (a) The illustration of Moon Synthetic Data (Shift A, 2nd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

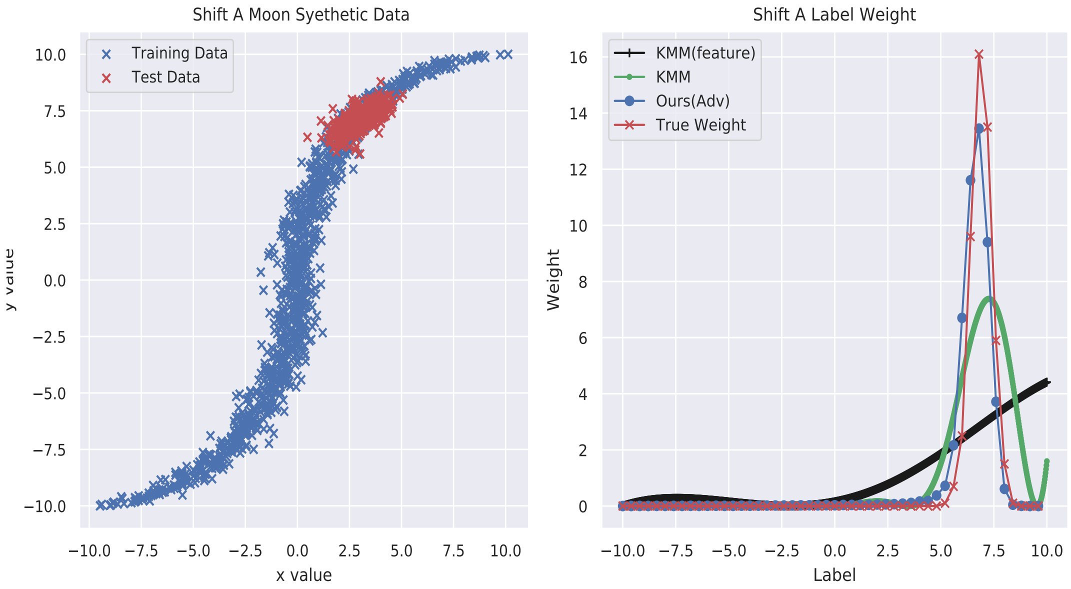


FIGURE A.9. (a) The illustration of Moon Synthetic Data (Shift A, 3rd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

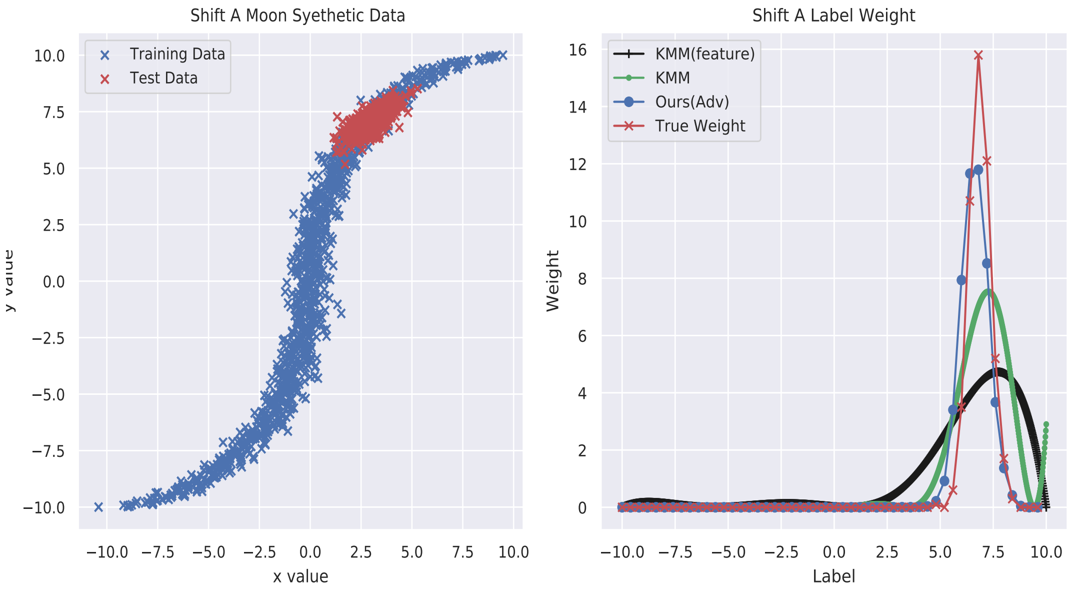


FIGURE A.10. (a) The illustration of Moon Synthetic Data (Shift A, 4th experiment), (b) The visualization of label weight P_Y^T / P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

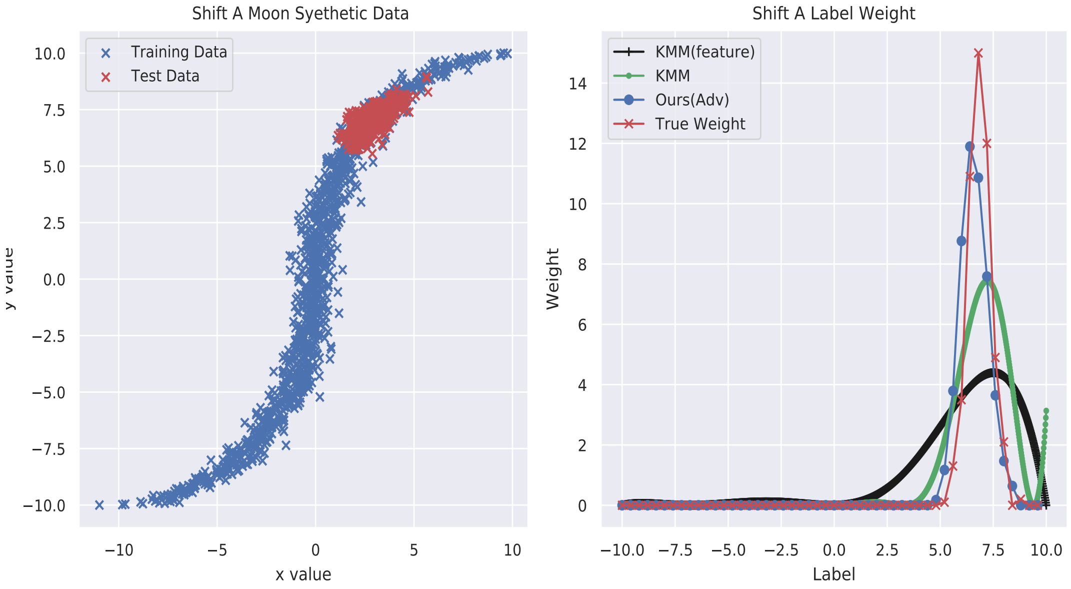


FIGURE A.11. (a) The illustration of Moon Synthetic Data (Shift A, 5th experiment), (b) The visualization of label weight P_Y^T / P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

A0.3.2 Results of Shift B

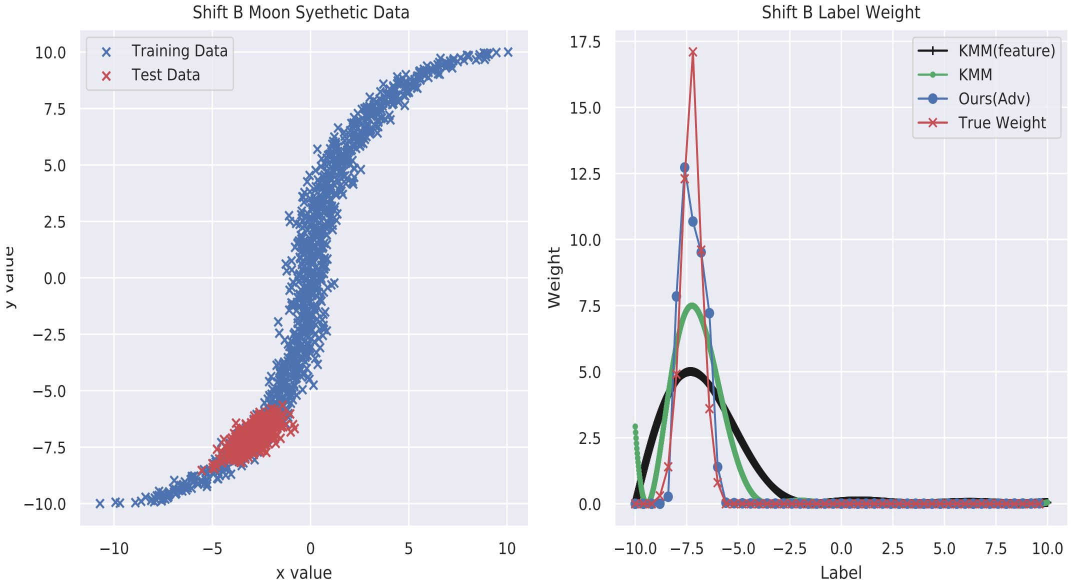


FIGURE A.12. (a) The illustration of Moon Synthetic Data (Shift B, 1st experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

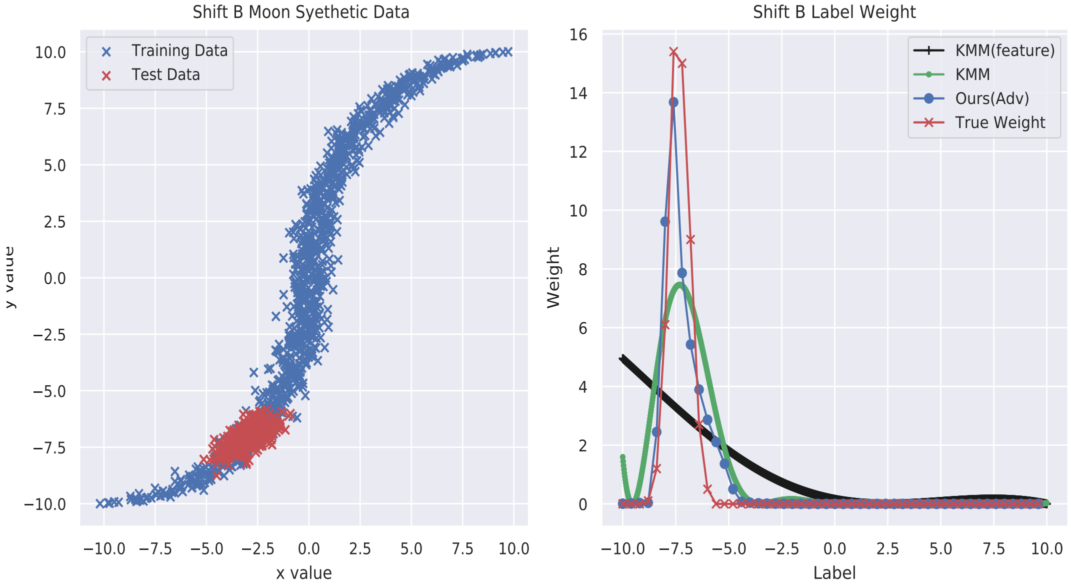


FIGURE A.13. (a) The illustration of Moon Synthetic Data (Shift B, 2nd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

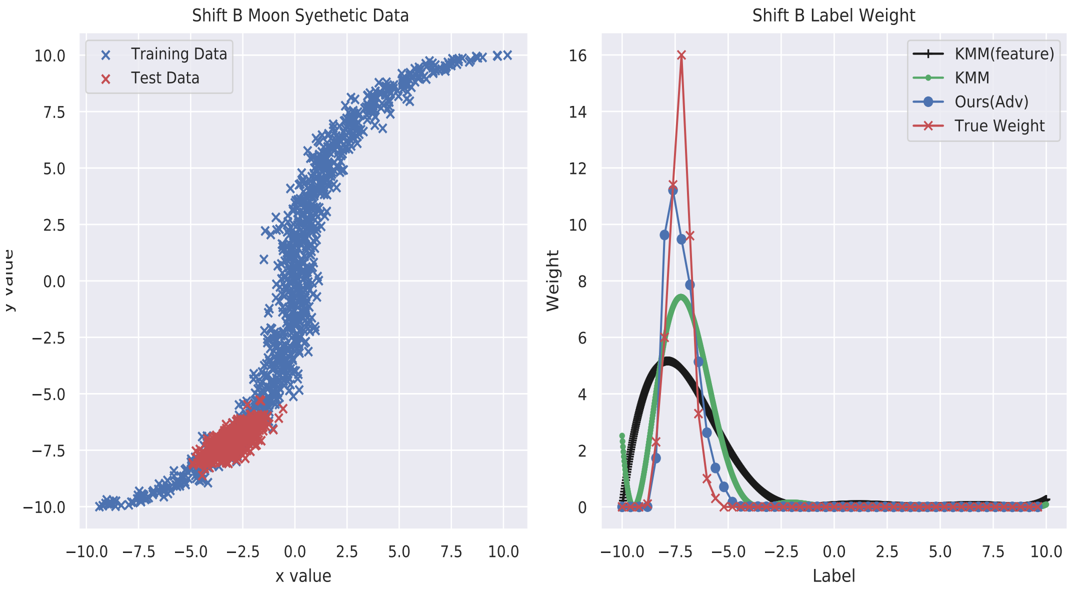


FIGURE A.14. (a) The illustration of Moon Synthetic Data (Shift B, 3rd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

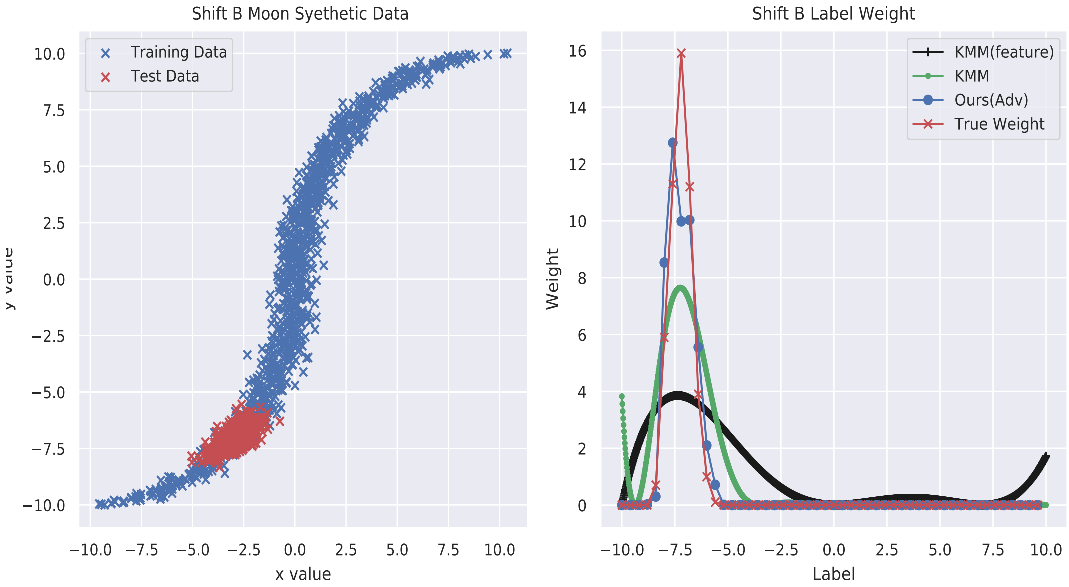


FIGURE A.15. (a) The illustration of Moon Synthetic Data (Shift B, 4th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

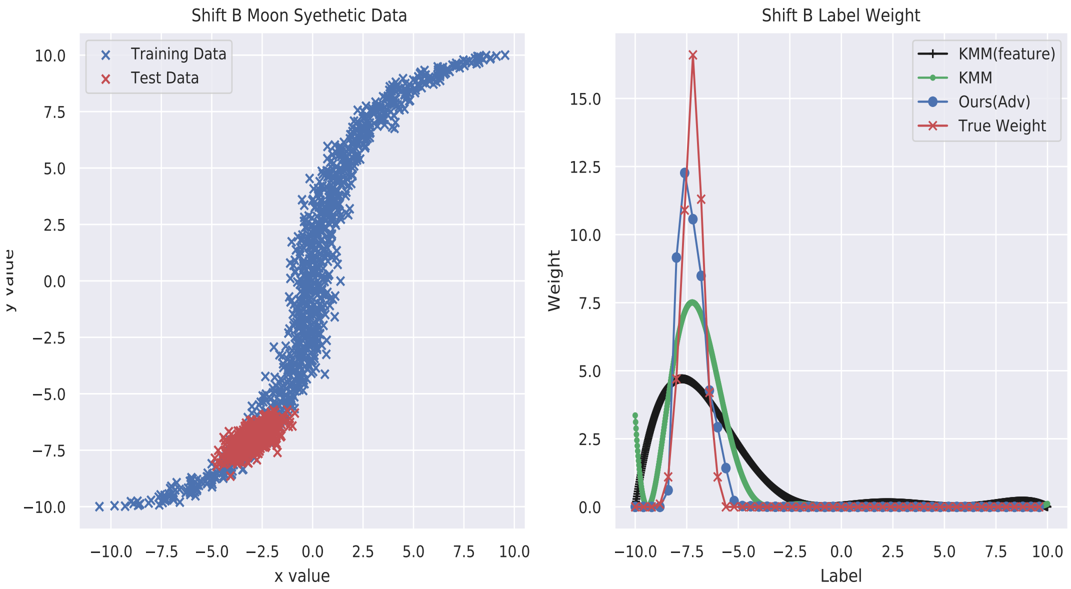


FIGURE A.16. (a) The illustration of Moon Synthetic Data (Shift B, 5th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

A0.3.3 Results of Shift C

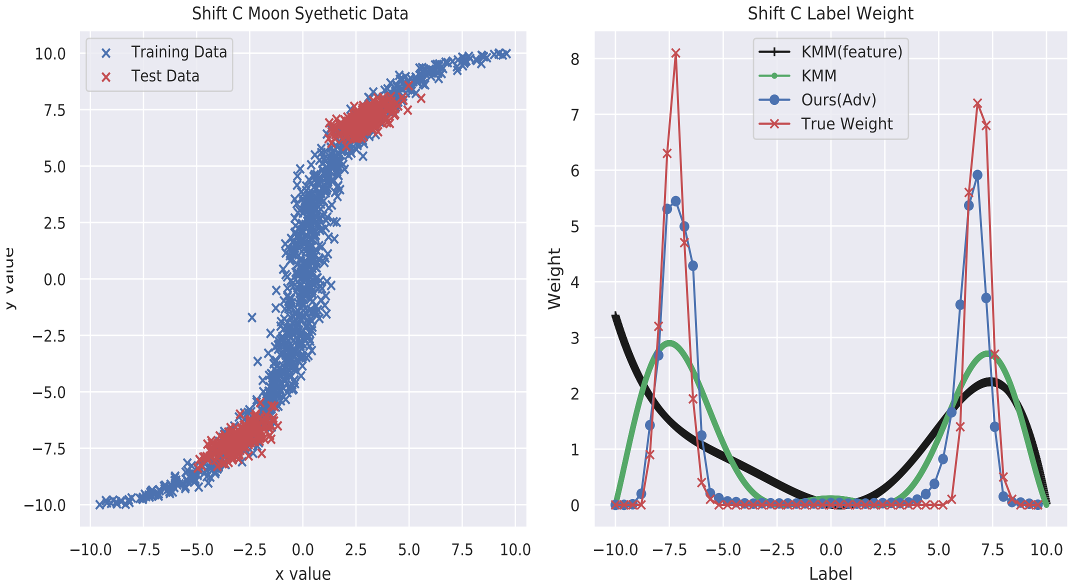


FIGURE A.17. (a) The illustration of Moon Synthetic Data (Shift C, 1st experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

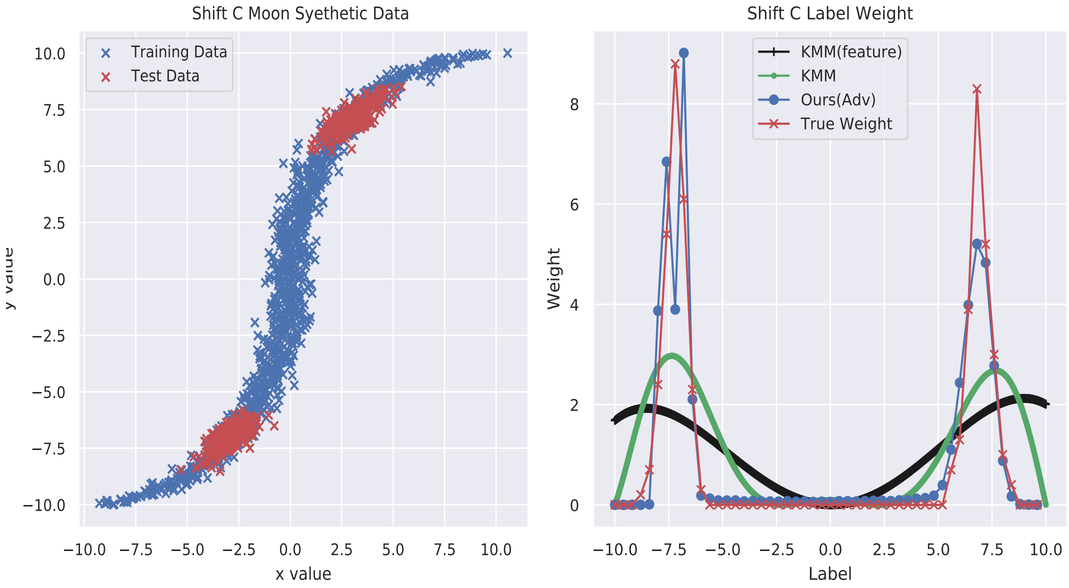


FIGURE A.18. (a) The illustration of Moon Synthetic Data (Shift C, 2nd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

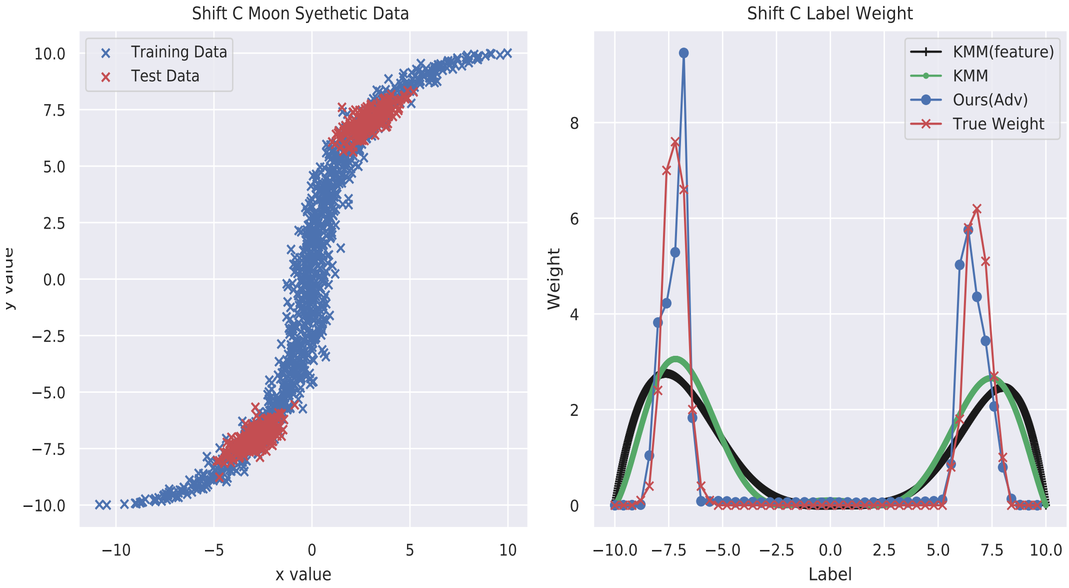


FIGURE A.19. (a) The illustration of Moon Synthetic Data (Shift C, 3rd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

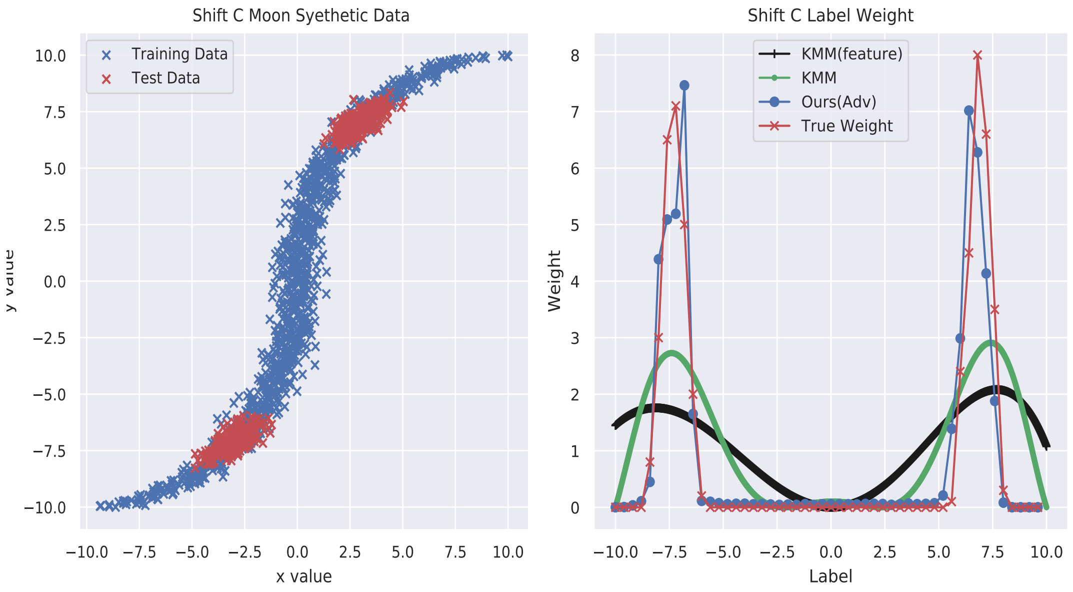


FIGURE A.20. (a) The illustration of Moon Synthetic Data (Shift C, 4th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

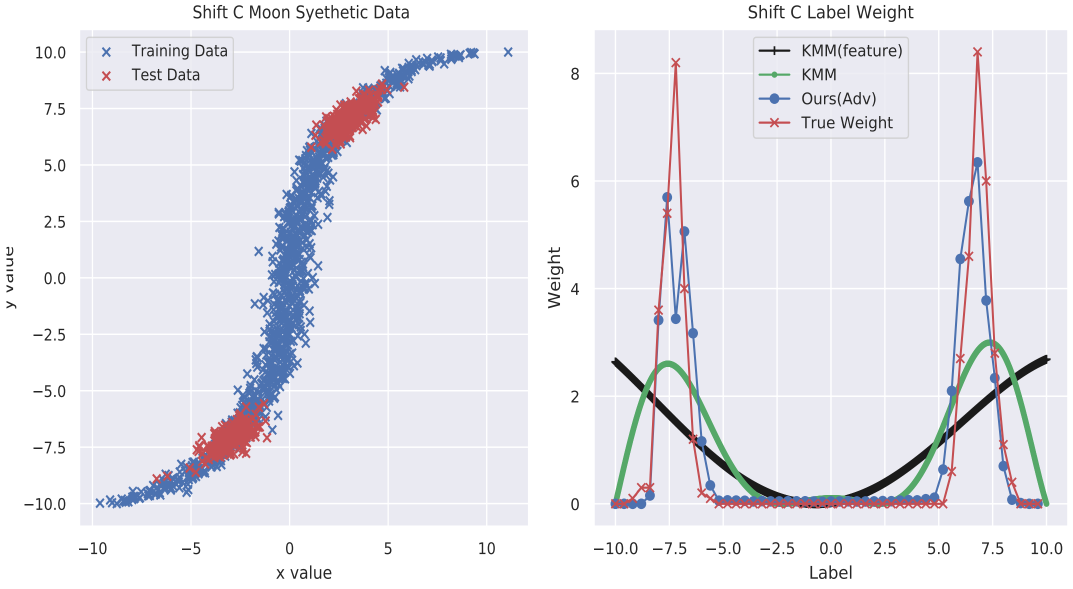
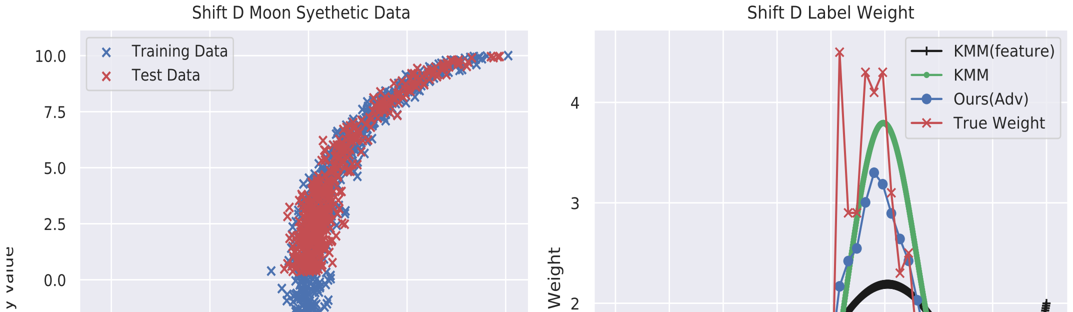


FIGURE A.21. (a) The illustration of Moon Synthetic Data (Shift C, 5th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

A0.3.4 Results of Shift D



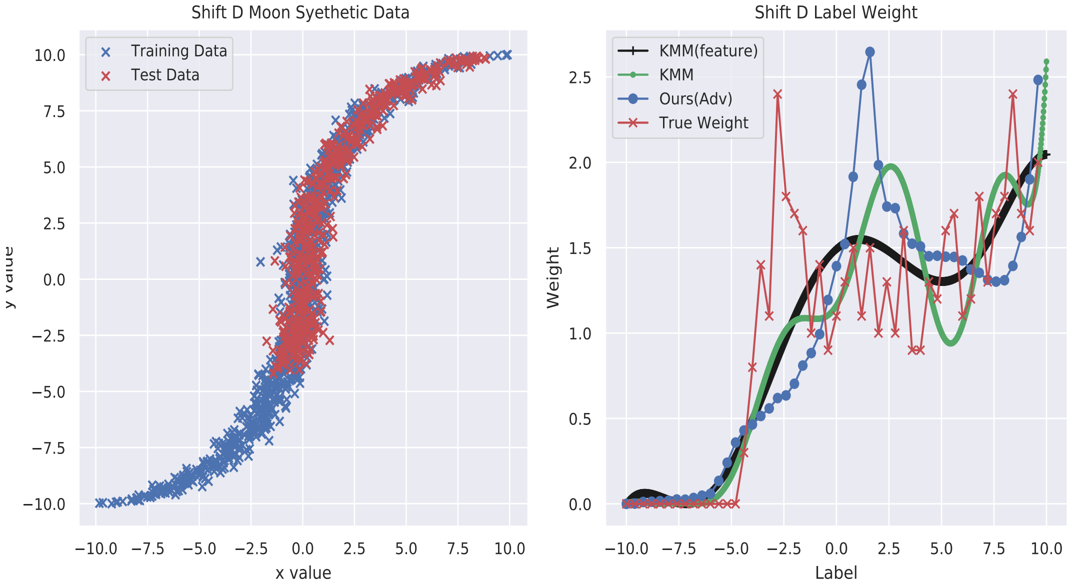


FIGURE A.23. (a) The illustration of Moon Synthetic Data (Shift D, 2nd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

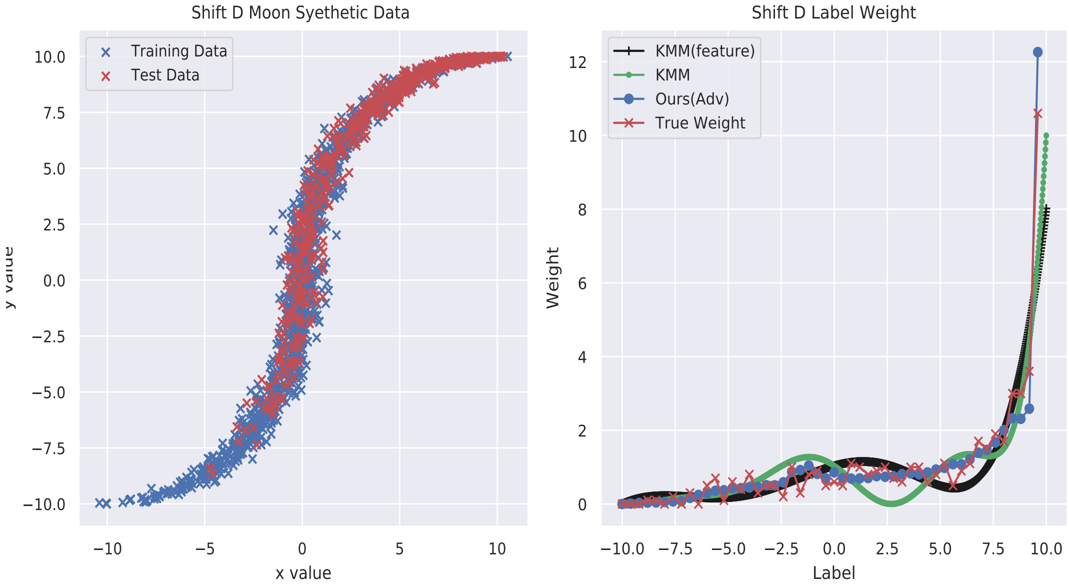


FIGURE A.24. (a) The illustration of Moon Synthetic Data (Shift D, 3rd experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

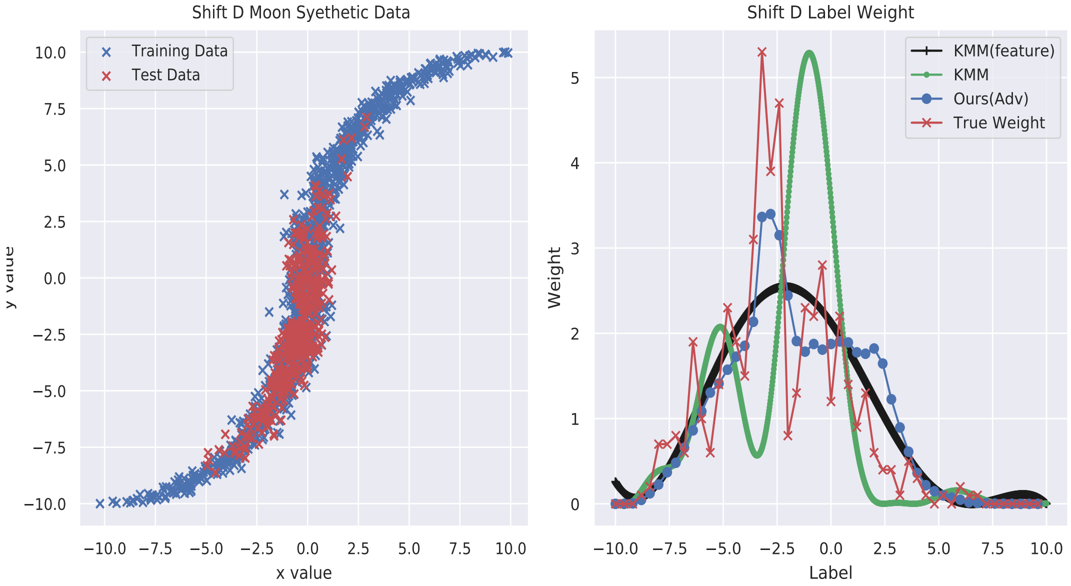


FIGURE A.25. (a) The illustration of Moon Synthetic Data (Shift D, 4th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

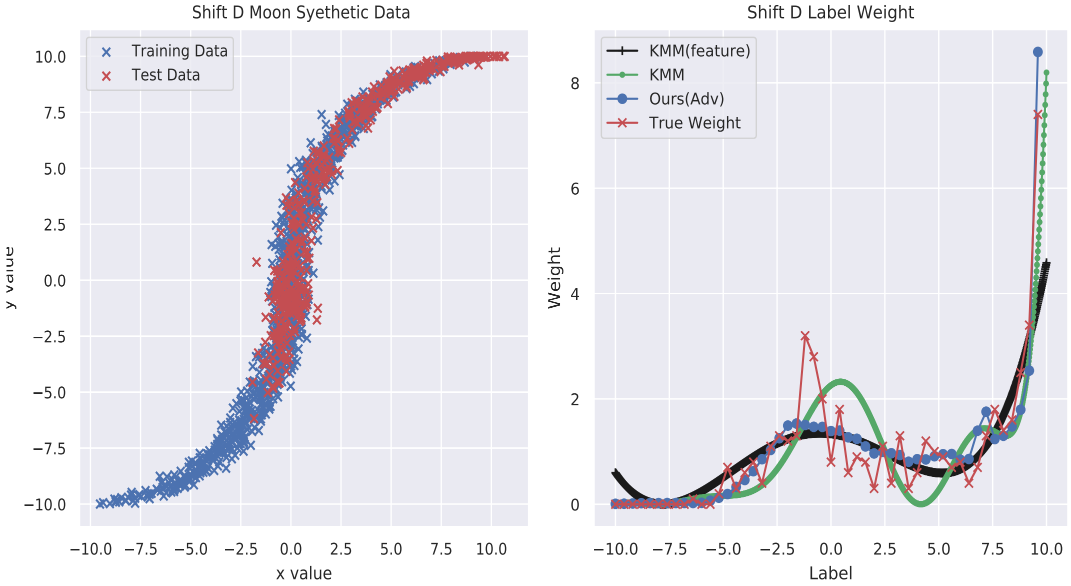


FIGURE A.26. (a) The illustration of Moon Synthetic Data (Shift D, 5th experiment), (b) The visualization of label weight P_Y^T/P_Y^S of KMM, KMM(feature), our framework and the Ground Truth.

A0.4 Results of dsprite Dataset

The details of experimental settings of dsprite could be found at the table A.5. The results of it could be found at the main paper.

Regressor Details	
Architecture	Discriminator of DCGAN
Batch Size	128
Training epochs	500
Optimizer	Adam
Learning Rate	1e-4
Label Transformation Details	
Architecture	Three-Layer Network
Label Influence Recovery Details	
Generator Architecture	Generator of DCGAN
Training Method	TAC-GAN [61]
Distribution Matching	
Optimizer	Adam
Learning Rate	1e-4
Training epochs	1000

TABLE A.5. The experimental details on dsprite dataset.

APPENDIX B

Appendix of Chapter 3

B0.1 Details of Solving α

To solve the $rSMI(V^{x_i}, V^{\hat{y}_i})$, we directly estimate the density ratio using a linear combination of kernel functions of $\{v_j^{x_i}\}_{j=1}^M \in V^{x_i}$ and $\{v_j^{\hat{y}_i}\}_{j=1}^M \in V^{\hat{y}_i}$:

$$\frac{S_i}{\beta S_i + (1-\beta)Q_i} = \omega_\alpha(v^{x_i}, v^{\hat{y}_i}) = \sum_{l=1}^{m} \alpha_l \phi_l(v^{x_i}, v^{\hat{y}_i}) = \alpha^T \phi(v^{x_i}, v^{\hat{y}_i}), \tag{B.1}$$

where $\phi \in \mathbb{R}^m$ is the kernel function, $\alpha \in \mathbb{R}^m$ is the parameter vector we need to solve, and m is the number of kernels. α is learned so that the following squared error $J(\alpha)$ [235] is minimized:

$$J(\alpha) = \mathbb{E}_{\beta S_i + (1-\beta)Q_i}[(\omega_\alpha(v^{x_i}, v^{\hat{y}_i}) - \omega^*(v^{x_i}, v^{\hat{y}_i}))^2] = \mathbb{E}_Q[(1-\beta)\omega_\alpha^2] + \mathbb{E}_S[\beta\omega_\alpha^2 - 2\omega_\alpha] + J_0,$$

where J_0 is a constant number respect to α, and therefore can be safely ignored. Thus, the optimization problem is given as:

$$\min_\alpha [\alpha^T H \alpha - 2\alpha^T h],$$

where

$$H = (1-\beta)\mathbb{E}_Q[\phi\phi^T] + \beta\mathbb{E}_S[\phi\phi^T], \qquad h = \mathbb{E}_S[\phi].$$

For computational efficiency, we define the kernel function $\phi(v^{x_i}, v^{\hat{y}_i})$ as the product of $K(v^{x_i}; k_c) \in \mathbb{R}^m$ and $L(v^{x_i}; l_c) \in \mathbb{R}^m$, which are kernel functions of v^{x_i} and $v^{\hat{y}_i}$ respectively:

$$\phi(v^{x_i}, v^{\hat{y}_i}) = K(v^{x_i}) \circ L(v^{\hat{y}_i}),$$

where $\circ$ denotes the Hadamard product. Approximating the expectations in H and h by empirical averages, and adding a quadratic regularizer $\alpha^T R\alpha$ to avoid over-fitting, the objective function in our optimize problem becomes:

$$\hat{J}(\alpha) = [\alpha^T \hat{H}\alpha - 2\hat{h}^T\alpha + \lambda\alpha^T R\alpha], \tag{B.2}$$

where R is the positive semi-definite regularization matrix, and

$$\hat{H} = \frac{1-\beta}{n}(K \circ L)(K \circ L)^T + \frac{\beta}{n^2}(KK^T) \circ (LL^T), \qquad \hat{h} = \frac{1}{n^2}(K1_n) \circ (L1_n),$$

where n is the number of samples, 1_n is the n-dimensional vector filled by ones, and K and L are two $m \times n$ matrices composed by kernel functions. The equation B.2 is a unconstrained quadratic problem, and thus could be solved by analytically and the optimal solution of $\hat{\alpha}$ is:

$$\hat{\alpha} = (\hat{H} + \lambda R)^{-1}\hat{h}.$$

B0.2 Experimental Analysis

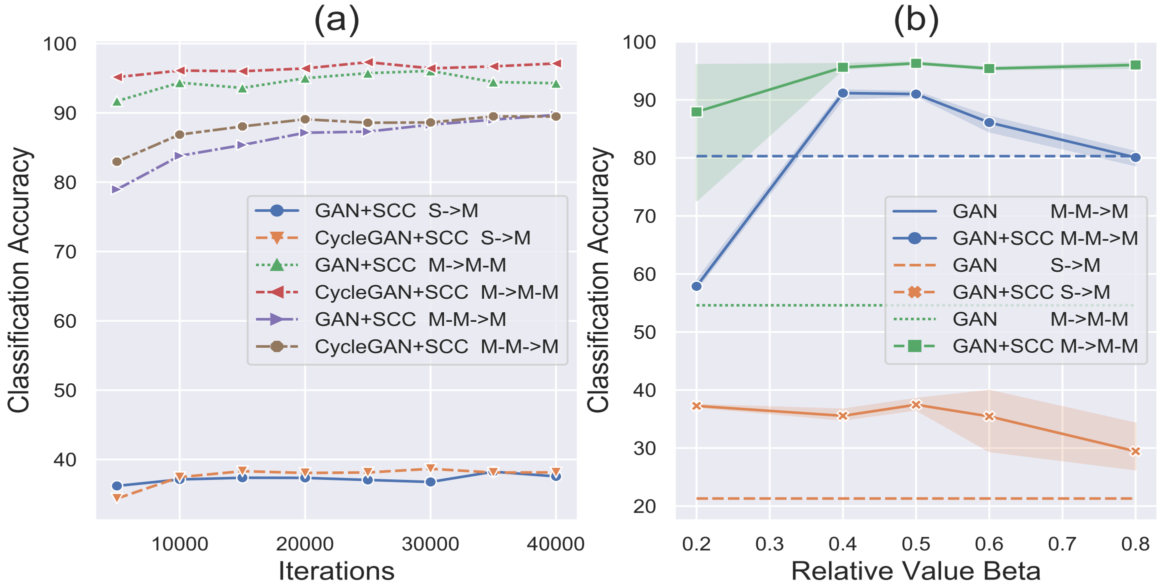


FIGURE B.1. The training curves and the sensitive analysis about β on Digits datasets

B0.2.1 β Analysis

We conduct the sensitive analysis of β on the digits datasets (each experiment is repeated 3 times) and the results are shown as Figure B.1 (b). We can see the performance of translation models are all improved with varied β, and we use 0.5 for convenience.

B0.2.2 Generation Diversity Analysis

We conduct the generation diversity experiments on the edge2shoes dataset. Following MUNIT [88], we calculate the average LPIPS distance between 1900 pairs of randomly generated images (sampled from 100 input images). MUNIT with SCC has the average LPIPS of 0.120, improving the diversity of original MUNIT model with 0.104 LPIPS score. Therefore, our SCC has no negative impact on generation diversity. Some generation examples are given as Figure B.2.

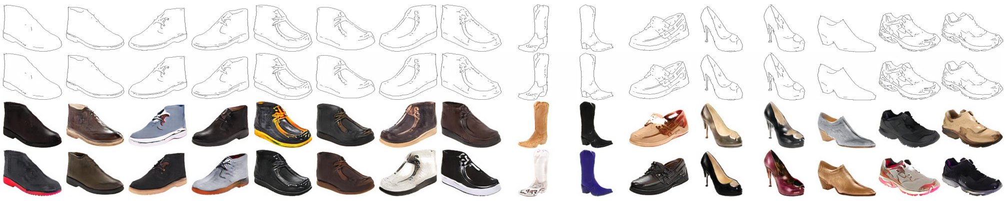


FIGURE B.2. The generation example of MUNIT+SCC on the edge2shoes. Specifically, images at first two rows are source domain images and the others are translated images by MUNIT+SCC.

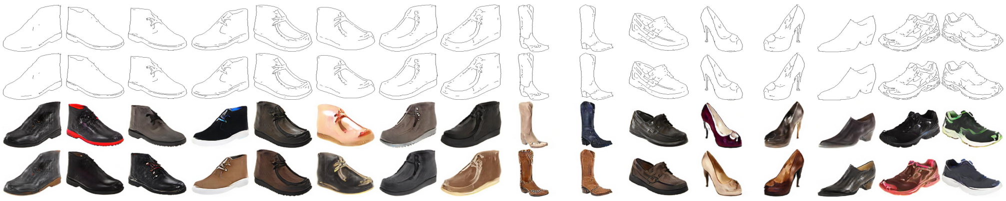


FIGURE B.3. The generation example of MUNIT on the edge2shoes. Specifically, images at first two rows are source domain images and the others are translated images by MUNIT.

B0.2.3 Stability Analysis

We conduct the training stability analysis of our SCC on the digits datasets and the results are shown as Figure B.1 (a). We can see the training procedure is stable with our SCC.

B0.3 Experiments

B0.3.1 KID scores of Qualitative evluation

Following the recent work [114], we use KID score [19] as the evaluation metric to evaluate the . The results are reported as Table B.1, and we can see that the vanilla GAN method coupled with our SCC can achieve the comparable results with those methods with larger model size. In addition, a simple generator based on res-blocks trained by the combination of cycle, geometry and our SCC constraint can achieve SOTA performance on almost all datasets.

B0.4 Experimental Details

B0.4.1 Digits

All digits images are resized to 32×32 resolution. Following [51], the network details of this experiment are given in Table B.2.

Following all settings of the original models, the learning rate for generator and discriminator is 0.0002, the training epochs is 40000 and the batch size is 64.

B0.4.2 Cityscapes

All images are resized to 128×128 resolution. Following [321, 51], the network details of this experiment are given in Table B.3.

TABLE B.1. KID scores for style transfer tasks. The results of baselines (AGGAN [249] , DRIT [133] , UNIT [148] , MUNIT [88]) are from [114]. Here U (light) is the light version of U-GAT-IT. Specifically, VGG(cosine)/VGG(L2) refer to the Contextual loss [175] and Content loss [57], respectively, and they optimize contextual and L2 distance of input and translated images' **VGG** features, respectively.

	Params	selfie2anime	horse2zebra	photo2por	anime2selfie	zebra2horse	por2photo
AGGAN	\	14.63±0.55	7.58±0.71	2.33±0.36	12.72±1.03	8.80±0.66	2.19±0.40
DRIT	65.0M	15.08±0.62	9.79±0.62	5.85±0.54	14.85±0.60	10.98±0.55	4.76±0.72
UNIT	\	14.71±0.59	10.44±0.67	**1.20±0.31**	26.32±0.92	14.93±0.75	1.42±0.24
MUNIT	46.6M	13.85±0.41	11.41±0.83	4.75±0.52	13.94±0.72	16.47±0.954	3.30±0.47
U-GAT-IT(full)	134.0M	11.61±0.57	7.06±0.8	1.79±0.34	11.52±0.57	7.47±0.71	1.69±0.53
U-GAT-IT(light)	74.0M	12.31±0.50	7.25±0.8	3.43±0.28	15.22±0.51	9.83±0.58	2.67±0.33
U (light)+SCC	74.0M	**10.37±0.32**	5.19±0.46	3.19±0.26	**10.30±0.47**	7.80±0.48	1.85±0.26
GAN+VGG(cosine)	588.1M	12.77±0.38	9.39±0.39	3.95±0.26	14.81±0.41	10.36±0.51	3.05±0.25
GAN+VGG(L2)	588.1M	11.42±0.42	6.87±0.58	1.87±0.25	12.28±0.45	9.15±0.49	1.77±0.27
GAN+VGG(L1)	588.1M	11.32±0.45	8.71±0.39	2.59±0.27	13.18±0.39	9.76±0.53	2.31±0.28
GAN + SCC	14.1M	11.37±0.41	7.28±0.52	3.86±0.39	11.61±0.40	7.15±0.46	1.88±0.25
CycleGAN	28.3M	13.08±0.49	8.05±0.72	1.84±0.34	11.84±0.74	8.0±0.66	1.82±0.36
Cycle + SCC	28.3M	11.66±0.41	6.59±0.49	2.91±0.22	10.83±0.44	6.77±0.52	1.62±0.15
GcGAN-rot	16.9M	11.89±0.42	7.05±0.45	2.24±0.26	13.28±0.35	7.67±0.47	1.84±0.28
GcGAN + SCC	16.9M	10.75±0.42	5.12±0.44	1.97±0.24	10.96±0.40	7.10±0.50	1.64±0.22
CUT	18.1M	12.1±0.42	8.45±0.45	2.85±0.33	12.45±0.54	8.99±0.5	2.23±0.31
CUT + SCC	18.1M	11.75±0.41	6.26±0.44	2.31±0.3	12.05±0.44	8.4±0.43	2.11±0.26
Gc+Cycle+SCC	45.2M	10.61±0.44	**4.82±0.68**	1.64±0.24	10.92±0.35	**6.28±0.52**	**1.31±0.27**

Following all settings of the original models, the learning rate for all generators and discriminators is 0.0002, the batch size is 1 and the training epochs for CUT is 400 and other models is 200.

B0.4.3 Maps

All images are resized to 256 × 256 resolution. Following [321, 51], the network details is similar to the details of Cityscape, but the generator contains 9 res-blocks for images with 256 × 256 resolution. Following all settings of the original models, the learning rate for all generators and discriminators is 0.0002, the batch size is 1 and the training epochs for CUT is 400 and other models is 200.

Table B.2. The network details of digits translation tasks, where C = Feature channel, K = Kernel size, S = Stride size, Deconv/Conv = Deconvolutional/Convolutional layer and "channels" donotes the image channels of target domain, such as 1 for MNIST, 3 for MNIST-M.

Generator				
index	Layers	C	K	S
1	Conv + LeakyReLU	64	4	2
2	Conv + LeakyReLU	128	4	2
3	Conv + LeakyReLU	128	3	1
4	Conv + LeakyReLU	128	3	1
5	Deconv + LeakyReLU	64	4	2
6	Deconv + LeakyReLU	channels	4	2
7	Tanh	-	-	-
Discriminator				
index	Layers	C	K	S
1	Conv + LeakyReLU	64	4	2
2	Conv + LeakyReLU	128	4	2
3	Conv + LeakyReLU	256	4	2
4	Conv + LeakyReLU	512	4	2
5	Conv	512	4	2

B0.4.4 Style Transfer

All settings are same with Maps B0.4.3. The details of datasets as follows:

selfie2anime This dataset is from U-GAT-IT [114], which contains 3400 training images and 100 images for test.

horse2zebra This dataset is from CycleGAN [321], whose training sets contains 1,067 horse images and 1,334 zebra images. The test set consists of 120 horse images and 140 zebra images.

portrait2photo This dataset is from DRIT [133], whose training sets contains 6,452 photo images and 1,811 portrait images. The test set consists of 751 photo images and 400 portrait images. Following all settings of the original models, the learning rate for all generators and discriminators is 0.0002 and the training epochs for CUT is 400 and other models is 200.

TABLE B.3. The network details of digits translation tasks, where C = Feature channel, K = Kernel size, S = Stride size, Deconv/Conv = Deconvolutional/Convolutional layer and ResBlk = A residual block

Generator				
index	Layers	C	K	S
1	Conv + ReLU	64	7	1
2	Conv + ReLU	128	3	2
3	Conv + ReLU	256	3	3
4-9	ResBlk + ReLU	256	3	1
10	Deconv + ReLU	128	3	2
11	Deconv + ReLU	64	3	2
12	Conv	3	7	1
13	Tanh	-	-	-
Discriminator				
index	Layers	C	K	S
1	Conv + LeakyReLU	64	4	2
2	Conv + LeakyReLU	128	4	2
3	Conv + LeakyReLU	256	4	2
4	Conv + LeakyReLU	512	4	1
5	Conv	512	4	1

B0.5 Analysis on the Cat2Dog Dataset

To analyze the performance of our SCC on geometry-variant datasets, we incorporate our SCC constraint into CycleGAN model and train it on the cat → dog dataset. The results are shown as Figure B.4 , we can see that the trained translation model can successfully translate dog images at the top row to cat images and preserve the basic image content (*i.e.* locations of eyes, mouth, directions of faces), even if there are some changes of geometric structure. However, as images at the bottom row show, the translation model fails to translate the dog images to cat images in a meaningful way, as the mouth of dogs block the background but the mouth of cats do not, and so the translation model need to "imagine" some background area that be blocked, which needs us to propose more constraints.

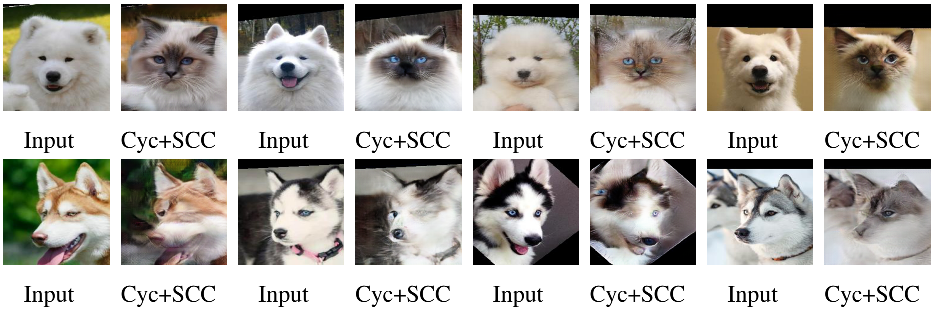
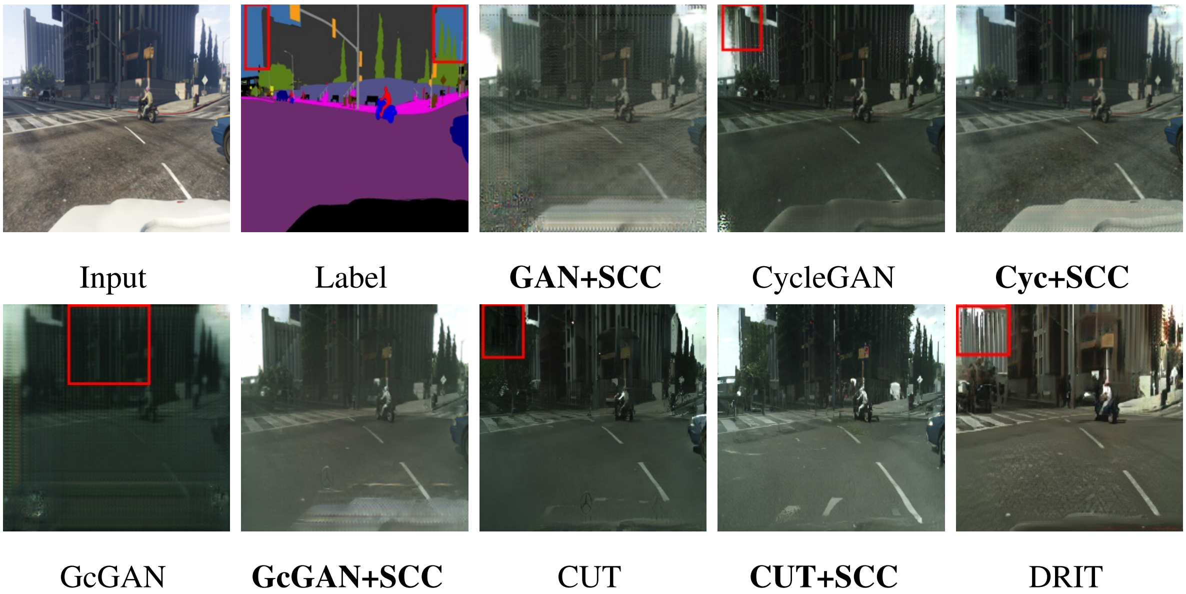
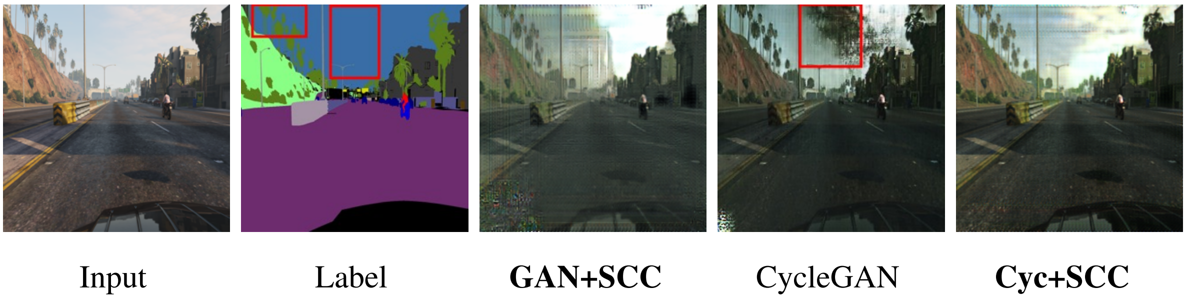


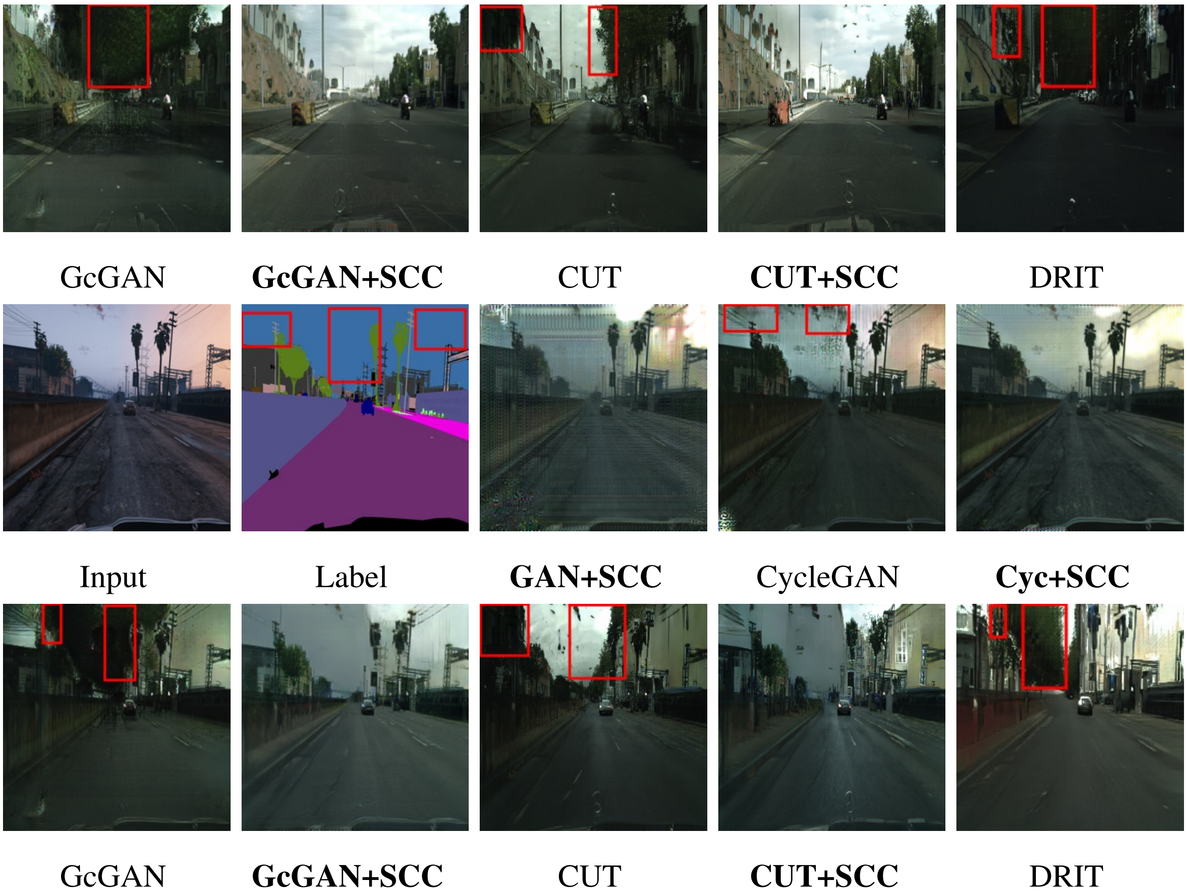
FIGURE B.4. Qualitative results on a geometry-variant dataset, including Dog → Cat. Images at the top row are successful cases, while images at the bottom row are failure cases.

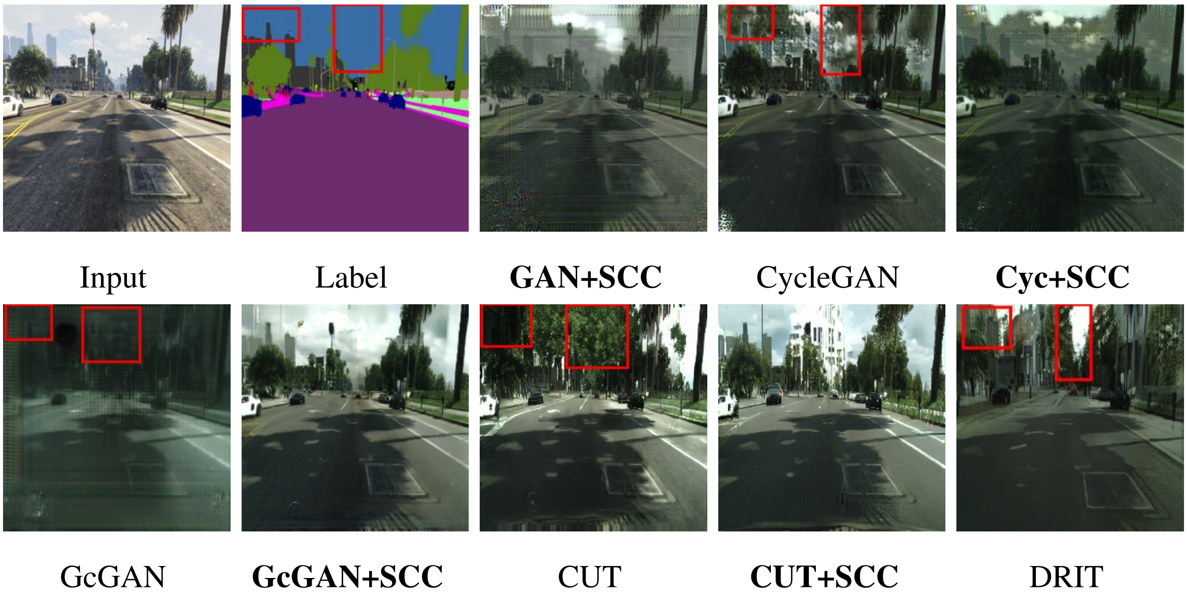
B0.6 Generated Samples

B0.7 GTA → Cityscapes






GcGAN
GcGAN+SCC
CUT
CUT+SCC
DRIT
Input
Label
GAN+SCC
CycleGAN
Cyc+SCC
GcGAN
GcGAN+SCC
CUT
CUT+SCC
DRIT


Input
Label
GAN+SCC
CycleGAN
Cyc+SCC
GcGAN
GcGAN+SCC
CUT
CUT+SCC
DRIT

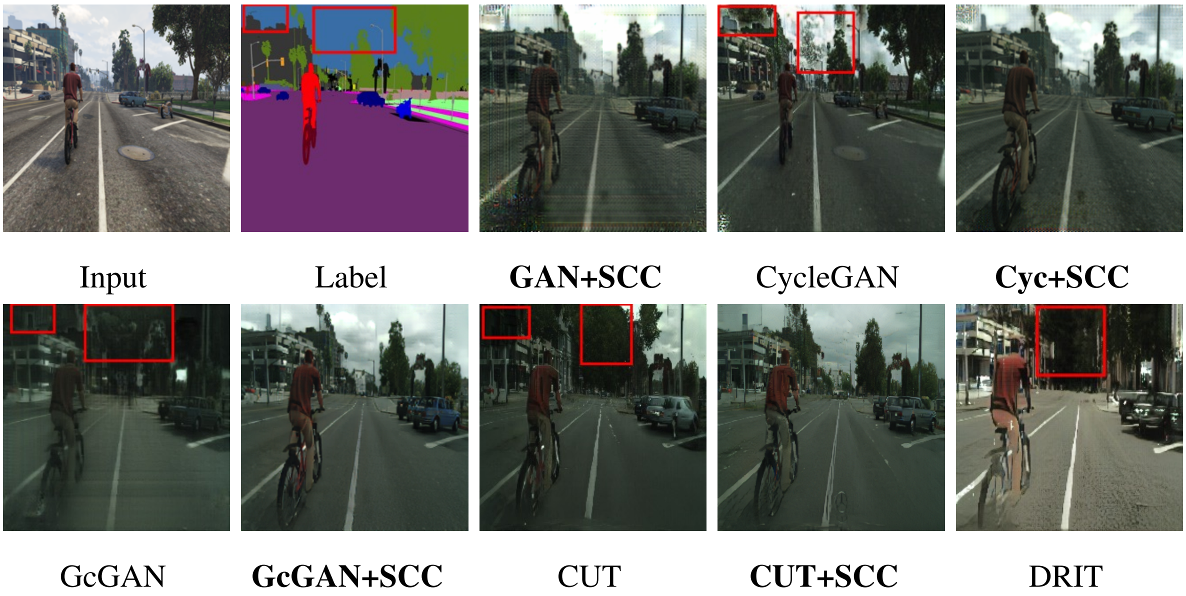
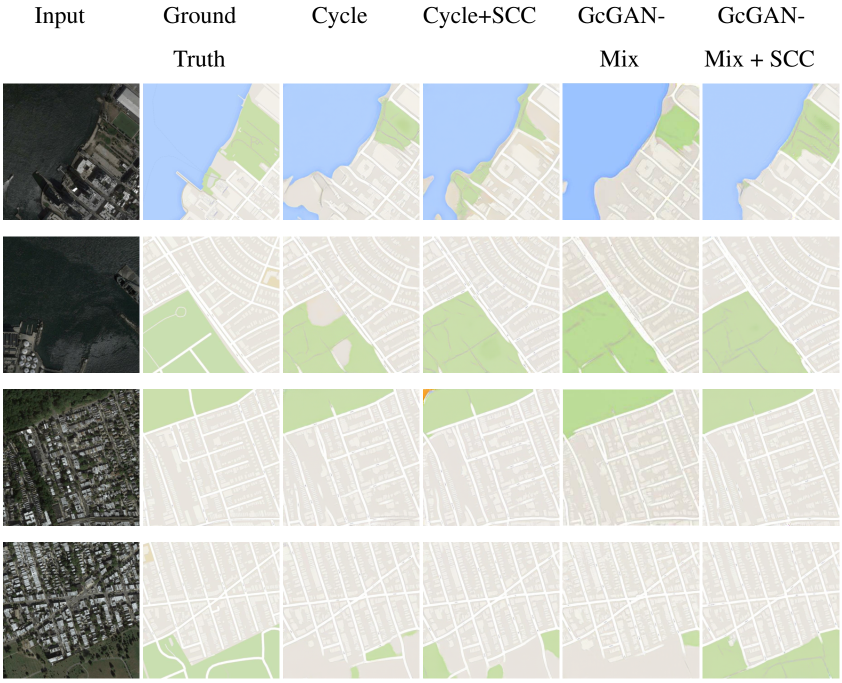


TABLE B.5. Qualitative results on GTA → Cityscapes. Obviously, the semantic information, such as sky, is better preserved by the translation model further constrained by our SCC.

B0.7.1 Maps



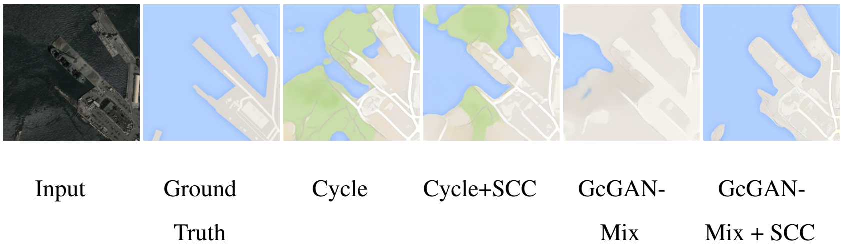


TABLE B.6. Qualitative results on the Maps dataset.

B0.7.2 Cityscapes

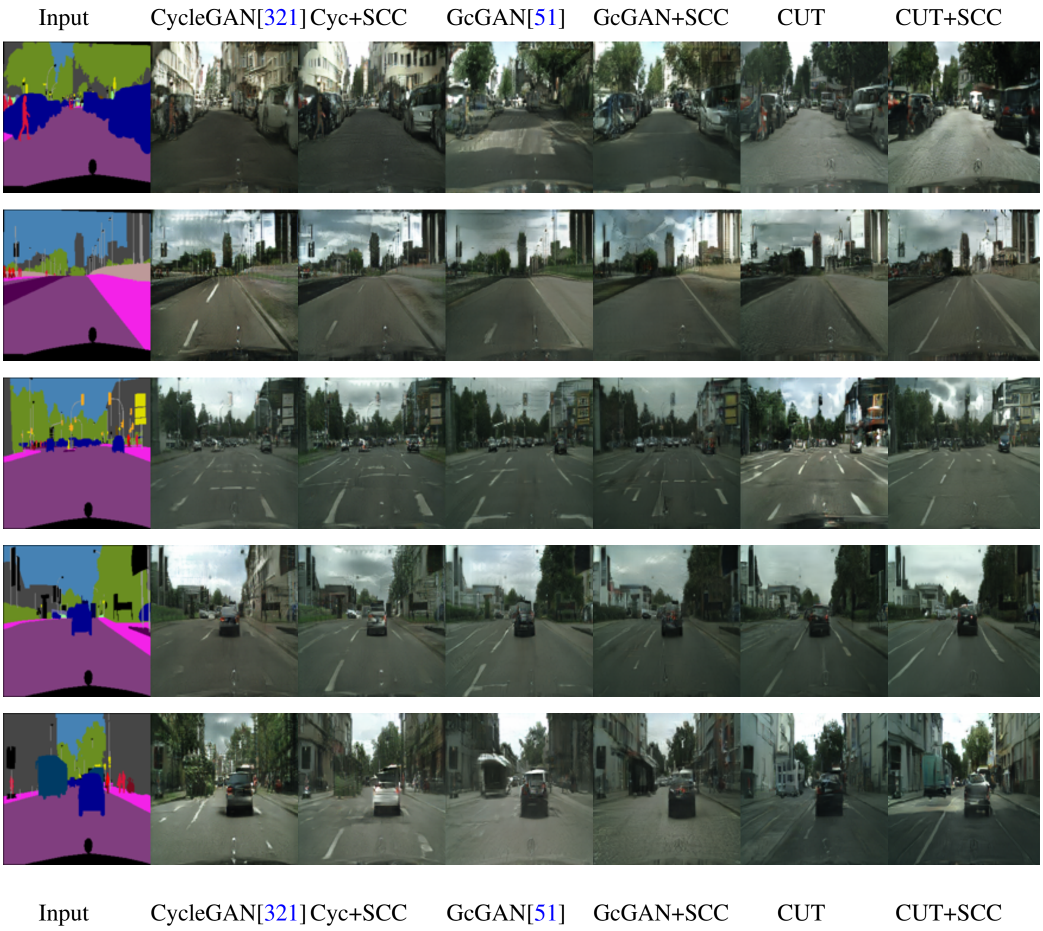
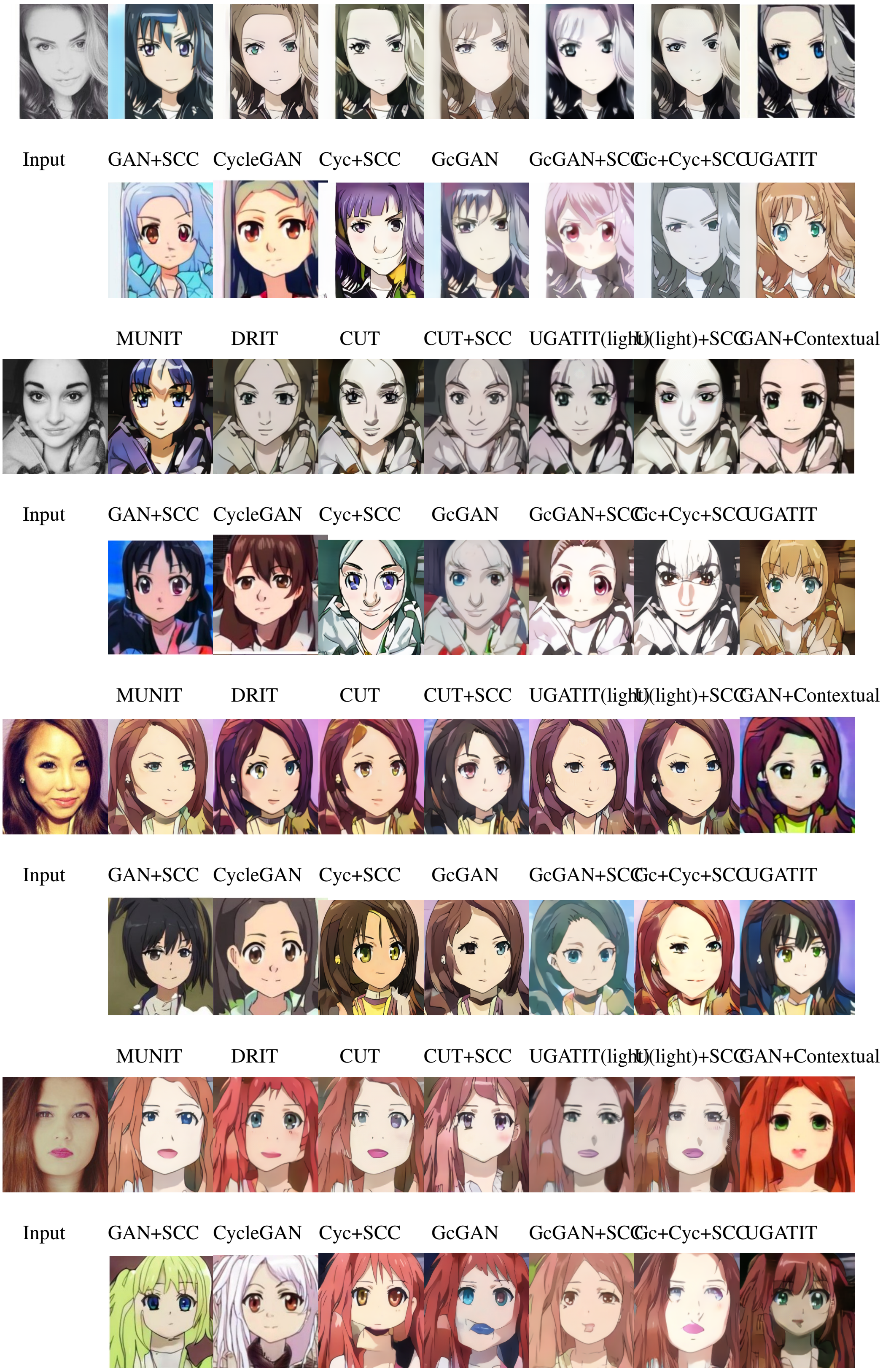


TABLE B.7. Qualitative results on the Cityscape Dataset.

B0.7.3 Qualitative Results


Input
GAN+SCC
CycleGAN
Cyc+SCC
GcGAN
GcGAN+SCC
Gc+Cyc+SCC
UGATIT
MUNIT
DRIT
CUT
CUT+SCC
UGATIT(light)
U(light)+SCC
GAN+Contextual
Input
GAN+SCC
CycleGAN
Cyc+SCC
GcGAN
GcGAN+SCC
Gc+Cyc+SCC
UGATIT
MUNIT
DRIT
CUT
CUT+SCC
UGATIT(light)
U(light)+SCC
GAN+Contextual
Input
GAN+SCC
CycleGAN
Cyc+SCC
GcGAN
GcGAN+SCC
Gc+Cyc+SCC
UGATIT
MUNIT
DRIT
CUT
CUT+SCC
UGATIT(light)
U(light)+SCC
GAN+Contextual
Input
GAN+SCC
CycleGAN
Cyc+SCC
GcGAN
GcGAN+SCC
Gc+Cyc+SCC
UGATIT

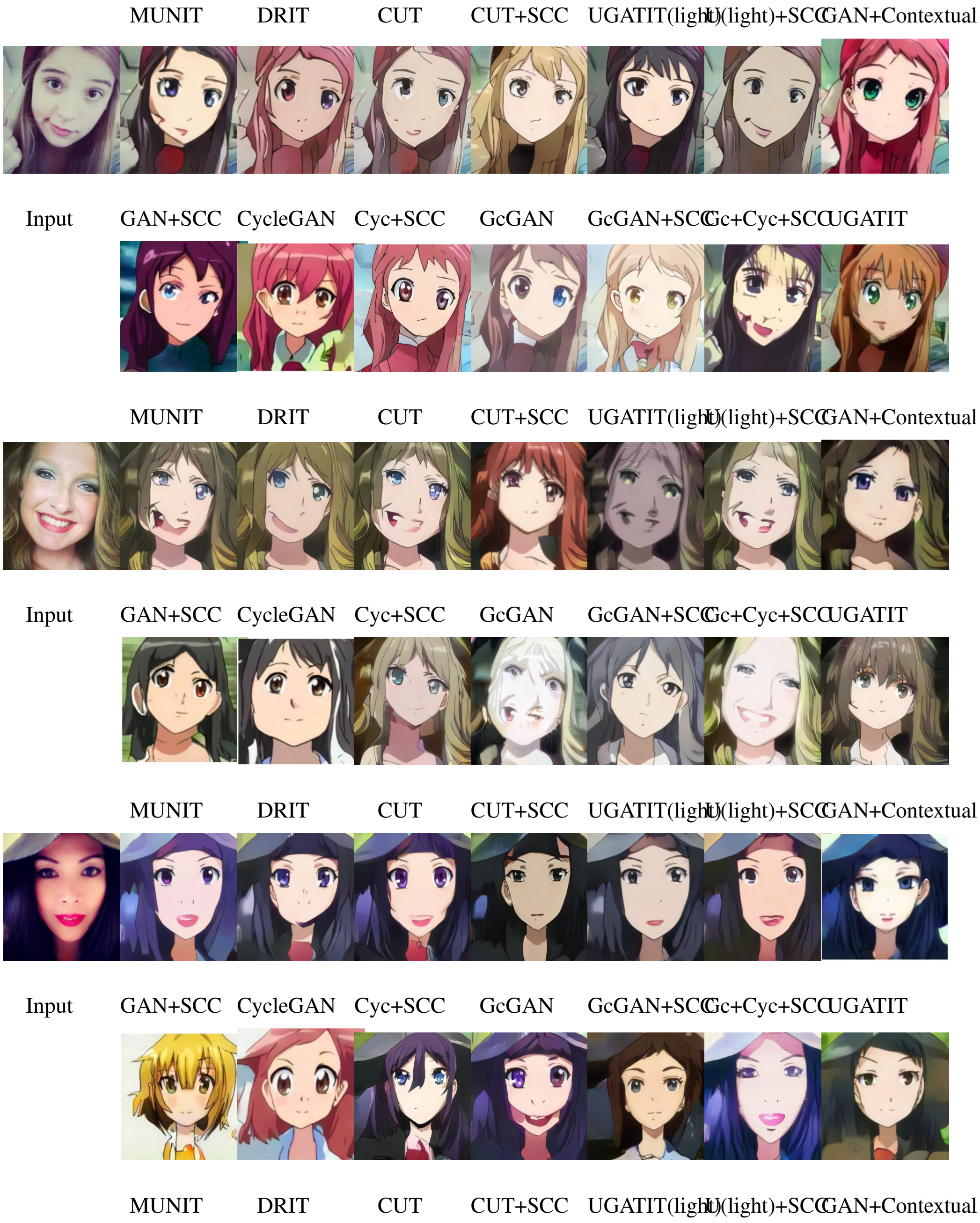
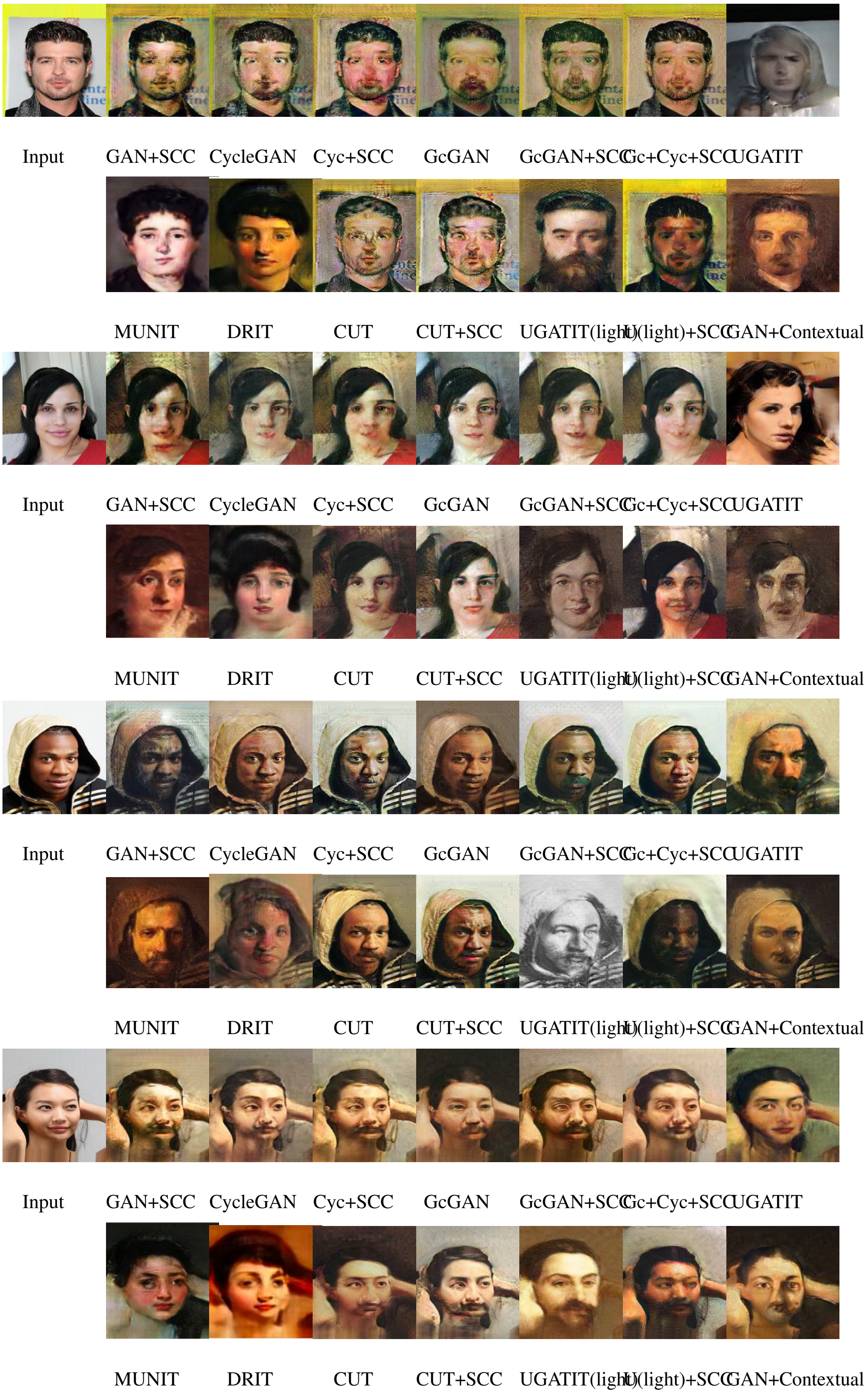


TABLE B.8. Qualitative results on Selfie → Anime. Obviously, the geometry structure, such as face shape, is better preserved by the translation model further constrained by our SCC.


Input
GAN+SCC
CycleGAN
Cyc+SCC
GcGAN
GcGAN+SCC
Gc+Cyc+SCC
UGATIT
MUNIT
DRIT
CUT
CUT+SCC
UGATIT(light)
U(light)+SCC
GAN+Contextual
Input
GAN+SCC
CycleGAN
Cyc+SCC
GcGAN
GcGAN+SCC
Gc+Cyc+SCC
UGATIT
MUNIT
DRIT
CUT
CUT+SCC
UGATIT(light)
U(light)+SCC
GAN+Contextual
Input
GAN+SCC
CycleGAN
Cyc+SCC
GcGAN
GcGAN+SCC
Gc+Cyc+SCC
UGATIT
MUNIT
DRIT
CUT
CUT+SCC
UGATIT(light)
U(light)+SCC
GAN+Contextual
Input
GAN+SCC
CycleGAN
Cyc+SCC
GcGAN
GcGAN+SCC
Gc+Cyc+SCC
UGATIT
MUNIT
DRIT
CUT
CUT+SCC
UGATIT(light)
U(light)+SCC
GAN+Contextual

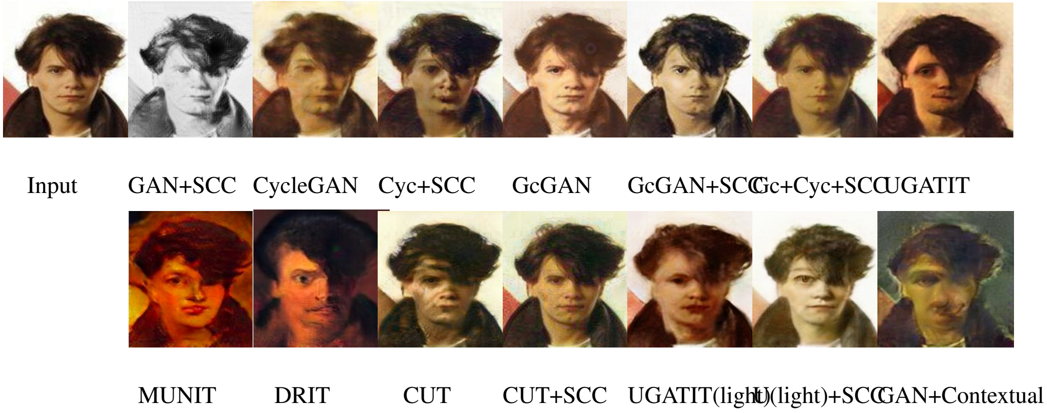
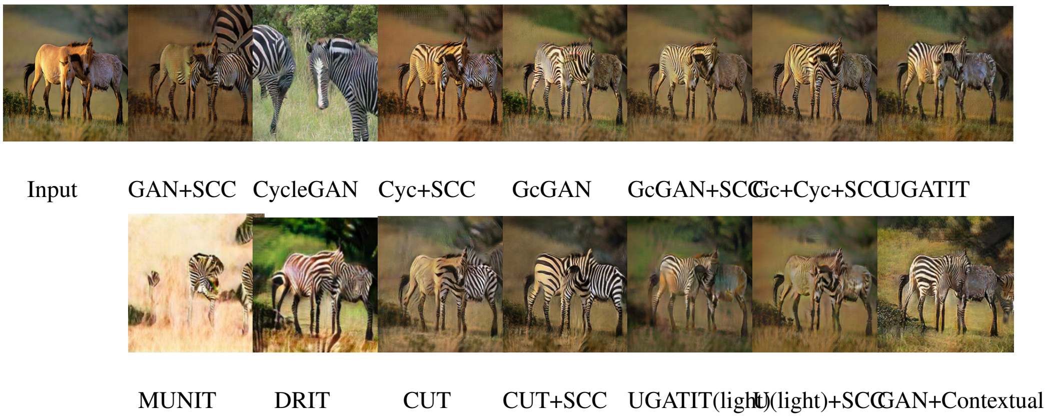
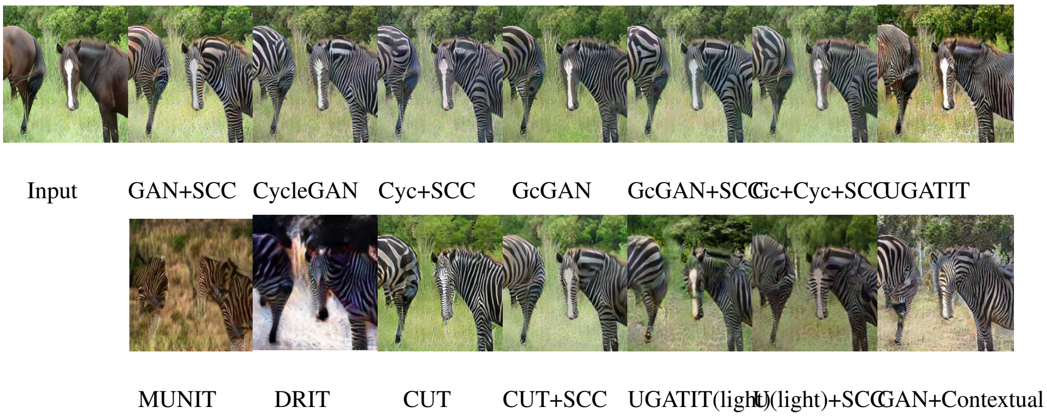


TABLE B.9. Qualitative results on photo → portrait. Obviously, the semantic information, such as face shape, is better preserved by the translation model further constrained by our SCC.





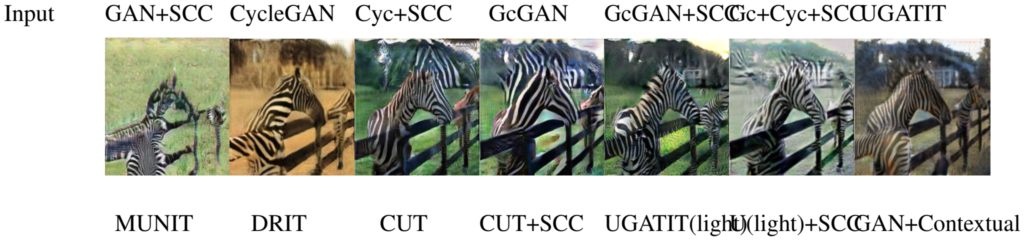


Table B.10. Qualitative results on Horse → Zebra. Obviously, the semantic information, such as horse shape, is better preserved by the translation model further constrained by our SCC.

B0.7.4 Digits

GAN

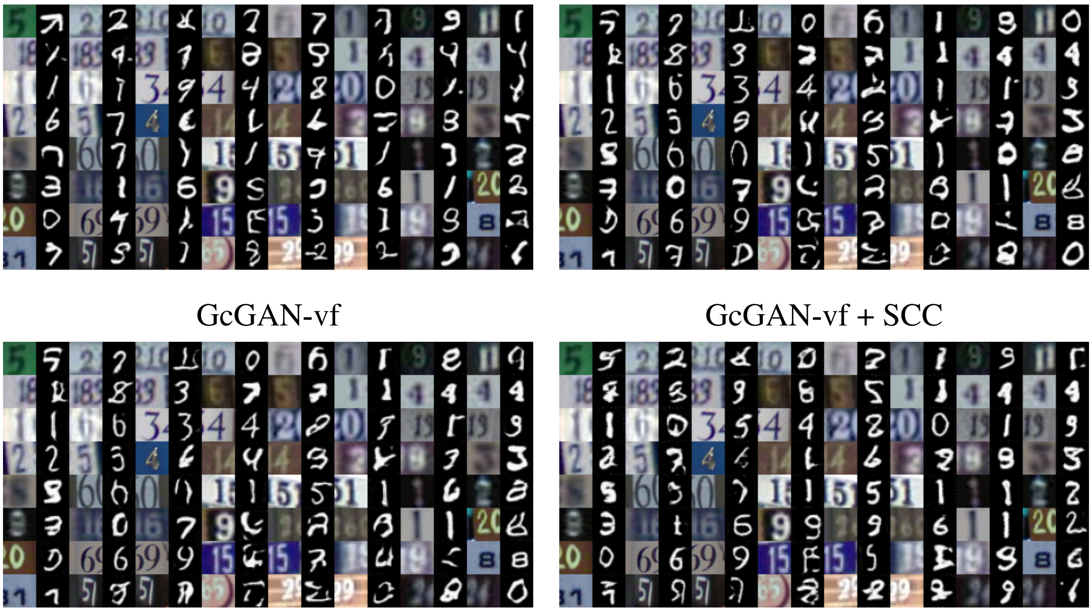
GAN + SCC

CycleGAN

CycleGAN + SCC

GcGAN-rot

GcGAN-rot + SCC



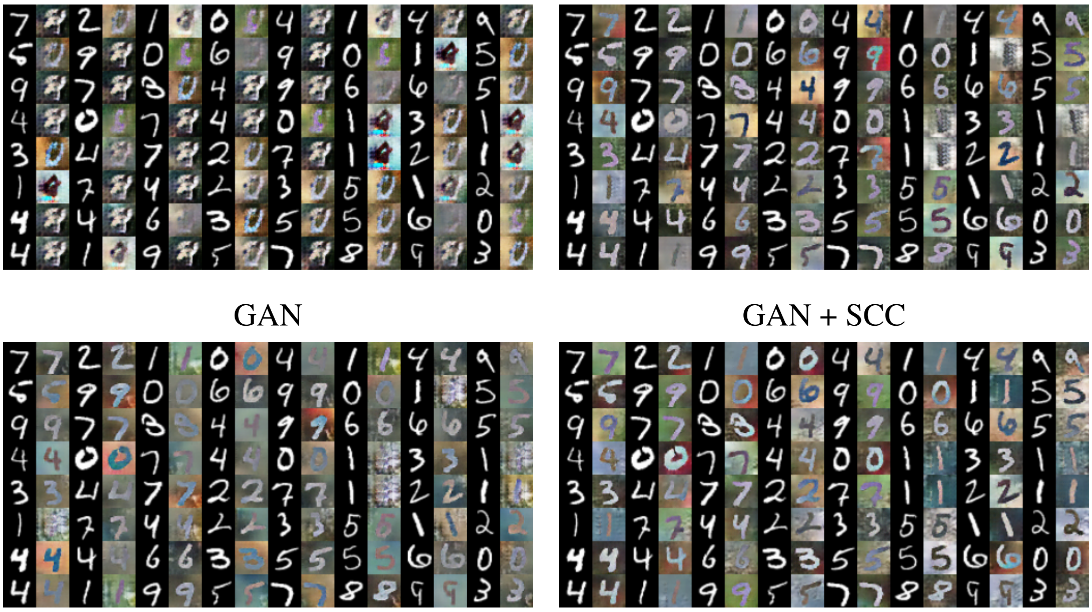
GcGAN-vf

GcGAN-vf + SCC

Gc-rot+Cycle+SCC

Gc-vf+Cycle

TABLE B.11. Qualitative comparisons on SVHN→MNIST.

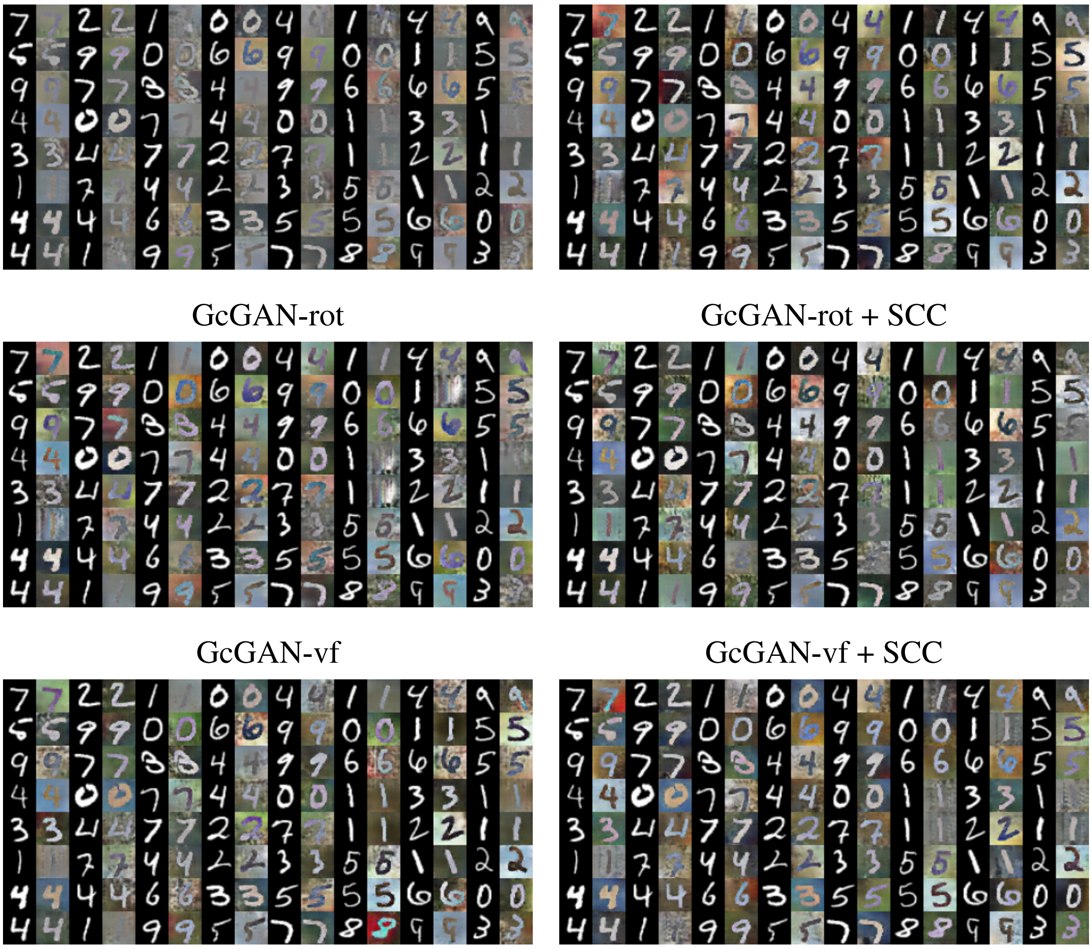


GAN

GAN + SCC

CycleGAN

CycleGAN + SCC



GcGAN-rot

GcGAN-rot + SCC

GcGAN-vf

GcGAN-vf + SCC

Gc-rot+Cycle+SCC

Gc-vf+Cycle

TABLE B.12. Qualitative comparisons on MNIST→MNIST-M.

B0.7.5 Ablation Study

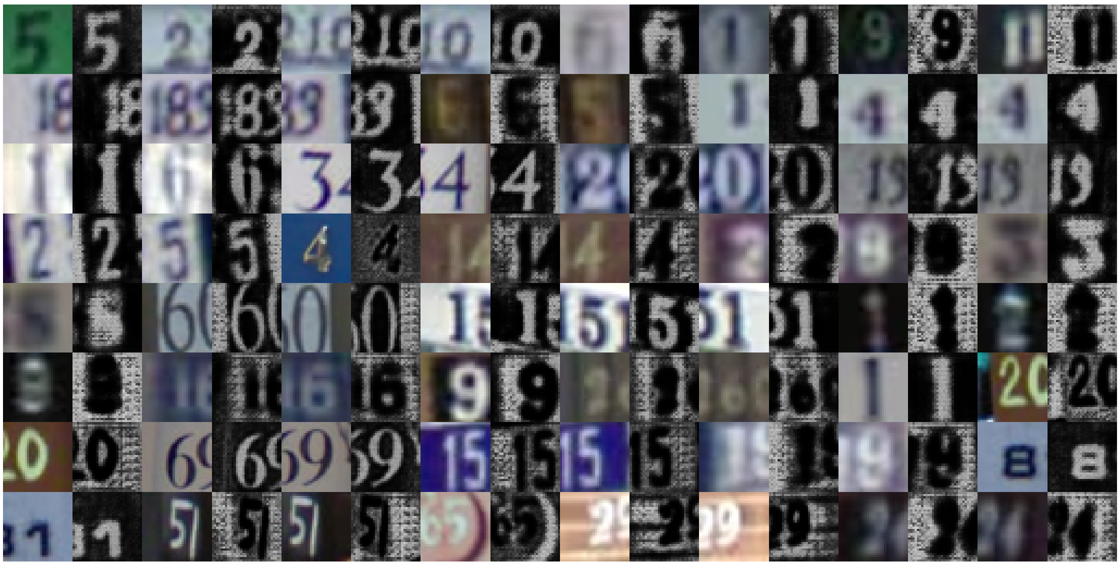


FIGURE B.5. The overlarge λ_{SCC} example on SVHN→MNIST.

An example of SVHN to MNIST translation when λ_{SCC} is set to 25 is shown as Figure B.5. The images are almost translated without any changes in geometry structures. However, the overlarge λ_{SCC} causes the translation model neglect the style information from adversarial loss, resulting in some images with opposite color. This phenomenon indicates that our SCC has good performance on the preservation of geometry structure but should be appropriate with style information.

APPENDIX C

Appendix of Chapter 4

C0.1 Environmental Settings

We follow the environmental settings of [135] in dynamics generalization. The details of settings are given as follows:

- **Pendulum** We modify the mass m and the length l of Pendulum to change its dynamics.
- **Half-Cheetah** We modify the mass of regid link m and the damping of joint d of Half-Cheetah agent to change its dynamics.
- **Crppled_Cheetah** We cripple the id of leg c of Half-Cheetah agent to change its dynamics.
- **Ant** We modify the mass of ant's leg m to change its dynamics. Specifically, we modify two legs by multiplying its original mass with m, and others two with $\frac{1}{m}$.
- **Slim_Humanoid** We modify the mass of rigid link m and the dampling of joint d of the Slim_Humanoid agent to change its dynamics.
- **Hopper** We modify the mass of m of the Hopper agent to change its dynamics.

The training and test modified parameter list can be found at the Table C.1.

C0.2 Algorithm

The training procedure is give at Algorithm 2.

TABLE C.1. The environmental settings in our paper.

	Training Parameter List	Test Parameter List	Episode Length
Pendulum	$m \in$ {0.75,0.8,0.85,0.90,0.95, 1,1.05,1.1,1.15,1.2,1.25} $l \in$ {0.75,0.8,0.85,0.90,0.95, 1,1.05,1.1,1.15,1.2,1.25}	$m \in$ {0.2,0.4,0.5,0.7, 1.3,1.5,1.6,1.8} $l \in$ {0.2,0.4,0.5,0.7, 1.3,1.5,1.6,1.8}	200
Half-Cheetah	$m \in$ {0.75,0.85,1.00,1.15,1.25} $d \in$ {0.75,0.85, 1.00,1.15,1.25}	$m \in$ {0.2,0.3,0.4,0.5, 1.5,1.6,1.7,1.8} $d \in$ {0.2,0.3,0.4,0.5, 1.5,1.6,1.7,1.8}	1000
C_Cheetah	$c \in$ {0,1,2,3}	$c \in$ {4,5}	1000
Ant	$m \in$ {0.85,0.90,0.951.00}	$m \in$ {0.20,0.25,0.30,0.35,0.40, 0.45,0.50,0.55,0.60}	1000
Slim_Humanoid	$m \in$ {0.80,0.90,1.00,1.15,1.25} $d \in$ {0.80,0.90,1.00,1.15,1.25}	$m \in$ {0.40,0.50,0.60,0.70, 1.50,1.60,1.70,1.80} $d \in$ {0.40,0.50,0.60,0.70, 1.50,1.60,1.70,1.80}	1000
Hopper	$m \in$ {0.5, 0.75, 1.0, 1.25, 1.5}	$m \in$ {0.25, 0.375, 1.75, 2.0}	500

C0.3 Training Details

Similar to the [135], we train our model-based RL agents and relational context encoder for 20 epochs, and we collect 10 trajectories by a MPC controller with 30 horizon from environments at each epoch. In addition, the cross entropy method (CEM) with 200 candidate actions is chosen as the planing method. Specifically, the batch size for each experiment is 128, β is 6e-1. All module are learned by a Adam optimizer with 0.001 learning rate.

C0.4 Network Details

Similar to the [135], the relational encoder is constructed by a simple 3 hidden-layer MLP, and the output dim of environmental-specific vector $\hat{z}$ is 10. The relational head is modelled as a single FC layer. The dynamics prediction model is a 4 hidden-layer FC with 200 units.

Algorithm 1 The training algorithm process of our relational intervention approach

```
Initialize parameters of relational encoder $\phi$, dynamics prediction model $\theta$ and relational head $\varphi$
Initialize dataset $\mathcal{B} \leftarrow \emptyset$
for Each Iteration do
    sample environments $\mathcal{M}^i$ from training environments $\{\mathcal{M}_i^{tr}\}_{i=0}^K$   ▷ Collecting Data
    for $T = 1$ to TaskHorizon do
        Get the estimation of the environment-specified factor $\hat{z}_{t-k:t-1}^i = g(\tau_{t-k:t-1}^i; \phi)$
        Collect $(s_t, a_t, s_{t+1}, r_t, \tau_{t-k:t-1}^i)$ from $\mathcal{M}^i$ with dynamics prediction model $\theta$
        Update $\mathcal{B} \leftarrow \mathcal{B} \cup (s_t, a_t, s_{t+1}, r_t, \tau_{t-k:t-1}^i)$
    end for
    for Each Dynamics Training Iteration do   ▷ Update $\phi$,$\theta$ and $\varphi$
        for $k = 1$ to K do
            Sample data $\tau_{t-k:t-1}^{i,b,P}$ , $\tau_{t:M}^{i,b,K}$ and $\tau_{t-k:t-1}^{j,b,P}$ , $\tau_{t:M}^{j,b,K}$ with batch size B,from $\mathcal{B}$
            Get the estimation of the environment-specified factor $\hat{z}_{t-k:t-1}^{i,B,,P} = g(\tau_{t-k:t-1}^{i,B,P}; \phi)$ and
            $\hat{z}_{t-k:t-1}^{ij,B,,P} = g(\tau_{t-k:t-1}^{j,B,P}; \phi)$
            Estimate the probability $w$ of $\hat{z}_{t-k:t-1}^{i,B,,P}$ and $\hat{z}_{t-k:t-1}^{j,B,,P}$ belonging to the same environment.
            Compute the total loss
            $\mathcal{L}^{tot}$=$\mathcal{L}_{\phi,\theta}^{pred}(\tau_{t:M}^{i,B,,K}, \hat{z}_{t-k:t-1}^{i,B,,P}) + \mathcal{L}_{\phi,\varphi}^{i-relation}(\hat{z}_{t-k:t-1}^{i,B,,P}) + \mathcal{L}_{\phi,\theta}^{dist}(\tau_{t:M}^{i,B,K}, \hat{z}_{t-k:t-1}^{i,B,P})$
            Update $\theta$ , $\phi$ , $\varphi \leftarrow \nabla_{\theta,\phi\,\varphi} \frac{1}{B} \mathcal{L}^{tot}$
        end for
    end for
end for
```

C0.5 Connection between Relation Loss and Mutual Information

Given a pair of data $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we donote the joint distribution of X and Y are P_{XY}, and their marginal distributions are P_X and P_Y, respectively. By definition, the mutual information between X and Y is:

$$I(X;Y) = \mathbb{E}_{P_{XY}}[\log(\frac{p(x,y)}{p(x)p(y)})] \tag{C.1}$$

To estimate mutual information between X and Y, [256] proposes a probabilistic classifier method. Concretely, we can use a Bernoulli random variable C to classify one given data pair (x, y) from the joint distribution P_{XY} $(C = 1)$ or from the product of marginal distribution $P(X)P(Y)$ $(C = 0)$. Therefore, the mutual information $I(X; Y)$ between

X and Y can be rewrite as:

$$\begin{aligned} I(X;Y) &= \mathbb{E}_{P_{XY}}[\log(\frac{p(x,y)}{p(x)p(y)})] \\ &= \mathbb{E}_{P_{XY}}[\log(\frac{p(x,y|C=1)}{p(x,y|C=0)})] \\ &= \mathbb{E}_{P_{XY}}[\log(\frac{p(C=0)P(C=1|x,y)}{p(C=1)P(C=0|x,y)})] \end{aligned} \tag{C.2}$$

Obviously, $\frac{p(C=0)}{p(C=1)}$ can be approximated by the sample size, *i.e.* $\frac{n_{P_X P_Y}}{n_{P_{XY}}}$, while $\frac{P(C=1|x,y)}{P(C=0|x,y)}$ can be measured by a classifier $h(C|x,y)$, and it can be learned by our relation loss with relational head h:

$$\mathcal{L}^{relation}_{\varphi,\phi} = -\Big[C \cdot \log\ h([x,y];\varphi) + (1-C) \cdot \log\left(1 - h([x,y];\varphi)\right)\Big], \tag{C.3}$$

where $C=1$ if the given pair (x,y) is from the joint distribution P_{XY}, and $C=0$ if the given pair (x,y) is from the product of the marginal distributions $P_X P_Y$. Because $\frac{p(C=0)}{p(C=1)}$ tend to be a constant, optimizing our relation loss is actually estimating the mutual information $I(X;Y)$ between X and Y. As such, if we regard the pairs of $(\hat{z})$ from the same trajectory/environment as positive pairs, and others are negative pairs, optimizing 4.2 is actually maximizing the mutual information between $(\hat{z})$ from the same trajectory/environment, and thus preserve the trajectory/environment invariant information. If the readers are interested in the concrete bound about this method to estimate mutual information, please refer to [256].

C0.6 Fair Comparison with TMCL

Because TMCL needs an adaptation process when deploying it into the real world while our method does not. For the fair comparison and show the significance of our method over TMCL, we test the performance of TMCL with no adaptation, and show the results below:

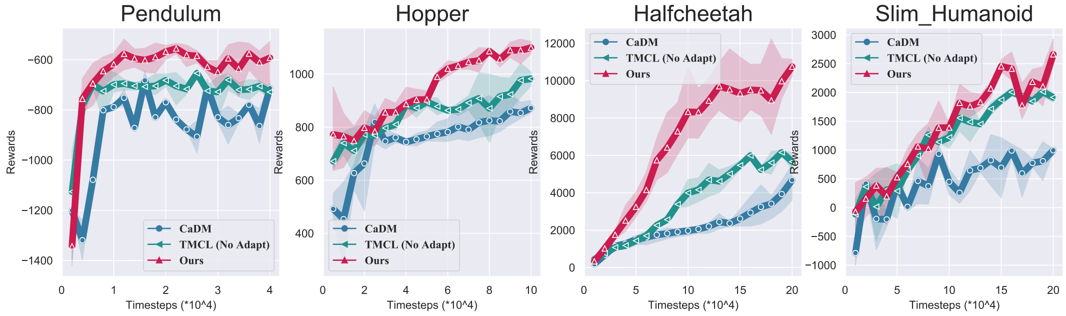


FIGURE C.1. The estimated similarities between $\hat{Z}$s (learned by the model without Intervention module) from different environments and anchors (mass=1) on different tasks, where the box represent the range of 50% samples, the line in the middle of a box denotes the average similarity and the top/bottom lines denote the max/min similarities.

We can see that the average returns of TMCL without adaptation are significantly lower than ours, especially for the classic control task Halfcheetah. The experimental comparison with TMCL without adaptation is direct evidence to support our claim: environment-separated Zs is important for the generalization of dynamics functions, and our method can significantly outperform baselines in zero-shot unseen test environments with different dynamics. Specifically, the performance of TMCL with no adaptation is still superior to the CaDM, this is because TMCL uses the invariance of Z within a trajectory (Z should predict other states within a trajectory in TMCL), which is similar to our paper with no intervention module.

C0.7 Similarities Visualiaztion

To evaluate the correctness of the estimated similarity of our intervention module, we use a $\hat{z}^i$ estimated from the environment where the mass is 1 as the anchor, and randomly sample 200 $\hat{z}^j$ estimated from different environments (including mass = 1). Then we calculate the similarity between anchor $\hat{z}^i$ and $\hat{z}^j$, and visualize the similarities according to their environments. As Figure C.2 shows, $\hat{z}^j$s belonging to the same environment with the anchor $\hat{z}^i$ have significant higher similarities than those belonging to other environments, and even higher than 0.8 in some tasks (all are higher than 0.6), which shows that our intervention module can successfully identify whether two $\hat{z}$s from the same environment or not.

To study the role of the intervention module, we also visualize the similarity of $\hat{z}^i$ learned by the model without the intervention module, and the results are given as Figure C.3. Figure C.3 shows that many contexts from different environments still have high similarities. This indicates that the existing relational learning cannot separate environment-specified factors Zs. By contrast, after incorporating the intervention module, the contexts from different environments have significantly smaller similarities than those from the same environments. The comparison between Figures C.2 and C.3 directly shows that our intervention module is valuable to predict whether two contexts are from the same environment or not.

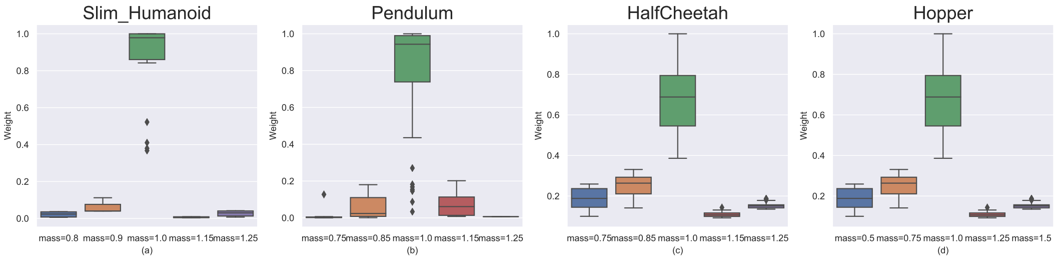


FIGURE C.2. The estimated similarities between $\hat{Z}$s (learned by the model with Intervention module) from different environments and anchors (mass=1) on different tasks, where the box represent the range of 50% samples, the line in the middle of a box denotes the average similarity and the top/bottom lines denote the max/min similarities.

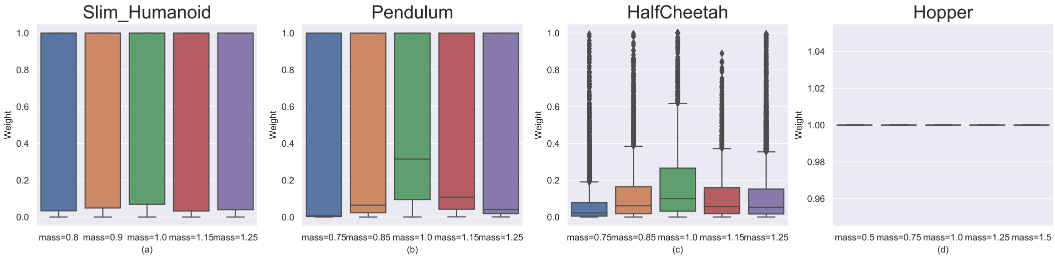


FIGURE C.3. The estimated similarities between $\hat{Z}$s (learned by the model **without** Intervention module) from different environments and anchors (mass=1) on different tasks, where the box represent the range of 50% samples, the line in the middle of a box denotes the average similarity and the top/bottom lines denote the max/min similarities.

C0.8 Prediction Errors on Traing Environments

The prediction errors of each method on training environment are given at Figure C.4.

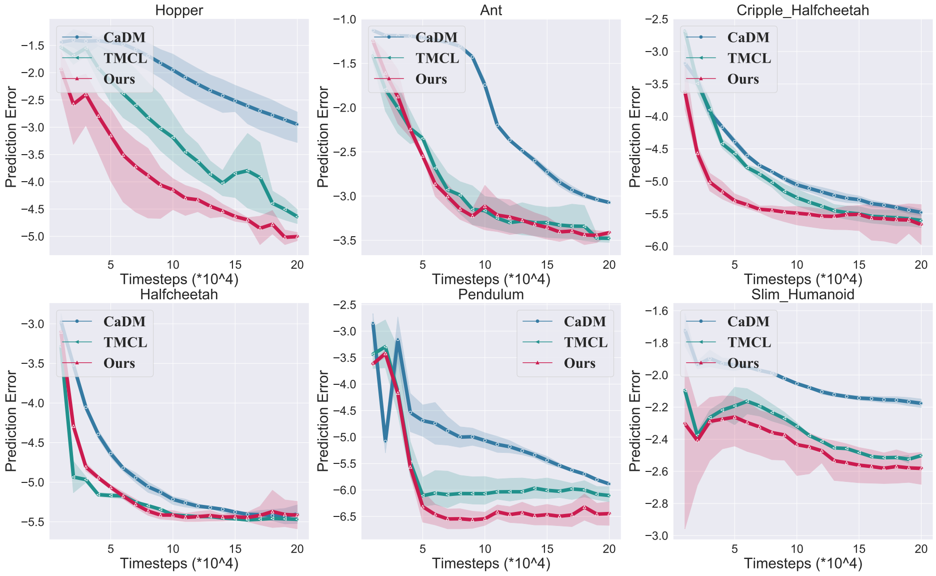


FIGURE C.4. The average prediction errors of dynamics models on training environments during training process (over three times). Specifically, the x axis is the training timesteps and y axis is the log value of average prediction prediction errors. More figures are given at Appendix C0.8.

C0.9 Prediction Errors on Test Environments

The prediction errors of each method on test environments are given at Table C.2. Specifically, we test each test environment 10 times, and plot the average prediction error to reduce random errors (Figure C.5).

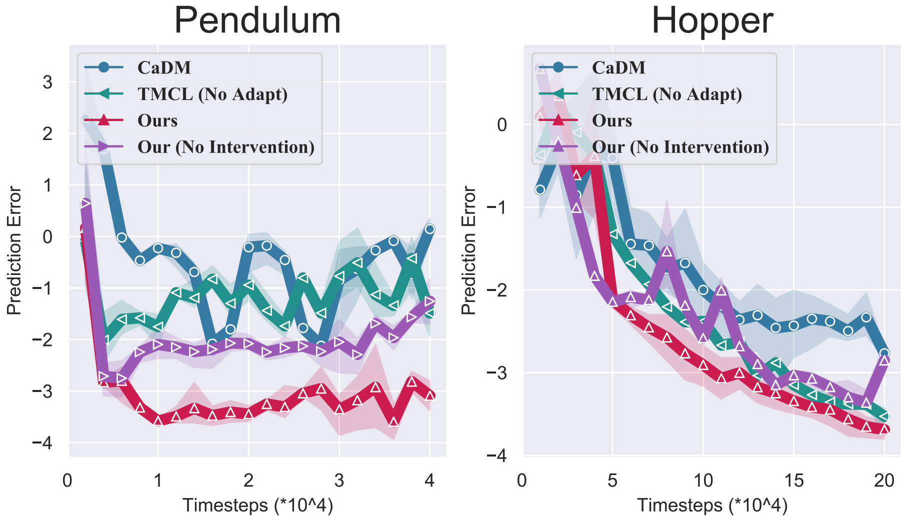


FIGURE C.5. The average prediction errors of dynamics models on test environments during training process (over three times). Specifically, the x axis is the training timesteps and y axis is the average log value of prediction prediction errors.

TABLE C.2. The prediction errors of methods on test environments

	CaDM [135]	TMCL [219]	Ours
Hopper	0.0551± 0.0236	0.0316 ± 0.0138	**0.0271** ± **0.0011**
Ant	0.3850 ± 0.0256	0.1560 ± 0.0106	**0.1381** ± **0.0047**
C_Halfcheetah	0.0815 ± 0.0029	0.0751 ±0.0123	**0.0525** ± **0.0061**
HalfCheetah	0.6151 ± 0.0251	1.0136 ± 0.6241	**0.4513** ±**0.2147**
Pendulum	0.0160 ±0.0036	0.0130± 0.0835	**0.0030** ± **0.0012**
Slim_Humanoid	0.8842 ± 0.2388	0.3243 ± 0.0027	**0.3032** ± **0.0046**

C0.10 Prediction Errors on Specified Environment

The prediction errors of each method on specified environment are given at Table C.3, C.4 and C.5.

TABLE C.3. The prediction errors of methods on specified environment of Hopper Task.

mass	CaDM [135]	TMCL [219]	Ours
0.25	0.0443 ± 0.0049	0.0294 ± 0.0131	**0.0120** ± **0.0025**
1.75	0.0459 ± 0.0006	0.0131 ± 0.0138	**0.0132** ± **0.0013**

TABLE C.4. The prediction errors of methods on specified environment of Ant Task.

mass	CaDM [135]	TMCL [219]	Ours
0.30	0.0928 ± 0.0019	0.0910 ± 0.0200	**0.0669** ± **0.0040**
0.50	0.1013 ± 0.0057	0.0887 ± 0.0212	**0.0671** ± **0.0034**

TABLE C.5. The prediction errors of methods on specified environment of Slim_Humanoid Task.

mass	CaDM [135]	TMCL [219]	Ours
0.50	0.1614 ± 0.0165	0.1860 ± 0.0040	**0.1282** ± **0.0295**
0.70	0.1512 ± 0.0152	0.1550 ± 0.0186	**0.1236** ± **0.0162**
1.50	0.1601 ± 0.0202	0.1873 ± 0.0087	**0.1444** ± **0.0233**
1.70	0.1439 ± 0.02029	0.1688 ± 0.01032	**0.1217** ± **0.0206**

C0.11 The Average Returns on Test Environments during Training Process

The average returns on test environments during training process are given at Figure C.6.

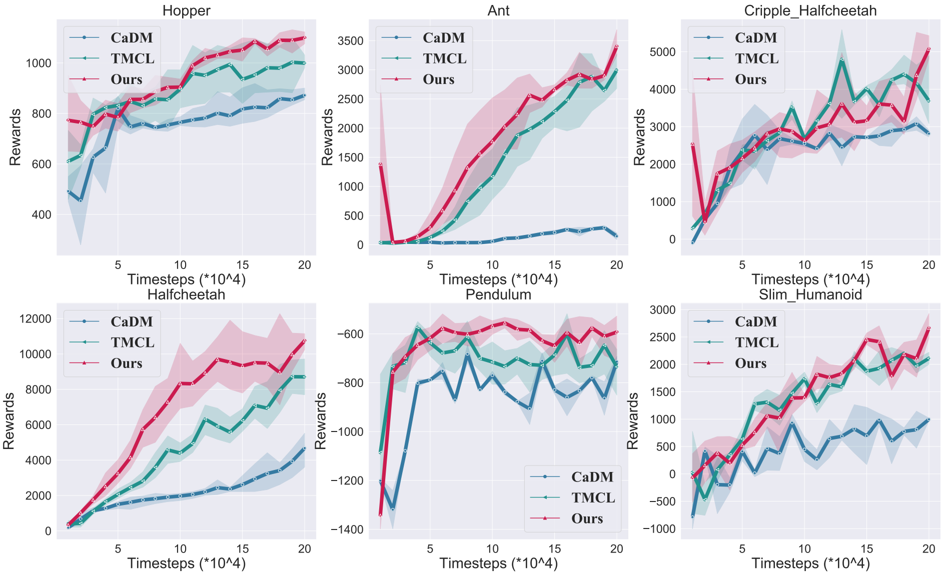


FIGURE C.6. The average rewards of trained model-based RL agents on unseen environments. The results show the mean and standard deviation of returns averaged over three runs.

C0.12 Quantitative Clustering Performance Comparison

To quantitatively evaluate the $\hat{Z}$s' clustering performance, we use K-means algorithm to predict each Z's environment id, and compare them with the true environment id. The details are provided in demo of K-means and evaluation metrics. The results are given at below. Specifically, TMCL has lower clustering performances than CaDM, but TMCL still has higher returns on test environments than CaDM. This is because TMCL clusters environments via multiplying dynamics functions rather than separating Zs.

TABLE C.6. Quantitatively clustering evaluation results of $\hat{Z}$ on Pendulum.

	homo	compl	v-meas	ARI	AMI
CaDM	1	0.655	0.627	0.516	0.599
TMCL	0	0.298	0.217	0.088	0.165
Ours (no Intervention)	0	0.768	0.762	0.760	0.653
Ours	**1**	**0.932**	**0.932**	**0.937**	**0.931**

According to the quantitative clustering performance measures, we can see that the clustering performance of our method is superior to baselines by a large margin, and the results are consistent with the performance on the test environments.

TABLE C.7. Quantitatively clustering evaluation results of $\hat{Z}$ on Half-cheetah.

	homo	compl	v-meas	ARI	AMI
CaDM	0	0.262	0.260	0.203	0.257
TMCL	0	0.239	0.165	0.051	0.126
Ours (no Intervention)	0	0.368	0.362	0.265	0.353
Ours	0	**0.416**	**0.411**	**0.312**	**0.405**

TABLE C.8. Quantitative clustering evaluation results of $\hat{Z}$ on Slim_Humanoid.

	homo	compl	v-meas	ARI	AMI
CaDM	0	0.046	0.045	0.027	0.042
TMCL	0	0.002	0.002	0.000	0.000
Ours	0	**0.055**	**0.052**	**0.037**	**0.058**

TABLE C.9. Quantitative clustering evaluation results of $\hat{Z}$ on Cripple_Halfcheetah.

	homo	compl	v-meas	ARI	AMI
CaDM	1	0.733	0.716	0.686	0.701
TMCL	0	0.253	0.000	0.000	0.000
Ours	**1**	**0.853**	**0.851**	**0.860**	**0.849**

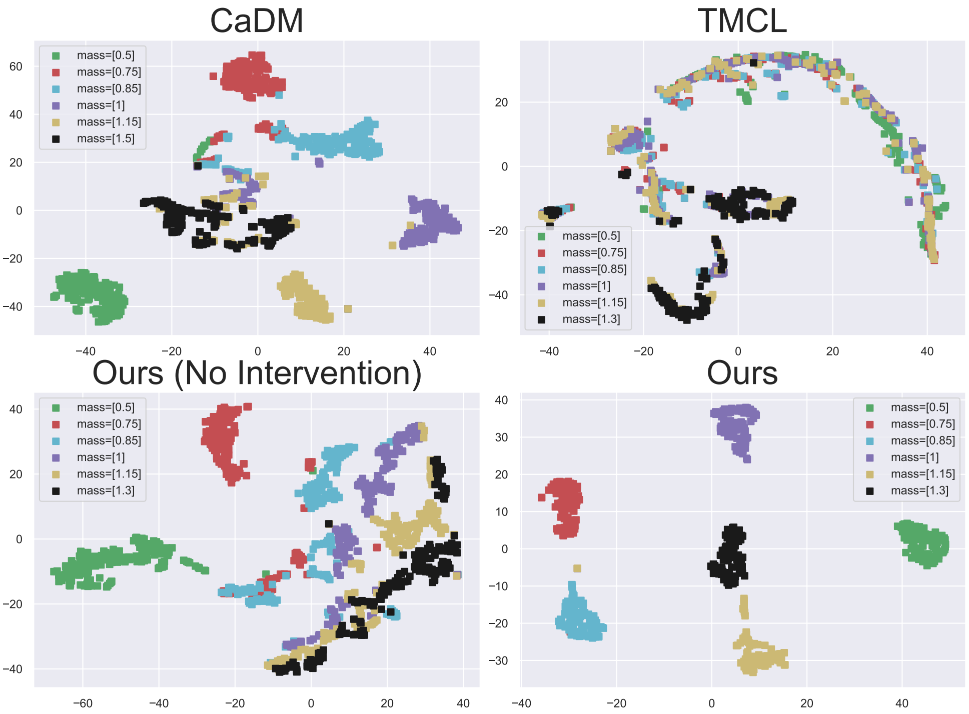
TABLE C.10. Quantitative clustering evaluation results of $\hat{Z}$ on Hopper.

	homo	compl	v-meas	ARI	AMI
CaDM	0	0.019	0.018	0.010	0.015
TMCL	0	0.023	0.008	0.000	0.003
Ours	0	**0.130**	**0.108**	**0.049**	**0.089**

C0.13 Visualization

C0.14 T-SNE Visualization

C0.15 PCA Visualization



CaDM

TMCL

Ours (No Intervention)

Ours

FIGURE C.7. The T-SNE visualization of estimated context (environmental-specific) vectors in the **Pendulum** task, where mass = 0.5 and mass =1.3 are from test environments.

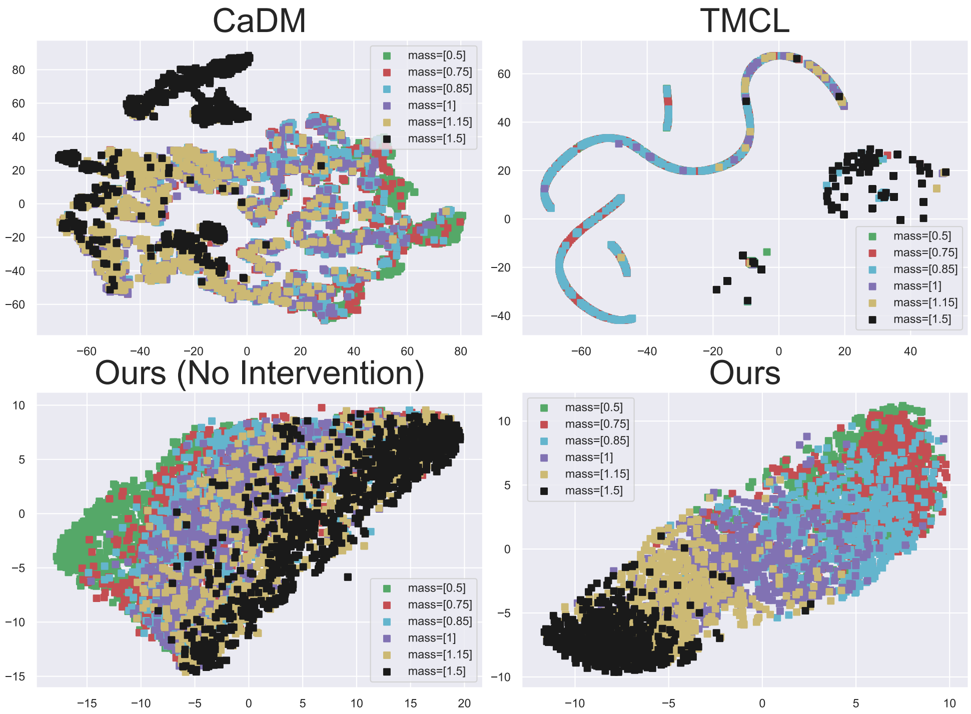


FIGURE C.8. The T-SNE visualization of estimated context (environmental-specific) vectors in the **Halfcheetah** task, where mass = 0.5 and mass =1.5 are from test environments.

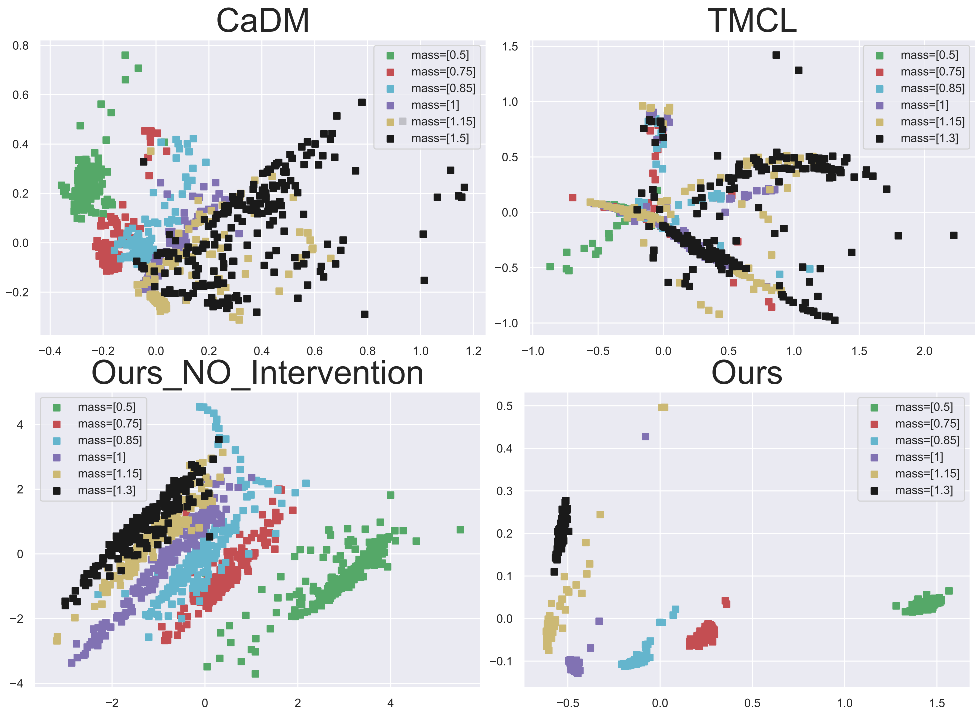


FIGURE C.9. The PCA of estimated context (environmental-specific) vectors in **Pendulum** task, where mass = 0.5 and mass =1.3 are from test environments.

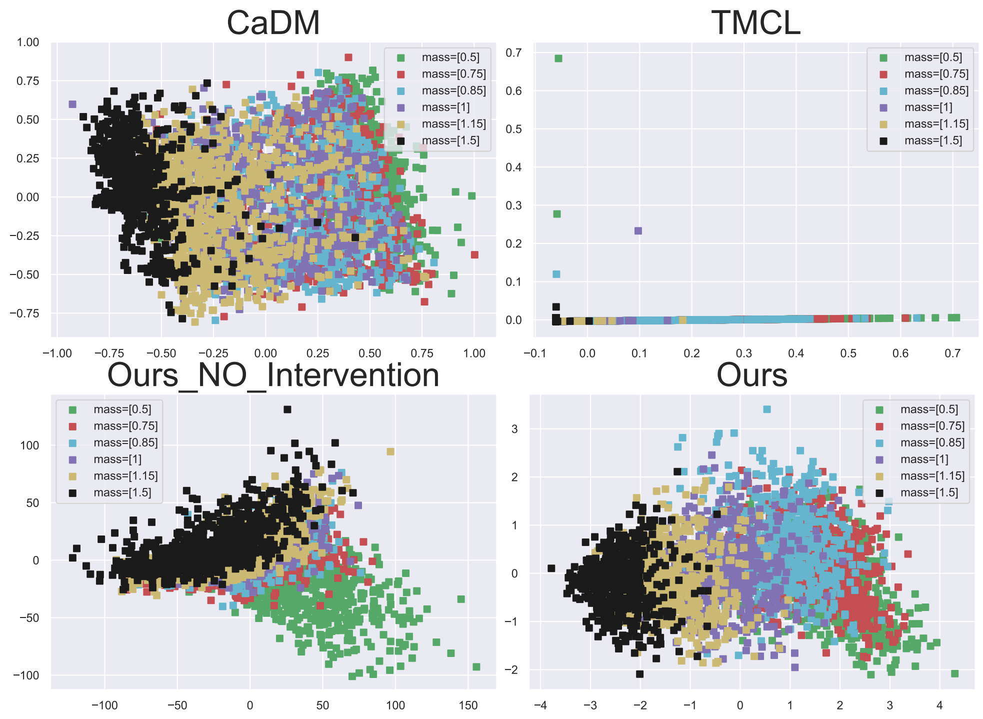


FIGURE C.10. The PCA of estimated context (environmental-specific) vectors in **HalfCheetah** task, where mass = 0.5 and mass =1.5 are from test environments.

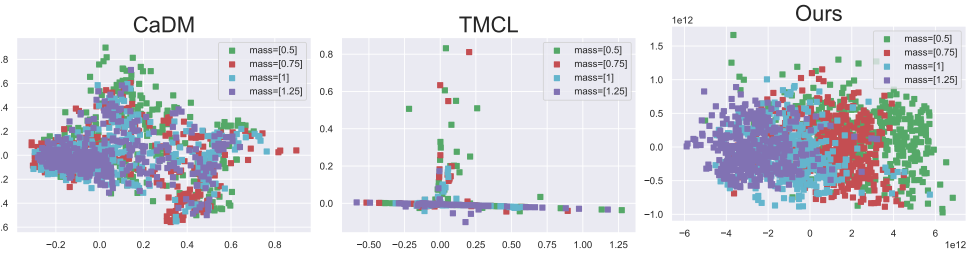


FIGURE C.11. The PCA of estimated context (environmental-specific) vectors in the **Hopper** task.

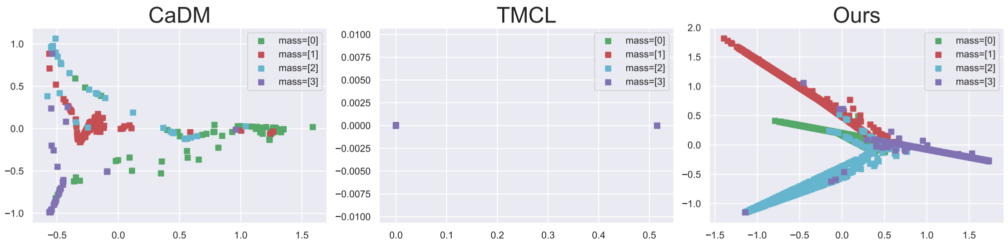


FIGURE C.12. The PCA of estimated context (environmental-specific) vectors in the **Cripple_Halfcheetah** task.

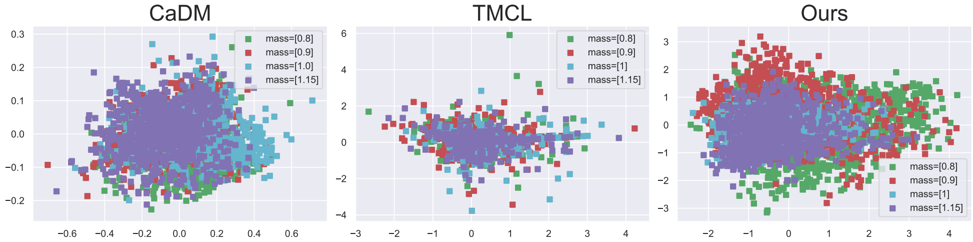


FIGURE C.13. The PCA of estimated context (environmental-specific) vectors in the **Slim_Humanoid** task.

Appendix D

Appendix of Chapter 5

D1 Appendix

We public all training details at Appendix D1.1 and D1.3.

D1.1 Environmental Settings

We follow the environmental settings of [135, 70] and give the details of settings as follows:

- **Pendulum** We modify the mass m and the length l of Pendulum to change its dynamics.
- **Half-Cheetah** We modify the mass of rigid link m and the damping of joint d of Half-Cheetah agent to change its dynamics.
- **Swimmer** We modify the mass of rigid link m and the damping of joint d of Swimmer agent to change its dynamics.
- **Ant** We modify the mass of ant's leg m to change its dynamics. Specifically, we modify two legs by multiplying its original mass with m, and others two with $\frac{1}{m}$.
- **Slim_Humanoid** We modify the mass of rigid link m and the dampling of joint d of the Slim_Humanoid agent to change its dynamics.
- **Hopper** We modify the mass of m of the Hopper agent to change its dynamics.

Specifically, all training and test parameter lists are set as $\{0.75, 0.8, 0.85, 0.90, 0.95, 1, 1.05, 1.1, 1.15, 1.2, 1.25\}$ and $\{0.2, 0.4, 0.5, 0.7, 1.3, 1.5, 1.6, 1.8\}$, respectively.

D1.2 Algorithm

The training procedure is give at Algorithm 2.

Algorithm 2 The training algorithm process of our relational intervention approach

```
Initialize parameters of context encoder $\phi$, dynamics prediction model $\theta$ and relational head $\varphi$
Initialize dataset $\mathcal{B} \leftarrow \emptyset$
for Each Iteration do
    sample environments $\mathcal{M}^i$ from training environments $\{\mathcal{M}_i^{tr}\}_{i=0}^{K}$   ▷ Collecting Data
    for $T = 1$ to TaskHorizon do
        Get the estimation of the environment-specified factor $\hat{z}_{t-k:t-1}^{i} = g(\tau_{t-k:t-1}^{i}; \phi)$
        Collect $(s_t, a_t, s_{t+1}, r_t, \tau_{t-k:t-1}^{i})$ from $\mathcal{M}^i$ with dynamics prediction model $\theta$
        Update $\mathcal{B} \leftarrow \mathcal{B} \cup (s_t, a_t, s_{t+1}, r_t, \tau_{t-k:t-1}^{i})$
        Initialize trajectory prototype $C_{tra}^{i}$ for each sampled trajectory
    end for
    for Each Dynamics Training Iteration do   ▷ Update $\phi$,$\theta$ and $\varphi$
        for $k = 1$ to K do
            Sample data $\tau_{t-k:t-1}^{i,b,P}$ , $C_{tra}^{i}$ and $\tau_{t-k:t-1}^{j,b,P}$ , $C_{tra}^{j}$ with batch size B,from $\mathcal{B}$
            Get the estimation of the environment-specified factor $\hat{z}_{t-k:t-1}^{i,B,P} = g(\tau_{t-k:t-1}^{i,B,P}; \phi)$ and
             $\hat{z}_{t-k:t-1}^{j,B,P} = g(\tau_{t-k:t-1}^{j,B,P}; \phi)$
            Estimate the similarity $w$ between $C_{tra}^{i}$ and $C_{tra}^{j}$
            Construct $w$ between $C_{tra}^{i}$ and $C_{tra}^{j}$
            Combing top_k similar $C_{tra}^{i}$ into environmental prototypes $C_{env}^{i}$
            $\mathcal{L}^{tot}$=$\mathcal{L}_{\phi,\theta}^{pred}(\tau_{t:M}^{i,B,,K}, \hat{z}_{t-k:t-1}^{i,B,P}) + \mathcal{L}_{\phi,\varphi}^{i-relation}(\hat{z}_{t-k:t-1}^{i,B,P}, C^{i})$ with prototypes in different levels.
            Update $\theta$ , $\phi$ , $\varphi \leftarrow \nabla_{\theta,\phi\,\varphi}\frac{1}{B}\mathcal{L}^{tot}$
        end for
    end for
end for
```

D1.3 Training Details

Similar to the [135, 70], we train our model-based RL agents and context encoder for 20 epochs, and we collect 10 trajectories by a MPC controller with 30 horizon from environments at each epoch. In addition, the cross entropy method (CEM) with 200 candidate actions is chosen as the planing method. Specifically, the batch size for each experiment is 128, β is 6e-1. All module are learned by a Adam optimizer with 0.001 learning rate.

D1.4 Prediction Error

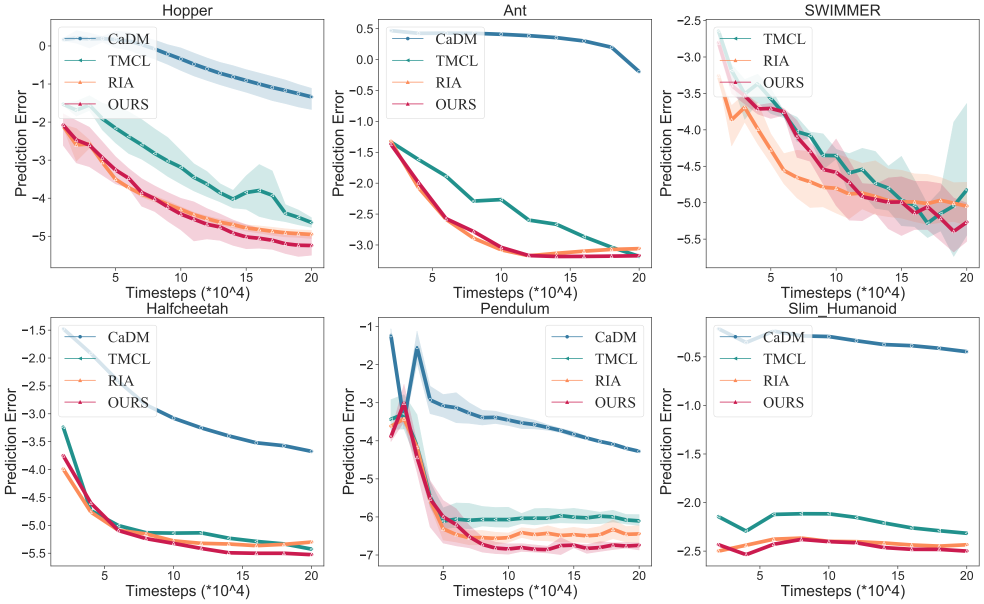


FIGURE D.1. The average prediction errors of model-based RL agents on training environments. The results show the mean and standard deviation of prediction errors averaged over five runs. Specifically, we use the no adaptation version of TMCL for a fair comparison.

TABLE D.1. The average prediction errors of model-based RL agents on test environments. The results show the mean and standard deviation of prediction errors averaged over five runs. Specifically, we use the no adaptation version of TMCL for a fair comparison.

	TMCL	CaDM	RIA	Ours
HalfCheetah	0.025±0.011	0.027±0.013	0.021±0.031	**0.015±0.022**
Pendulum	0.013±0.084	0.016±0.004	0.003±0.001	**0.002±0.001**
Ant	0.069±0.013	0.054±0.026	0.051±0.024	**0.037±0.012**
Hopper	0.061±0.013	0.069 ±0.028	0.032±0.023	**0.026±0.011**
Slim_Humanoid	0.111±0.023	0.145±0.021	0.105±0.012	**0.100±0.015**
Swimmer	0.078±0.065	0.067±0.085	0.037±0.065	**0.023±0.021**

D1.5 Network Details

Similar to the [135], the context encoder is constructed by a simple 3 hidden-layer MLP, and the output dim of environmental-specific vector $\hat{z}$ is 10. The relational head is modelled as a single FC layer. The dynamics prediction model is a 4 hidden-layer FC with 200 units.

D1.6 Direct Causal Effect between Trajectory Prototypes

Concretely, the direct causal effect difference between two trajectory prototypes c^j_{tra} and c^k_{tra} can be calculated through the controllable causal effect [197] given as following:

$$\begin{aligned} CDE_{c^j_{tra},c^k_{tra}}(s_t, a_t) =& \mathbb{E}[S_{t+1}|do(S_t = s_t, A_t = a_t), do(Z = c^j_{tra})] \quad \text{(D.1)} \\ &- \mathbb{E}[S_{t+1}|do(S_t = s_t, A_t = a_t), do(Z = c^k_{tra})] \quad \text{(D.2)} \\ =& \mathbb{E}[S_{t+1}|S_t = s_t, A_t = a_t, Z = c^j_{tra}] - \mathbb{E}[S_{t+1}|S_t = s_t, A_t = a_t, Z = c^k_{tra}], \quad \text{(D.3)} \end{aligned}$$

where do is the do-calculus [196]. Because we control all variables that can influence on the S_{t+1}, and there is no other confounder between the mediators (S_t, A_t) and S_{t+1} except Z [70], we can remove all do operators in equation D.2. Therefore, the intervention distribution of controlling Z, and (S_t, A_t) equation D.2 is equal to the conditional distribution equation D.3. In addition, the direct causal effects between c^j_{tra} and c^k_{tra} may differ for different values of S_t and A_t, so we should sample S_t and A_t independently of Z to calculate the average controlled direct effect.

Concretely, we directly use a mini-batch of S_t and A_t pairs (s^i_t, a^i_t) to calculate the average controlled direct effect of them as Figure 5.2 (a) shows:

$$w_{jk} = \frac{1}{N}\sum_{i=1}^{N}|CDE_{c^j_{tra},c^k_{tra}}(s^i_t, a^i_t)|, \quad \text{(D.4)}$$

where N is the batch size, j and k are the id of trajectory prototypes.

D1.7 Connection between Relation Loss and Mutual Information

We denote the environmental-specific factor as Z and its prototypes as C. By definition, the mutual information between Z and C should be:

$$I(Z;C) = \mathbb{E}_{P_{ZC}}[\log(\frac{p(z,c)}{p(z)p(c)})] \tag{D.5}$$

where P_{ZC} is he joint distribution of Z and C, and P_Z and P_C are their marginal distributions. To estimate mutual information between Z and C, we can use the probabilistic classifier method proposed by [256]. Concretely, we can use a Bernoulli random variable Y to classify one given data pair (z, c) from the joint distribution P_{ZC} ($Y = 1$) or from the product of marginal distribution $P(Z)P(C)$ ($Y = 0$) . Therefore, the mutual information $I(Z;C)$ between Z and C can be rewrite as:

$$\begin{aligned} I(Z;C) &= \mathbb{E}_{P_{ZC}}[\log(\frac{p(z,c)}{p(z)p(c)})] \\ &= \mathbb{E}_{P_{ZC}}[\log(\frac{p(z,c|Y=1)}{p(z,c|Y=0)})] \\ &= \mathbb{E}_{P_{ZC}}[\log(\frac{p(Y=0)P(Y=1|z,c)}{p(Y=1)P(Y=0|z,c)})] \end{aligned} \tag{D.6}$$

Obviously, $\frac{p(Y=0)}{p(Y=1)}$ can be approximated by the sample size, *i.e.* $\frac{n_{P_Z P_C}}{n_{P_{ZC}}}$, while $\frac{P(Y=1|z,c)}{P(Y=0|z,c)}$ can be measured by a classifier $h(Y|z, c)$ with the below our relational loss:

$$\mathcal{L}^{relation}_{\varphi,\phi} = -\Big[Y \cdot \log\ h([z,c];\varphi) + (1-Y)\cdot \log\left(1 - h([z,c];\varphi)\right)\Big], \tag{D.7}$$

where $Y = 1$ if the given pair (z, c) is from the joint distribution P_{XY}, and $Y = 0$ if the given pair (z, c) is from the product of the marginal distributions $P_Z P_C$. Because $\frac{p(Y=0)}{p(Y=1)}$ tend to be a constant, optimizing our relation loss is actually estimating the mutual information $I(Z;C)$ between Z and C. Therefore, optimizing equation D.7 is actually maximizing the mutual information between $(\hat{z})$ and its corrsponding prototype which represents the semantics of trajectorey or environment. If the readers are interested in the concrete bound about this method to estimate mutual information, please refer to [256, 70].

D1.8 Visualization and Analysis

TABLE D.2. The quantitative evaluation results of estimated environmental-specific factors.

	ARI				AMI				V-means			
	TMCL	CaDM	RIA	Ours	TMCL	CaDM	RIA	Ours	TMCL	CaDM	RIA	Ours
HalfCheetah	0.006	0.128	0.212	**0.570**	0.058	0.175	0.333	**0.681**	0.06	0.176	0.314	**0.680**
Pendulum	0.060	0.471	0.754	**0.971**	0.054	0.529	0.838	**0.967**	0.051	0.531	0.724	**0.975**
Slim_Humanoid	0.001	0.139	0.212	**0.472**	0.004	0.121	0.245	**0.613**	0.004	0.139	0.213	**0.612**
Swimmer	0.052	0.583	0.586	**0.615**	0.052	0.582	0.597	**0.638**	0.012	0.528	0.595	**0.637**

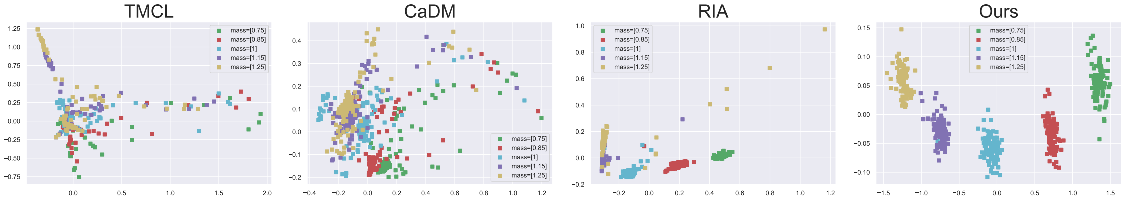


FIGURE D.2. The PCA visualization of environmental-specific factors estimated by TMCL [219], CaDM [135], RIA [70] and ours on the Pendulum task.

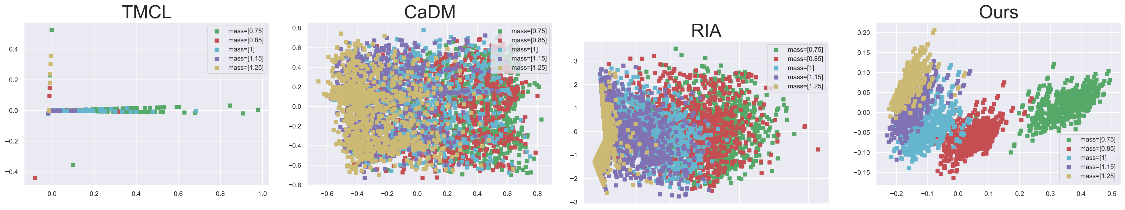


FIGURE D.3. The PCA visualization of environmental-specific factors estimated by TMCL [219], CaDM [135], RIA [70] and ours on the Halfcheetah task.

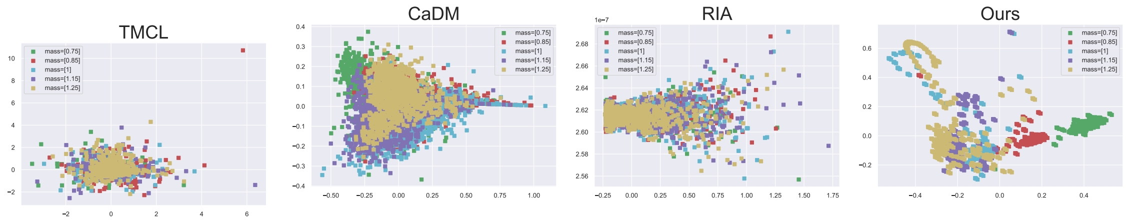


FIGURE D.4. The PCA visualization of environmental-specific factors estimated by TMCL [219], CaDM [135], RIA [70] and ours on the slim_humanoid task.

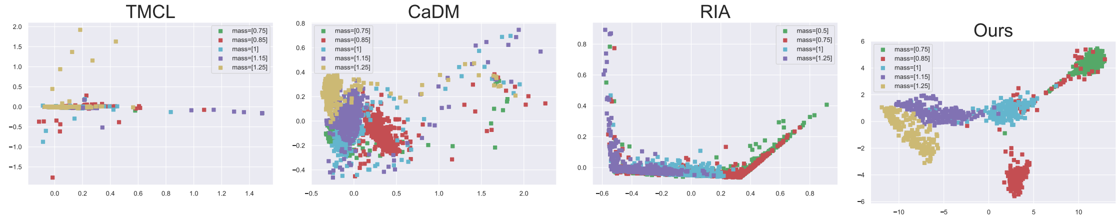


FIGURE D.5. The PCA visualization of environmental-specific factors estimated by TMCL [219], CaDM [135], RIA [70] and ours on the swimmer task.

APPENDIX E

Appendix of Chapter 6

E0.1 Details about Question-Relevant Caption Generation

Concretely, we denote features of image patches extracted by ITE as $f_v^i \in \mathbb{R}^{K \times D_v^i}$ and question features as $f_q^i \in \mathbb{R}^{L \times D_q^i}$, where i is the number of the layer of ITE, K is the number of images patches, L is the number of token in the given question, D_v^i is the dimension of patch feature in the i-th layer of ITE network and D_q^i is the dimension of textual feature in the i-th layer of ITE network. For cross-attention head in i-th layer, the cross-attention scores W^i between each image patch and each token in question can be calculated directly as

$$W^i = \text{softmax}\left(\frac{f_q^i W_Q^i {W_K^i}^\top {f_v^i}^\top}{\sqrt{D_q^i}}\right). \tag{E.1}$$

where $W_Q^i \in \mathbb{R}^{D_q^i \times D_q^i}$ is the query head and $W_K^i \in \mathbb{R}^{D_v^i \times D_q^i}$ is the key head in the i-th layer of ITE network. With Equation E.3, we obtain a cross-attention matrix $W^i \in \mathbb{R}^{L \times K}$, where each row is the cross-attention scores of each token in the question over all image patches. Specifically, the attention matrix W^i can be regarded as the patch importance for ITE to calculate the similarity of whole image and question, but it still contains redundancy that contributes only a minor performance loss [18], indicating that some patches are uninformative. In order to find these less relevant image patches, we follwing GradCAM and compute the derivative of the cross-attention score from ITE function $\text{sim}(v, q)$, *i.e.*, $\partial\, \text{sim}(v, q)/\partial W$, and multiplying its gradient matrix with the cross-attention scores element-wisely. The relevance of the k^{th} image patch with the question, r_k^i, can be computed as the average over H attention heads and the sum over L textual tokens:

$$r_k^i = \frac{1}{H} \sum_{l=1}^{L} \sum_{h=1}^{H} \min\left(0, \frac{\partial\, \mathrm{sim}(v,q)}{\partial W_{lk}^{ih}}\right) W_{lk}^{ih}, \tag{E.2}$$

where h is the index of attention heads and i is the layer index of ITE.

E0.2 Experimental Results of Supervised Learning Methods in A-OKVQA

We show the experimental comparisons between our method and supervised model on A-OKVQA dataset [216] as Table E.8 shows. We can observe that our method outperform almost all supervised model with smaller size language model. This strongly support our method's effectiveness in leveraging reasoning power of large language models.

TABLE E.1. The experimental comparisons with models trained in A-OKVQA training dataset.

Methods	A-OKVQA	
	Val	Test
Models Fine-Tuned in A-OKVQA Training Set		
Pythia [97]	25.2	21.9
ViLBERT [161]	30.6	25.9
LXMERT [248]	30.7	25.9
KRISP [171]	33.7	27.1
GPV-2 [105]	**48.6**	**40.7**
Zero-Shot Evaluation with Plug-in Frozen Large Language Model		
Ours$_{6.7B}$	33.3	32.2
Ours$_{13B}$	33.3	33.0
Ours$_{30B}$	36.9	36.0
Ours$_{66B}$	38.7	38.2
Ours$_{175B}$	42.9	**40.7**

E0.3 Template-Based Question Design

We design question templates for each part of speech type of answers as Table E.7 shows.

TABLE E.2. The question templates for answers with different part of speech.

Part of Speech of Answer	Question Templates
Noun	What item is this in this picture? What item is that in this picture?
Verb	What action is being done in this picture? Why is this item doing in this picture? Which action is being taken in this picture? What action is item doing in this picture? What action is item performing in this picture?
Adjective	How to describe one item in this picture? What is item's ADJ TYPE in this picture? What is the ADJ TYPE in this picture?
Num	How many things in this picture?

E0.4 Sensitive Analysis

We evaluate the sensitive analysis about the QA pairs and number of captions in prompt for LLM as Table E.8 shows. We can observe that the differences in QA scores on OK-VQA dataset are not higher than 1 as long as QA pairs in prompts. The results demonstrate the performance of our method is robust with different numbers of QA pairs and captions.

TABLE E.3. The experimental results of using different number of captions and QA pairs as prompts. The experiments are run on OK-VQA with OPT 30B.

Caption / QA Pairs	0	10	20	30	40	50
0	3.3	19.6	22.7	23.4	24.0	24.8
10	40.9	41.6	42.1	42.1	41.9	42.2
20	41.2	41.3	41.3	41.7	42.2	42.0
30	41.0	41.0	41.7	41.8	41.6	41.5
40	40.3	40.7	40.6	40.3	40.3	41.1
50	40.6	40.6	40.7	40.9	40.6	41.1

TABLE E.4. The experimental results of using different number of patches to generate question-relevant captions. The experiments are run on OK-VQA with OPT 30B.

Patch_num	10	20	40	Full
	41.2	41.8	41.6	39.8

TABLE E.5. The experimental results of generating different number of question-relevant captions. The experiments are run on OK-VQA with OPT 30B.

Caption_num	PICa	10	30	50	100
	17.7	38.3	40.9	41.4	41.8

E0.5 Examples

E0.6 Details about Question-Relevant Caption Generation

Concretely, we denote features of image patches extracted by ITE as $f_v^i \in \mathbb{R}^{K \times D_v^i}$ and question features as $f_q^i \in \mathbb{R}^{L \times D_q^i}$, where i is the number of the layer of ITE, K is the number of images patches, L is the number of token in the given question, D_v^i is the dimension of patch feature in the i-th layer of ITE network and D_q^i is the dimension of textual feature in the i-th layer of ITE network. For cross-attention head in i-th layer, the cross-attention scores W^i between each image patch and each token in question can be calculated directly as

$$W^i = \mathrm{softmax}\left(\frac{f_q^i W_Q^i {W_K^i}^\top {f_v^i}^\top}{\sqrt{D_q^i}}\right). \tag{E.3}$$

where $W_Q^i \in \mathbb{R}^{D_q^i \times D_q^i}$ is the query head and $W_K^i \in \mathbb{R}^{D_v^i \times D_q^i}$ is the key head in the i-th layer of ITE network. With Equation E.3, we obtain a cross-attention matrix $W^i \in \mathbb{R}^{L \times K}$, where each row is the cross-attention scores of each token in the question over all image patches. Specifically, the attention matrix W^i can be regarded as the patch importance for ITE to calculate the similarity of whole image and question, but it still contains redundancy that contributes only a minor performance loss [18], indicating that some patches are uninformative. In order to find these less relevant image patches, we follwing GradCAM and compute the derivative of the cross-attention score from ITE function

$\mathrm{sim}(v, q)$, *i.e.*, $\partial\, \mathrm{sim}(v, q)/\partial W$, and multiplying its gradient matrix with the cross-attention scores element-wisely. The relevance of the k^{th} image patch with the question, r^i_k, can be computed as the average over H attention heads and the sum over L textual tokens:

$$r^i_k = \frac{1}{H} \sum_{l=1}^{L} \sum_{h=1}^{H} \min\left(0, \frac{\partial\, \mathrm{sim}(v, q)}{\partial W^{ih}_{lk}}\right) W^{ih}_{lk}, \tag{E.4}$$

where h is the index of attention heads and i is the layer index of ITE.

E0.7 Experimental Results of Supervised Learning Methods in A-OKVQA

We show the experimental comparisons between our method and supervised model on A-OKVQA dataset [216] as Table E.8 shows. We can observe that our method outperform almost all supervised model with smaller size language model. This strongly support our method's effectiveness in leveraging reasoning power of large language models.

TABLE E.6. The experimental comparisons with models trained in A-OKVQA training dataset.

Methods	A-OKVQA	
	Val	Test
Models Fine-Tuned in A-OKVQA Training Set		
Pythia [97]	25.2	21.9
ViLBERT [161]	30.6	25.9
LXMERT [248]	30.7	25.9
KRISP [171]	33.7	27.1
GPV-2 [105]	**48.6**	**40.7**
Zero-Shot Evaluation with Plug-in Frozen Large Language Model		
Ours$_{6.7B}$	33.3	32.2
Ours$_{13B}$	33.3	33.0
Ours$_{30B}$	36.9	36.0
Ours$_{66B}$	38.7	38.2
Ours$_{175B}$	42.9	**40.7**

E0.8 Template-Based Question Design

We design question templates for each part of speech type of answers as Table E.7 shows.

TABLE E.7. The question templates for answers with different part of speech.

Part of Speech of Answer	Question Templates
Noun	What item is this in this picture? What item is that in this picture?
Verb	What action is being done in this picture? Why is this item doing in this picture? Which action is being taken in this picture? What action is item doing in this picture? What action is item performing in this picture?
Adjective	How to describe one item in this picture? What is item's ADJ TYPE in this picture? What is the ADJ TYPE in this picture?
Num	How many things in this picture?

E0.9 Sensitive Analysis

We evaluate the sensitive analysis about the QA pairs and number of captions in prompt for LLM as Table E.8 shows. We can observe that the differences in QA scores on OK-VQA dataset are not higher than 1 as long as QA pairs in prompts. The results demonstrate the performance of our method is robust with different numbers of QA pairs and captions.

TABLE E.8. The experimental results of using different number of captions and QA pairs as prompts. The experiments are run on OK-VQA with OPT 30B.

Caption QA Pairs	0	10	20	30	40	50
0	3.3	19.6	22.7	23.4	24.0	24.8
10	40.9	41.6	42.1	42.1	41.9	42.2
20	41.2	41.3	41.3	41.7	42.2	42.0
30	41.0	41.0	41.7	41.8	41.6	41.5
40	40.3	40.7	40.6	40.3	40.3	41.1
50	40.6	40.6	40.7	40.9	40.6	41.1

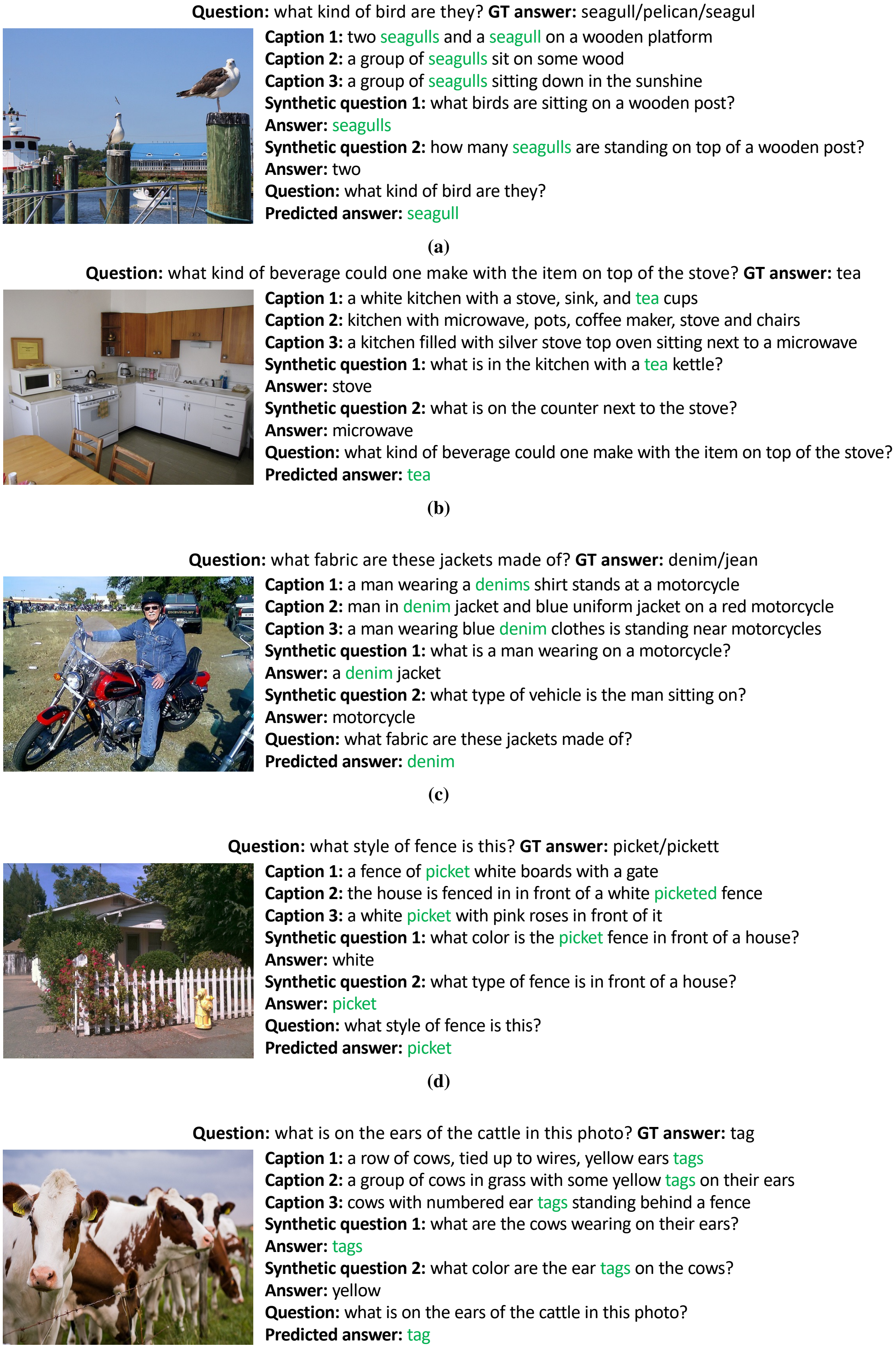
E0.10 Examples

TABLE E.9. The experimental results of using different number of patches to generate question-relevant captions. The experiments are run on OK-VQA with OPT 30B.

Patch_num	10	20	40	Full
	41.2	41.8	41.6	39.8

TABLE E.10. The experimental results of generating different number of question-relevant captions. The experiments are run on OK-VQA with OPT 30B.

Caption_num	PICa	10	30	50	100
	17.7	38.3	40.9	41.4	41.8



Question: what kind of bird are they? **GT answer:** seagull/pelican/seagul

Caption 1: two seagulls and a seagull on a wooden platform
Caption 2: a group of seagulls sit on some wood
Caption 3: a group of seagulls sitting down in the sunshine
Synthetic question 1: what birds are sitting on a wooden post?
Answer: seagulls
Synthetic question 2: how many seagulls are standing on top of a wooden post?
Answer: two
Question: what kind of bird are they?
Predicted answer: seagull

(a)

Question: what kind of beverage could one make with the item on top of the stove? **GT answer:** tea

Caption 1: a white kitchen with a stove, sink, and tea cups
Caption 2: kitchen with microwave, pots, coffee maker, stove and chairs
Caption 3: a kitchen filled with silver stove top oven sitting next to a microwave
Synthetic question 1: what is in the kitchen with a tea kettle?
Answer: stove
Synthetic question 2: what is on the counter next to the stove?
Answer: microwave
Question: what kind of beverage could one make with the item on top of the stove?
Predicted answer: tea

(b)

Question: what fabric are these jackets made of? **GT answer:** denim/jean

Caption 1: a man wearing a denims shirt stands at a motorcycle
Caption 2: man in denim jacket and blue uniform jacket on a red motorcycle
Caption 3: a man wearing blue denim clothes is standing near motorcycles
Synthetic question 1: what is a man wearing on a motorcycle?
Answer: a denim jacket
Synthetic question 2: what type of vehicle is the man sitting on?
Answer: motorcycle
Question: what fabric are these jackets made of?
Predicted answer: denim

(c)

Question: what style of fence is this? **GT answer:** picket/pickett

Caption 1: a fence of picket white boards with a gate
Caption 2: the house is fenced in in front of a white picketed fence
Caption 3: a white picket with pink roses in front of it
Synthetic question 1: what color is the picket fence in front of a house?
Answer: white
Synthetic question 2: what type of fence is in front of a house?
Answer: picket
Question: what style of fence is this?
Predicted answer: picket

(d)

Question: what is on the ears of the cattle in this photo? **GT answer:** tag

Caption 1: a row of cows, tied up to wires, yellow ears tags
Caption 2: a group of cows in grass with some yellow tags on their ears
Caption 3: cows with numbered ear tags standing behind a fence
Synthetic question 1: what are the cows wearing on their ears?
Answer: tags
Synthetic question 2: what color are the ear tags on the cows?
Answer: yellow
Question: what is on the ears of the cattle in this photo?
Predicted answer: tag

(e)

FIGURE E.1. Success case analysis for OK-VQA. Green color indicates answer cues and correct prediction.

Question: why is timing of the essence when delivering this food item? **GT answer:** temperature/hot still/stay hot

Caption 1: two pizza boxes have pepper pizza and take out
Caption 2: two boxes are opened up of two different pizzas
Caption 3: there are two small baked pizzas on the table
Synthetic question 1: what are two large pizzas sitting in?
Answer: boxes
Synthetic question 2: where are two large pizzas sitting next to each other?
Answer: table
Question: why is timing of the essence when delivering this food item?
Predicted answer: hot

(a)

Question: what era is this furniture from? **GT answer:** victorian/1940s

Caption 1: a living room with a small television in front of the window
Caption 2: a vintage tv is sitting on a nice table in the living room
Caption 3: a large house shaped model is sitting in a living room
Synthetic question 1: what type of room has a tv in the center?
Answer: living
Synthetic question 2: how large is the tv in the living room?
Answer: small
Question: what era is this furniture from?
Predicted answer: vintage

(b)

Question: what kind of sporting event is this? **GT answer:** soccer/not sure/pole vault

Caption 1: man on horse coming off from arena, holding something
Caption 2: a man is riding a horse during a soccer game
Caption 3: a man holding a red flag near a large person in a green field
Synthetic question 1: who is riding a horse in the middle of a stadium?
Answer: man
Synthetic question 2: what color is the flag on display at a football game?
Answer: red
Question: what kind of sporting event is this?
Predicted answer: football

(c)

Question: what type of clouds are in the picture? **GT answer:** cumulus/cumuli/nimbus

Caption 1: a cloudy - filled sky on a cloudy day over a zebras
Caption 2: the clouds are gray and full of clouds
Caption 3: there are many different clouds in this sky
Synthetic question 1: what is in the background of a photo of a zebra?
Answer: sky
Synthetic question 2: what type of sky is above on a cloudy day?
Answer: cloudy
Question: what type of clouds are in the picture?
Predicted answer: cloud

(d)

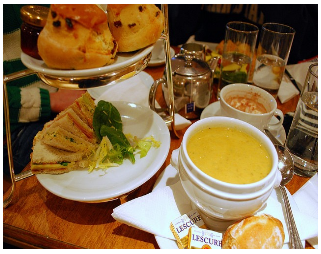
Question: how many people can this bus carry? **GT answer:** 50/40/39

Caption 1: a passenger bus traveling on a street side
Caption 2: blue commuter bus with parked on the side of the road
Caption 3: a bus that says aradara rides down the street
Synthetic question 1: what color bus is driving down the street?
Answer: blue
Synthetic question 2: what is making it's way down the street?
Answer: bus
Question: how many people can this bus carry?
Predicted answer: many

(e)

FIGURE E.2. Failure case analysis for OK-VQA. Red color indicates incorrect prediction.

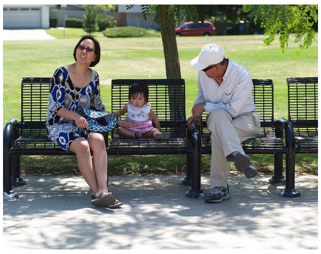
Question: which food has the least carbs? **GT answer:** soup/vegetable/salad



Caption 1: a table holding food including soup, sandwiches and fruit
Caption 2: the soup is very creamy in the bowl
Caption 3: sandwiches and soup is sitting on a table spread
Synthetic question 1: where is soup served on a table?
Answer: bowl
Synthetic question 2: what is on a plate next to a bowl of soup?
Answer: sandwich
Question: which food has the least carbs?
Predicted answer: soup

(a)

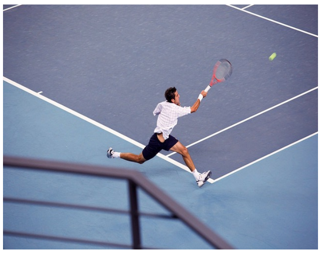
Question: in which way are the adults shown here likely related to the child? **GT answer:** parents/grandparents



Caption 1: a family sitting down on a bench in a park
Caption 2: a family sitting behind a park bench talking to a toddler
Caption 3: two people sitting on benches with a baby next to them
Synthetic question 1: what is sitting on a bench?
Answer: a baby
Synthetic question 2: who sits next to a toddler on a bench?
Answer: couple
Question: in which way are the adults shown here likely related to the child?
Predicted answer: parents

(b)

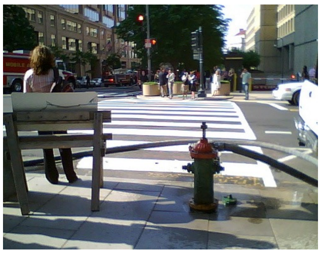
Question: what other surface is this game played on? **GT answer:** grass/clay/concrete



Caption 1: a blue surface with a blue tennis court
Caption 2: a man running across a blue tennis court with a racquet
Caption 3: a blue tennis court with a single game of tennis in progress
Synthetic question 1: what color is the tennis court?
Answer: blue
Synthetic question 2: what sport is a man playing on a blue court?
Answer: tennis
Question: what other surface is this game played on?
Predicted answer: grass

(c)

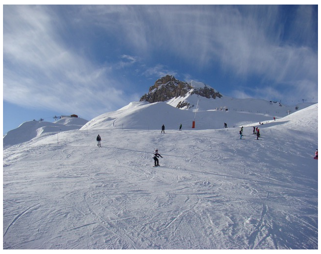
Question: what are they waiting to do when they stand next to the street? **GT answer:** cross/ride bus/light change



Caption 1: traffic and pedestrians at an intersection near a fire hydrant
Caption 2: a sidewalk and pedestrian crosswalk on a busy city street
Caption 3: a red fire hydrant stands besides a street that has a crosswalk
Synthetic question 1: where is a fire hydrant on a busy street?
Answer: crosswalk
Synthetic question 2: where are people waiting at a crosswalk?
Answer: intersection
Question: what are they waiting to do when they stand next to the street?
Predicted answer: cross

(d)

Question: what kind of resort are these people at? **GT answer:** ski resort/ski/snow



Caption 1: a group of people are skiing high up a slope
Caption 2: many people skiing down a ski slope during the day
Caption 3: a crowd of people on skis coming down the mountain
Synthetic question 1: what are people doing on a snow covered mountain?
Answer: ski
Synthetic question 2: who is skiing on a snow covered mountain?
Answer: people
Question: what kind of resort are these people at?
Predicted answer: ski resort

(e)

FIGURE E.3. Success case analysis for A-OKVQA. Green color indicates answer cues and correct prediction.

Question: this dish is suitable for which group of people? **GT answer:** vegetarian/vegan/family

Caption 1: a pasta dish sitting on top of a white plate
Caption 2: a broccoli pasta dish that has very pasta
Caption 3: a dish of pasta with noodles and tomato sauce
Synthetic question 1: what vegetable is on a white plate?
Answer: broccoli
Synthetic question 2: what color is a plate of pasta with broccoli on it?
Answer: white
Question: this dish is suitable for which group of people?
Predicted answer: children

(a)

Question: what is in front of the monitor? **GT answer:** chair/keyboard/webcam

Caption 1: a corner table with computer computer on the desk
Caption 2: a computer on the small desk in a small office area
Caption 3: view of a computer monitor in a light lit room
Synthetic question 1: what is a computer sitting on in a corner of a room?
Answer: desk
Synthetic question 2: how big is the desk in the corner?
Answer: small
Question: what is in front of the monitor?
Predicted answer: desk

(b)

Question: what type of shot is the woman about to hit? **GT answer:** forehand/tennis shot/swing

Caption 1: tennis player is hitting a tennis ball with her racket
Caption 2: a woman in pink outfit hitting a tennis ball
Caption 3: a woman in a cropped top and pants swinging a tennis racquet
Synthetic question 1: what is a tennis player doing with a tennis racket?
Answer: swinging
Synthetic question 2: who is swinging a tennis racket at a tennis ball?
Answer: woman
Question: what type of shot is the woman about to hit?
Predicted answer: volley

(c)

Question: what is in the bottles? **GT answer:** alcohol/liqueur/baileys

Caption 1: a sandwich on a plate with a glass of beer bottle
Caption 2: a table that has a sandwich, beer, and beer on it
Caption 3: a sandwich on a plate with a glass of beer bottle
Synthetic question 1: what is next to a sandwich and a beer?
Answer: bottle
Synthetic question 2: where is a sandwich with a beer and beer on a plate?
Answer: table
Question: what is in the bottles?
Predicted answer: beer

(d)

Question: why is the woman holding the umbrella? **GT answer:** shade/sun protection/get shadow

Caption 1: a young woman and the umbrella are on an orange blanket
Caption 2: a woman's umbrella and two dogs under an umbrella
Caption 3: a woman holding an umbrella is getting some light under her umbrella
Synthetic question 1: who is holding an umbrella while her dog sits under it?
Answer: woman
Synthetic question 2: what is a woman holding and a dog under it?
Answer: an umbrella
Question: why is the woman holding the umbrella?
Predicted answer: to protect herself from the sun

(e)

FIGURE E.4. Failure case analysis for A-OKVQA. Red color indicates incorrect prediction.

Question: what can the ram eat in this photo? **GT answer:** grass

Caption 1: the ram is standing outside on the green grass
Caption 2: a ram with white curly horns standing in a field
Caption 3: shaggy coated sheep with horns facing away in the center of a grass field
Synthetic question 1: where is a ram standing?
Answer: grass
Synthetic question 2: what animal is standing in a grassy field?
Answer: sheep
Question: what can the ram eat in this photo?
Predicted answer: grass

(a)

Question: what does the sign say? **GT answer:** stop

Caption 1: a stop sign with cloudy sky behind it
Caption 2: a red stop sign with a sky background
Caption 3: a tall stop sign on a rural road
Synthetic question 1: what color is the stop sign?
Answer: red
Synthetic question 2: what type of sky is behind a stop sign?
Answer: cloudy
Question: what does the sign say?
Predicted answer: stop

(b)

Question: what type animal is on the woman's pants? **GT answer:** owl/penguins

Caption 1: a girl is sitting on the ground in owl patterned pants
Caption 2: a woman with owly print pajamas pants is sitting in front of a pile of
Caption 3: a girl seated on the ground wearing pajamas
Synthetic question 1: where is a young girl wearing owl pants sitting?
Answer: the ground
Synthetic question 2: how is a young girl wearing owl pants doing?
Answer: sitting
Question: what type animal is on the woman's pants?
Predicted answer: owl

(c)

Question: how many children are at the table? **GT answer:** 3

Caption 1: three small little kids gather together on a dining table
Caption 2: a group of kids posing at a party table
Caption 3: three children sitting at a table with their food smiling at a picture
Synthetic question 1: what type of table are the three children sitting at?
Answer: dining
Synthetic question 2: how are the three children sitting at a table?
Answer: smiling
Question: how many children are at the table?
Predicted answer: 3

(d)

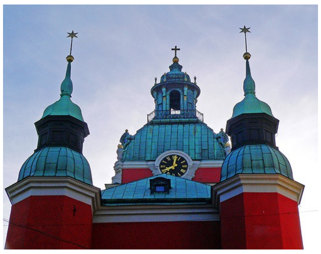
Question: is there broccoli in this dish? **GT answer:** yes

Caption 1: broccoli floret rice is in a large black pot
Caption 2: there is a closeup of a veggie salad
Caption 3: broccoli rice in a black bowl, ready to be eaten
Synthetic question 1: what is covered in broccoli in a pan?
Answer: rice
Synthetic question 2: what is a dish filled with broccoli and other vegetables in?
Answer: pot
Question: is there broccoli in this dish?
Predicted answer: yes

(e)

FIGURE E.5. Success case analysis for VQAv2. Green color indicates answer cues and correct prediction.

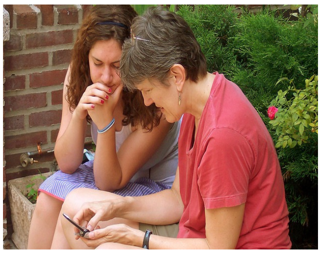
Question: what is atop this building? **GT answer:** cross/stars/cross and stars



Caption 1: the cathedral tower is with the clock on a steeple
Caption 2: a clock and a two crosses on top of a church
Caption 3: the top of a red cathedral with a clock on the tower
Synthetic question 1: what part of a building has a clock on it?
Answer: top
Synthetic question 2: what color is the building with a clock on top?
Answer: red
Question: what is atop this building?
Predicted answer: a clock

(a)

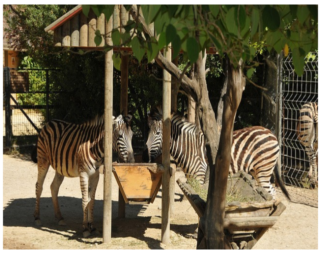
Question: what are they standing by? **GT answer:** bushes/tree/bricks



Caption 1: two girl sitting and talking, one is looking at something
Caption 2: an older woman and young woman using cellphones
Caption 3: two girls sitting on a brick wall during the day time
Synthetic question 1: who are sitting on a bench looking at their phones?
Answer: women
Synthetic question 2: what type of wall are the two women sitting on?
Answer: brick
Question: what are they standing by?
Predicted answer: brick wall

(b)

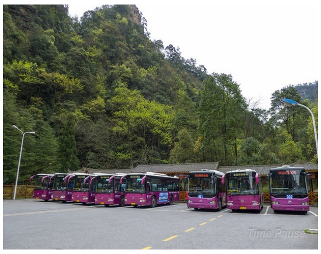
Question: how many zebras are there? **GT answer:** 3



Caption 1: two zebras walking by a feeder full of food
Caption 2: pair of zebras standing together at water trough in zoo
Caption 3: the zebras are eating out of a feeder box
Synthetic question 1: how many zebras are standing next to each other?
Answer: two
Synthetic question 2: what are the zebras doing?
Answer: eating
Question: how many zebras are there?
Predicted answer: 2

(c)

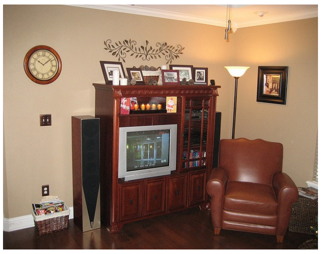
Question: how many buses are in the picture? **GT answer:** 8



Caption 1: a lot of buses sit parked in a line in front of a hill
Caption 2: a group of purple passenger buses all in a row
Caption 3: a row of purple bus buses next to each other
Synthetic question 1: how are the buses parked?
Answer: a line
Synthetic question 2: what color buses are parked in front of each other?
Answer: purple
Question: how many buses are in the picture?
Predicted answer: several

(d)

Question: are the numbers on the clock Roman numerals? **GT answer:** yes



Caption 1: a living room scene with a clock and tv
Caption 2: a chair is in front of a television that is being displayed
Caption 3: lounge chair with a clock that is hanging on the wall, and leather chair sits
Synthetic question 1: what is on in a living room?
Answer: television
Synthetic question 2: how is a wall clock displayed in a living room?
Answer: hanging
Question: are the numbers on the clock Roman numerals?
Predicted answer: no

(e)

FIGURE E.6. Failure case analysis for VQAv2. Red color indicates incorrect prediction.