Domain Generalization via Representation Learning

A dissertation submitted by

Boyang Lyu

in partial fulfillment of the requirements of

Doctor of Philosophy

in

Electrical and Computer Engineering

at

TUFTS UNIVERSITY

February 2024

© Tufts University 2024. All rights reserved.

Advisor: Shuchin Aeron

Domain Generalization via Representation Learning

Abstract

The success of traditional machine learning techniques is highly dependent on the assumption that the training and test data are drawn from independent and identical distributions. This assumption provides a theoretical guarantee and serves as a foundation for the high performance of traditional methods. However, in real-world applications, the performance of well-trained models often suffers from degradation due to the violation of this fundamental assumption. Factors such as environment, equipment, and human activity can easily lead to significant differences in data distribution across training and test data, which poses challenges to the generalization ability of models.

One direction of addressing the above problem is Domain Generalization, which aims to enhance the generalization ability of trained models, allowing them to perform well even on unseen test data with different distributions. In this thesis, we conduct a comprehensive review of previous work, focusing on the theoretical foundations, algorithms, and workflows associated with the representation learning-based domain generalization algorithms. We identify the gap between previous theoretical work and practical algorithms, and propose a novel theory to bridge this gap. We also explore the weakness of some domain generalization principles and propose an algorithm as a potential solution. In addition to focusing on algorithms, we recognize the importance of model selection for methods designed for domain generalization. In light of this, we propose a novel model selection method that takes into account the unique characteristics and challenges associated with domain generalization. This selection method considers the complexities of domain shifts and ensures the reliable assessment of model generalization across different domains. By incorporating this

validation method into the evaluation process, we can gain more insights into the application of domain generalization algorithms to practical problems.

Through this thesis, we contribute to the field of domain generalization by bridging the gap between previous theory and practice, offering potential solutions to address the failure cases observed in certain domain generalization methods and emphasizing the importance of considering the workflow of the domain generalization problem. The proposed theoretical advancements, algorithms, and validation method collectively aim to enable machine learning models to generalize effectively across diverse real-world domains with varying data distributions.

Acknowledgments

I am immensely fortunate to be embraced by a circle of individuals who embody compassion, kindness, and intelligence. Countless people have extended their warmth and support on this journey. Regrettably, it is impossible to enumerate them all here.

To my parents and grandparents. I was once asked if they would be proud of my graduation. The first answer came to my mind is "not really". Sounds weird, but later I realized that the answer came from a reason. I just know that their pride does not come from any academic achievements. Their pride is rooted in me as an individual. They have consistently stood by me, unwavering in their support for every choice I have made, all the while wishing for me to live a life that aligns with my true desires. I am truly blessed to have such parents and grandparents.

To my advisor Professor Shuchin Aeron, you may not fully realize how profoundly you have influenced and uplifted me. I consider myself incredibly fortunate to have had the privilege of being mentored by you. Every improvement I have made since embarking on this research journey owes itself to your guidance.

To my co-advisor, Professor Prakash Ishwar, your passion for learning, your unwavering commitment to discovery, and your sharp and critical thinking have left an indelible mark on me. I will forever cherish our meetings and discussions. Enrolling in your course that winter stands as one of the best decisions I have ever made. Our journey began there, and I earnestly hope that it will continue.

To Professor Sergio Fantini, you have broadened my horizons of research, guiding me into new scientific realms. Working alongside you in these past years has been a truly remarkable experience—one that I consider a stroke of luck. You are always inspiring, encouraging and willing to offer help and guidance. I treasure every moment we have shared.

To Professor Matthias Scheutz, you have guided me into a different dimension of research, teaching me to approach problems not merely from one perspective but

to explore them from the vantage points of diverse fields. You have taught me not to be bound by the knowledge I have acquired in one domain but to draw insights from others. I am grateful for your guidance and the profound inspiration you have provided, and I hold these lessons close to my heart.

To Thuan Nguyen, my only regret is that our paths did not intersect sooner. I extend my heartfelt gratitude for the numerous enriching discussions and unwavering support you have generously provided. This thesis will not be its current shape without you.

To Miriam Santi, you have been a pillar of support for different aspects of my school life. My college life would not have been such smooth without you.

To George D. Preble, I am profoundly grateful for the warmth and friendship you have extended to me. You have treated me like family, and I cherish every joyful moment we have shared.

To Joel Grodstein, during a particularly challenging period in my life, you extended a helping hand without even knowing me at that time. Your kindness will forever be etched in my memory.

To Ruijie Jiang and Shoaib Bin Masud, we have shared the highs and lows of this journey, and I am grateful for your companionship.

To my dearest friends Rui Cui, Meng Qu, Xiaoyi Ma, Wenjie Han, Yang Zhang, Meng Ji, Taoran Liu, Xiaocong Wang, Tong Huang, Jiuwei Chen, and Quan Guo, your friendship has been a constant source of strength. Regardless of the geographical distance that separates us, your presence always provides me with the power to overcome challenges. Conversations with you make even the toughest times more manageable, and life takes on a brighter hue when spent with you.

To my internship mentors Sandip Bose and Elizabeth Godoy, you are amazing people. I can only hope for future opportunities to collaborate with you further and continue my journey of learning from your exceptional expertise and guidance.

To Thao Pham, you have not only imparted the art of paper writing to me but also have patiently answered numerous naive questions from me. Collaborating with you has been a delightful experience, and I have greatly benefited from the wealth of knowledge and expertise you have shared.

To all my co-authors, I extend my heartfelt gratitude; your contributions have been instrumental in the success of our collaborative endeavors.

To new friends Boriana Boiadjieva and Aleksandar Sarić, whom I had the pleasure of meeting at Tufts, I want to say that school life would not have been as vibrant and enjoyable without your presence.

Indeed, languages can sometimes feel inadequate, especially when they are not one's native tongue. Nevertheless, the bonds formed with exceptional individuals have transcended linguistic boundaries. To all those fantastic people who have graced my life, your support not only brightens the dark days but also infuses ordinary days with warmth and joy.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

We have witnessed great success in machine learning models and their applications in the real world. The natural language generation model can generate human-like languages, serving as an editing assistant [3], the large AI vision model can automatically detect and localize target objects [75], and the model trained by medical images can aid in the early detection and diagnosis of diseases [142]. However, performance degradation is often observed when the well-trained model is applied in the real world.

For example, a language model trained using a specific corpus related to one particular linguistic register may struggle to perform well when faced with a different corpus with a distinct linguistic register [137]. Similarly, a medical image processing model trained exclusively on data from one hospital may not exhibit optimal performance when applied to test data from another hospital [57]. Why do models perform worse in these cases? To answer this question, let us delve into the algorithms employed for training these models. Most models, including various types of deep neural networks, are trained using Empirical Risk Minimization (ERM) [129]. Though ERM offers theoretical guarantees on performance [129], there is a critical assumption underlying the ERM framework, namely the training and test data should be independent and

identically distributed (i.i.d.). Given this fact, it becomes easier to understand the reasons behind the performance deterioration: the primary cause lies in the disparity between the data used during training and the data encountered during application. In other words, the models struggle to adapt effectively to new data that deviates from the distribution of the training data. As a consequence, they have difficulties in generalizing their learned knowledge, resulting in sub-optimal performance when confronted with novel, unseen data.

Methods that aim to mitigate this problem are broadly classified into two categories, namely Domain Adaptation (DA) [22] and Domain Generalization (DG) [24]. Both DA and DG aim to find a model that can generalize well in scenarios when the training data from the seen domain does not share the same data distribution as the test data from the unseen domain. Although sharing the similar objective, there is a fundamental difference in the settings for DA and DG.

DA typically requires the presence of unlabeled test data at the training time for the model to perform distribution matching between training and test data. However, this dependence on test data can limit the power of DA methods. By primarily focusing on matching the distributions of the training and test data, DA methods may restrict the model's generalization ability to the distributions encountered during the training [8], and when confronting data exhibiting a different distribution shift, re-training may be unavoidable. Additionally, accessing the test data at training time is not always trivial. For instance, in medical image classification, distribution shifts occur in data collected from different patients, but it is impossible to collect all data from future, unseen patients for the model training [151].

As a step forward, Domain Generalization (DG) [24], a more general framework has been proposed to address the challenges posed by the shift in distribution and the absence of test data [151]. The basic assumption of DG is that the training data is composed of data with multiple related but distinct distributions, or in other words,

there are multiple distinct seen domains during the training time for the model to learn from. Importantly, under the DG setting, the unseen test data is neither required nor utilized during the training phase. In the end, DG aims to develop models that exhibit strong performance on data with both seen and unseen distributions. Compared to DA, DG is more challenging, but is also more practical and aligns more closely with real-world applications. However, it is also worth noting that, regardless of the appeal of the goal, the DG problem cannot be tackled without proper assumptions on the unseen test data, as once the test cases are arbitrary, the model can perform arbitrarily bad. As a result, the development trajectory of DG largely overlaps with the DA problem, particularly in theoretical work.

In this thesis, we will first review the theoretical foundations of the DG problem, which were initially developed for the DA problem. We will focus on the limitations of these theoretical works, especially when used to guide practical algorithm development. Subsequently, we will delve into methods specifically designed for the DG problem, pay close attention to their weaknesses, and propose potential solution to address them. Furthermore, we aim to broaden our perspective beyond the model and algorithms themselves, shifting our attention to the whole workflow of DG. Specifically, we will rethink the long-standing methods used for model selection and validation during DG development, discuss potential issues associated with these methods, and propose future directions for improvement.

1.1 Outline

The main body of thesis include 5 chapters:

- In Chapter 2, we revisit the theoretical works that form the basis of many DG algorithms. This chapter introduces how the risk on the unseen domain (data) is bounded. As most of these works are derived from Domain Adaptation (DA), we

provide a review of DA, highlighting its significant impact on the development of DG. Additionally, we present different branches specifically designed for DG and their corresponding algorithms. This chapter serves as a comprehensive literature review on the DG problem.

- Chapter 3 is divided into two parts. In Part I, we analyze the limitations of previous theoretical bounds for the risk of the unseen domain. Identifying the gaps between theory and practical algorithms, we propose a new bound that overcomes these limitations, bridging the gap between previous theory and practice. Additionally, we introduce a novel algorithm for domain generalization based on our new upper bound and demonstrate the superior performance of our algorithm on several benchmark datasets for DG. Part II complements the theoretical insights of Part I. We provide an alternative lens through which to understand the DG problem, underscoring the importance of imposing specific constraints on the representation function. This chapter includes the joint works with Thuan Nguyen, Prakash Ishwar, Matthias Scheutz, and Shuchin Aeron, published as [89] and [103]. Specifically, the author of this thesis makes the following contributions to Part I work: (1) identifying the limitation of the previous theoretical work and the mismatch between previous theory and current implementation; (2) proposing the initial version of the theoretical bound with the invertibility constraint on the representation function. This stage of work can be found on arXiv [1] [2]; (3) assisting co-authors in refining the initial theoretical work by relaxing the invertibility constraint on representation function; (4) proposing the practical algorithm and conducting related experiments. For Part II, the author contributes on (1) proposing the reconstruction loss term for information preservation; (2) proving the trade-off between minimizing the reconstruction

[1] https://arxiv.org/pdf/2109.01902v3.pdf
[2] https://arxiv.org/pdf/2109.01902v4.pdf

loss and the domain discrepancy; (3) conducting related experiments.

- Chapter 4 delves deeper into the domain-invariant representation learning approach, concentrating specifically on scenarios that may lead to the breakdown of DG algorithms. In this chapter, rather than highlighting the success of previous DG work, we primarily focus on the limitations and weaknesses of existing principles and approaches. Based on these reflections, we propose a potential solution grounded in both theoretical insight and practical application. The main content of this chapter is based on the joint work with Thuan Nguyen, Prakash Ishwar, Matthias Scheutz, and Shuchin Aeron [101]. The author of this thesis specifically contributes on the algorithm design and its practical implementations for conducting corresponding experiments.

- In Chapter 5, we emphasize the importance of model validation and selection methods instead of models and algorithms themselves. We show that despite receiving limited attention so far, the validation and model selection methods for DG are not trivial. We demonstrate that some long-standing methods adopted by previous DG works may not be appropriate in the DG context. To support our argument, we provide proof and introduce a novel model selection method that has shown effectiveness compared to previous approaches. This chapter is based on the joint work with Thuan Nguyen, Matthias Scheutz, Prakash Ishwar, and Shuchin Aeron [90]. Specifically, contributions of the author of this thesis can be summarized as: (1) identifying limitations of conventional model validation/selection methods for the DG problem; (2) proposing the new validation method specifically designed for DG; (3) jointly proposing the theoretical foundations for the proposed validation method with co-authors.

Publications not included in this thesis

In order to align with the scope and focus of this thesis, the author has chosen to

include only a portion of the research work in this thesis. Other research work that falls outside the scope of this thesis includes:

- **Boyang Lyu**, Thao Pham, Giles Blaney, Zachary Haga, Angelo Sassaroli, Sergio Fantini, and Shuchin Aeron. Domain adaptation for robust workload level alignment between sessions and subjects using fNIRS. Journal of Biomedical Optics, 26(2):1 – 21, 2021 [91]

- Thuan Nguyen, **Boyang Lyu**, Prakash Ishwar, Matthias Scheutz, and Shuchin Aeron. Joint covariate-alignment and concept-alignment: a framework for domain generalization. In 2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP), pages 1–6. IEEE, 2022 [102]

- Ayca Aygun, **Boyang Lyu**, Thuan Nguyen, Zachary Haga, Shuchin Aeron, and Matthias Scheutz. Cognitive workload assessment via eye gaze and eeg in an interactive multi-modal driving task. In Proceedings of the 2022 International Conference on Multimodal Interaction, pages 337–348, 2022 [18]

- Matthias Scheutz, Shuchin Aeron, Ayca Aygun, JP de Ruiter, Sergio Fantini, Cristianne Fernandez, Zachary Haga, Thuan Nguyen, and **Boyang Lyu**. Estimating systemic cognitive states from a mixture of physiological and brain signals. Topics in Cognitive Science, 2023 [116]

1.2 Notations

In this section, we will introduce some general notations. These notations will be used throughout this thesis unless otherwise stated.

- $\mathcal{X} \subseteq \mathbb{R}^d$, $\mathcal{Y} \subseteq \mathbb{R}$ denote the input space and the label space, respectively.

- X is the input random variable, Y is the label random variable. The corresponding input data and label are denoted as $\boldsymbol{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$.
- We use superscript s to denote the observed (seen) domain and superscript u to denote the unseen test domain.

Chapter 2

From Domain Adaptation to Domain Generalization

In this chapter, we will conduct a comprehensive and review of the key concepts used in Domain Generalization (DG), highlighting the significant overlap between DG and Domain Adaptation (DA). By thoroughly exploring these concepts, we aim to provide a solid foundation for understanding the principles and methodologies employed in DG.

2.1 Empirical Risk Minimization

2.1.1 Notation

Consider a *domain* v as a triple $(\mu^{(v)}, f^{(v)}, g^{(v)})$ consisting of a distribution $\mu^{(v)}$ on the input $\boldsymbol{x} \in \mathbb{R}^d$, a representation function $f^{(v)} : \mathbb{R}^d \to \mathbb{R}^{d'}$, from the input space to the representation space, and a stochastic labeling function $g^{(v)} : \mathbb{R}^{d'} \to \mathcal{Y}$ from the representation space to the label space. Samples are independently drawn from each domain. We denote the unseen domain by $(\mu^{(u)}, f^{(u)}, g^{(u)})$ and S seen domains by $(\mu^{(s)}, f^{(s)}, g^{(s)})$, with $s = 1, \ldots, S$. Let $\mathcal{F} = \{f | f : \mathbb{R}^d \to \mathbb{R}^{d'}\}$ be the set of

representation functions, $\mathcal{G} = \{g|g : \mathbb{R}^{d'} \to \mathcal{Y}\}$ the set of stochastic *labeling functions*, $\mathcal{H} := \mathcal{G} \circ \mathcal{F}$ the set of *hypotheses*, with each hypothesis $h : \mathbb{R}^d \to \mathcal{Y}$ obtained by composing a $g \in \mathcal{G}$ with an $f \in \mathcal{F}$, *i.e.*, $h = g \circ f$. $f_{\#}\mu^{(v)}$ denotes the pushforward of distribution $\mu^{(v)}$ under the representation function f, *i.e.*, the distribution of $f(\boldsymbol{x})$ with $\boldsymbol{x} \sim \mu^{(v)}$. $h^{(v)} = g^{(v)} \circ f^{(v)}$ denotes the ground truth labeling rules.

The risk of using a hypothesis h in domain v is then defined by:

$$R^{(v)}(h) := \mathbb{E}_{\boldsymbol{x}\sim\mu^{(v)}}\left[\ell(h(\boldsymbol{x}), h^{(v)}(\boldsymbol{x}))\right], \tag{2.1}$$

where $\mathbb{E}[\cdot]$ denotes the expectation, $h^{(v)} = g^{(v)} \circ f^{(v)}$, and $\ell(\cdot,\cdot) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function that measure the difference between the hypothesis h and the true labeling $h^{(v)}$. These notations will be used in the following part. To be consistent with theory, when referring to algorithm design, we also denote f as feature extractor and g as classifier.

2.1.2 Empirical Risk Minimization for Single Domain

Minimizing the risk to find the optimal hypothesis h^* over the hypotheses set $\mathcal{H}$ is the fundamental idea of supervised machine learning algorithms, as shown below:

$$h^* = \operatorname*{argmin}_{h\in\mathcal{H}} R^{(v)}(h) \tag{2.2}$$

However, $R^{(v)}(h)$ is intractable due to the lack of knowledge on $\mu^{(v)}$. Thus, in practice, given a set of independent and identically distributed (i.i.d.) samples $\{(\boldsymbol{x}_i^{(v)}, y_i^{(v)})\}_{i=1}^{N^v}$

drawn from domain v, we compute an approximation of the risk as,

$$\hat{R}^{(v)}(h) = \frac{1}{N^{(v)}} \sum_{i=1}^{N^v} \ell(h(\boldsymbol{x}_i^{(v)}), h^{(v)}(\boldsymbol{x}_i^{(v)})) \tag{2.3}$$

$$= \frac{1}{N^{(v)}} \sum_{i=1}^{N^v} \ell(h(\boldsymbol{x}_i^{(v)}), y_i^{(v)}) \tag{2.4}$$

denoted as empirical risk. Most supervised machine learning algorithms optimize the above objective to find the optimal h^* in the hypotheses set $\mathcal{H}$ on domain v.

2.1.3 Empirical Risk Minimization for Multiple Domains

Now, considering the input is not only from one domain but from several different domains $(\mu^{(s)}, f^{(s)}, g^{(s)})$, with $s = 1, \ldots, S$, it is natural to combine all data from S domains to form a new dataset and apply the ERM, leading to the following objective:

$$\underset{h \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{N^s} \frac{1}{N^s} \ell(h(\boldsymbol{x}_i^{(s)}), h^{(s)}(\boldsymbol{x}_i^{(s)})) \tag{2.5}$$

where $h^{(s)} = g^{(s)} \circ f^{(s)}$ and $h^{(s)}(\boldsymbol{x}_i^{(s)}) = y_i^{(s)}$

The combination of data from different domains can be broadly observed in practical problems. For example, to train a model that can distinguish cell types in the blood samples from patients, it is inevitable to gather samples from different patients, *i.e.* different domains, to form the training dataset. Similarly, a face recognition model needs the training data to contain faces from different races, gender, etc. to achieve good accuracy.

2.2 Domain Adaptation

Although combining data from different domains (sources) is a natural extension of the ERM method, allowing the model to be exposed to more diverse data and potentially

improving its performance on unseen data, the model remains vulnerable to scenarios where the test data distribution diverges from the training data. Specifically, issues arise when the i.i.d. assumption is violated.

To address this issue, Domain Adaptation (DA) [22, 77] emerge as a potential solution. Formally, the DA problem assumes a set of data and label pairs $\{(\boldsymbol{x}_i^{(s)}, y_i^{(s)})\}_{i=1}^{N^s}$ from the seen domain $(\mu^{(s)}, f^{(s)}, g^{(s)})$ and the unlabeled data $\{\boldsymbol{x}_i^{(u)}\}_{i=1}^{N^u}$ from the unseen test domain $(\mu^{(u)}, f^{(u)}, g^{(u)})$ are both accessible. Under such setting, a formal theory on the risk in the unseen domain for binary classification problems is first proposed in [22], as stated below:

Theorem 2.2.1. (Theorem 1 in [22]). *Let f be a fixed representation function from the input space to representation space and $\mathcal{G}$ be a hypothesis space of VC-dimension k. If random labeled samples of size m are generated by applying f to i.i.d. samples from the seen domain, then with probability at least $1-\delta$, for every $g \in \mathcal{G}$:*

$$R^{(u)}(g) \leq R^{(s)}(g) + d_{\mathcal{G}}(f_{\#}\mu^{(u)}, f_{\#}\mu^{(s)}) + \lambda \tag{2.6}$$

$$\leq \hat{R}^{(s)}(g) + \sqrt{\frac{4}{m}\big(k \log \frac{2em}{k} + \log \frac{4}{\delta}\big)} + d_{\mathcal{G}}(f_{\#}\mu^{(u)}, f_{\#}\mu^{(s)}) + \lambda \tag{2.7}$$

where e is the base of the natural logarithm, $d_{\mathcal{G}}$ is $\mathcal{H}$-divergence[1]*, $R^{(u)}(g) = \mathbb{E}_{\boldsymbol{z}\sim f_{\#}\mu^{(u)}}\big[|g(\boldsymbol{z}) - g^{(u)}(\boldsymbol{z})|\big]$ denotes the risk in the unseen domain, $R^{(s)}(g) = \mathbb{E}_{\boldsymbol{z}\sim f_{\#}\mu^{(s)}}\big[|g(\boldsymbol{z}) - g^{(s)}(\boldsymbol{z})|\big]$ and $\hat{R}^{(s)}(g)$ denote the risk in the seen domain and its empirical estimation, respectively, and:*

$$\lambda = \inf_{g\in\mathcal{G}} \big(R^{(s)}(g) + R^{(u)}(g)\big) \tag{2.8}$$

is the combined risk.

$\mathcal{H}$-divergence [72] is a measure to quantify the distance between two distributions,

[1]Please note that in [22], the authors employed different notations compared to ours, where $\mathcal{H}$ represented the hypothesis space of the labeling function, whereas in our work, it is denoted as $\mathcal{G}$. Although we have adhered to the naming convention used in the original work, we wish to bring this subtle difference to the readers' attention.

the $\mathcal{H}$-divergence between two distributions $\mu^{(u)}, \mu^{(s)}$ is written as:

$$d_{\mathcal{H}}(\mu^{(u)}, \mu^{(s)}) = 2 \sup_{h \in \mathcal{H}} | \Pr_{x \sim \mu^{(u)}} (h(x) = 1) - \Pr_{x \sim \mu^{(s)}} (h(x) = 1)| \tag{2.9}$$

$\mathcal{H}$-divergence is firstly proposed in [72]. It is guaranteed to be equal or smaller than the L^1 distance, and when the hypothesis classes have finite VC-dimension, $\mathcal{H}$-divergence is general smaller than the L^1 distance and can be estimated from the finite samples.

Though the above upper bound sheds some light to the DA problem, for example, (2.8) indicates that we cannot expect a good generalization if no classifier can perform well on both the seen and unseen domain, there exists a fundamental problem that the whole bound depends on an unknown representation function f. Simply minimizing the above upper bound may not lead to a true DA due to this dependency.

Later, the authors extend their work for establishing a more general upper bound for the risk of the unseen domain. They consider a binary classification problem with hypothesis function $h : \mathcal{X} \to \mathcal{Y}$, thereby eliminating the constraint tied to the fixed representation function f. In this context, Ben-David *et al.* [21] propose the following theorem to bound the risk on the unseen domain:

Theorem 2.2.2. (Theorem 2 in [21]). *Let $\mathcal{H}$ be a hypothesis space of VC-dimension k, $\mathcal{U}^{(s)}$,$\mathcal{U}^{(u)}$ be two sets of unlabeled data with size of m' drawn independently from domain $\mu^{(s)}$ and $\mu^{(u)}$, respectively, then with probability at least $1-\delta$, for every $h \in \mathcal{H}$:*

$$R^{(u)}(h) \leq R^{(s)}(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}^{(s)},\mathcal{U}^{(u)}) + \lambda + C(m, k, \delta) \tag{2.10}$$

where $C(m, k, \delta) = 4\sqrt{\frac{2d\log(2m')+\log(\frac{2}{\delta})}{m'}}$, $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ is empirical estimated $\mathcal{H}\Delta\mathcal{H}$-divergence, $R^{(u)}(h) = \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\left[|h(\boldsymbol{x}) - h^{(u)}(\boldsymbol{x})|\right]$ denotes the risk in the unseen domain, $R^{(s)}(h) =$

$\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\left[|h(\boldsymbol{x}) - h^{(s)}(\boldsymbol{x})|\right]$ *denote the risk in the seen domain, and:*

$$\lambda = \inf_{h\in\mathcal{H}} \left(R^{(s)}(h) + R^{(u)}(h)\right) \tag{2.11}$$

is the combined risk.

A new measure $\mathcal{H}\Delta\mathcal{H}$-divergence [21] is introduced in the above theorem. Specifically, $\mathcal{H}\Delta\mathcal{H}$ is a symmetric difference hypothesis space of the hypothesis space $\mathcal{H}$, denoted as $\mathcal{H}\Delta\mathcal{H} = \{h(x) \oplus h'(x) | h, h' \in \mathcal{H}\}$, where $\oplus$ is the XOR function. Thus, $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}^{(s)}, \mathcal{U}^{(u)})$ can be written as

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}^{(s)}, \mathcal{U}^{(u)}) = 2 \sup_{h,h'\in\mathcal{H}} | \Pr_{x\sim\mu^{(u)}}(h(x) \neq h'(x)) - \Pr_{x\sim\mu^{(s)}}(h(x) \neq h'(x))| \tag{2.12}$$

Under such definition, every hypothesis in the $\mathcal{H}\Delta\mathcal{H}$ space stands for the set of difference between two hypotheses in the original $\mathcal{H}$ space [21]. The introduce of $\mathcal{H}\Delta\mathcal{H}$-divergence resolves the problem led by the L^1 distance by explicitly taking the hypothesis class into account [25]. Additionally, since $\mathcal{H}\Delta\mathcal{H}$-divergence is always smaller than or equal to the L^1 distance, it lead to a tighter upper bound compared to using the L^1 distance [21].

Given these advantages of the $\mathcal{H}\Delta\mathcal{H}$-divergence, Ben-David *et al.* subsequently propose its empirical estimation using finite samples from both seen and unseen domains:

Lemma 2 in [21]. *For $\mathcal{H}\Delta\mathcal{H}$ space and sample set $\mathcal{U}$ and $\mathcal{U}'$ of size m, the empirical $\mathcal{H}\Delta\mathcal{H}$-divergence is*

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}, \mathcal{U}') = 2\left(1 - \min_{h\in\mathcal{H}}\left[\frac{1}{m}\sum_{h(x)=0} I[x\in\mathcal{U}] + \frac{1}{m}\sum_{h(x)=1} I[x\in\mathcal{U}']\right]\right) \tag{2.13}$$

where $I[x\in\mathcal{U}]$ is 1 when $x\in\mathcal{U}$ holds and 0 otherwise.

As indicated in previous studies [21, 51], the estimation mentioned above can be

viewed as the error of a binary classification problem, in which samples from the seen and unseen domains are hypothetically labeled as 0 and 1, respectively. The value of the approximation can then be determined by calculating the error of the optimal classifier. The aforementioned theoretical works have start a new era of algorithm development for the Domain Adaptation (DA) problem. Building upon the foundations laid by [21, 22], a considerable body of research works [51, 85, 146, 147] have focused on minimizing the first and the second terms of the upper bounds (2.6) and (2.10) to achieve effective DA. More specifically, these algorithms seek to find representations that have unchanged distributions across domains and can also achieve small risk on the seen domain. Among these algorithms, we select Domain Adversarial Neural Network (DANN) [51] which serves as a pioneering approach in utilizing the aforementioned theoretical works as an exemplar of algorithms based on the domain-invariant representation learning.

2.2.1 Domain Adversarial Neural Network

Motivated by the theoretical works [21, 22], Domain Adversarial Neural Network (DANN) is first proposed in [51], where the authors explicitly state their key idea for DA as learning representations that informative of the task label but indiscriminate with respect to the shift between domains. To achieve this goal, the authors proposed an adversarial neural network-based algorithm, where the adversarial loss was used for approximating the divergence between seen and unseen domains. The algorithm's structure consists of three components: a common feature extractor f_{θ_e}, followed by two parallel classifiers $g_{\theta_c}, g_{\theta_d}$. With a little bit abuse of notation, we use f_{θ_e} to represent the encoder parameterized by θ_e, which is responsible for mapping the input to the representation space. Similarly, g_{θ_c}, parameterized by θ_c, represents the classifier mapping from the representation to the label space. So far, it is the typical model structure for the ERM, where both f_{θ_e} and g_{θ_c} are optimized together to minimize

the cross-entropy (CE) loss by the following objective function:

$$\mathsf{L}_c = -\frac{1}{N_s}\sum_{i=1}^{N_s} y_i^{(s)} \log p(g_{\theta_c}(f_{\theta_e}(\boldsymbol{x}_i^{(s)}))) \tag{2.14}$$

The standout component of the DANN algorithm is the introduction of the classifier g_{θ_d}, which is designed to distinguish the representation of the seen domain from those of the unseen domain. For clarity, we refer g_{θ_d} as domain discriminator, as widely used in DA works [31, 36, 44].The authors pseudo-labele the data from the seen domain as "0" (represented as $d^{(s)}$) and unseen domain as "1" (represented as $d^{(u)}$). Subsequently, they train the domain discriminator g_{θ_d} to minimize the following objective function:

$$\mathsf{L}_d = -\frac{1}{N^s}\sum_{i=1}^{N^s} d_i^{(s)} \log p(g_{\theta_d}(f_{\theta_e}(\boldsymbol{x}_i^{(s)}))) - \frac{1}{N^u}\sum_{j=1}^{N^u} d_i^{(u)} \log p(g_{\theta_d}(f_{\theta_e}(\boldsymbol{x}_j^{(u)}))) \tag{2.15}$$

Following the same logic as the Generative Adversarial Network (GAN) [54], the above two objectives are optimized in an adversarial way by optimizing:

$$(\theta_f^*, \theta_c^*) = \underset{\theta_f,\theta_c}{\operatorname{argmin}}(\mathsf{L}_d + \mathsf{L}_c) \tag{2.16}$$

$$\theta_d^* = \underset{\theta_d}{\operatorname{argmax}}\, \mathsf{L}_d \tag{2.17}$$

The feature extractor f_{θ_e} is trained to fool the domain discriminator by extracting indistinguishable representations from both the seen and unseen domain, while the domain discriminator aims to fight back to identify the source of the representation accurately. At the same time, feature extractor f_{θ_e} aims to learn representations that are useful for the target classification task.

The adversarial loss part is essential for DA both theoretically and practically. On one hand, from the theoretical perspective, it approximates and minimizes the $\mathcal{H}$-divergence in 2.13, thus approximately minimizing the second term in the upper

bounds derived in 2.6 and 2.10. On the other hand, the intuition behind this two-player game is interesting. It tries to eliminate the information about the domains from the extracted representations, making them to be domain-invariant representations. This concept is widely applied in the upcoming DA and DG works [120, 147].

However, though directly motivated by 2.2.1 and 2.2.2, a gap between the theory and practice exists. The second term that quantifies domain discrepancy lies in the data space instead of the representation space, in contrast to the implementation of the DANN algorithm. For now, let us keep it in mind and we will discuss this gap in detail in the next chapter.

2.2.2 Domain Adaptation Theory Development

Though [21, 22] have built solid foundations for the DA problem, some problems remain unsolved. As pointed out in [146], the combined risk term in 2.11 of Theorem 2.2.2 depends on the hypothesis class $\mathcal{H}$. As an extension of the previous theorems, Zhao *et al.* [146] propose the following theorem:

Theorem 2.2.3. (Theorem 4.1 in [146]). *Let $\mathcal{H}$ be a hypothesis class such that $h : \mathcal{X} \to [0, 1]$, $(\mu^{(s)}, h^{(s)})$ and $(\mu^{(u)}, h^{(u)})$ be the seen and unseen domains, then the following inequality holds:*

$$R^{(u)}(h) \leq R^{(s)}(h) + d_{\tilde{\mathcal{H}}}(\mu^{(u)}, \mu^{(s)}) + \min\{\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}[|h^{(s)}(\boldsymbol{x}) - h^{(u)}(\boldsymbol{x})|], \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}[|h^{(s)}(\boldsymbol{x}) - h^{(u)}(\boldsymbol{x})|]\} \tag{2.18}$$

where $\tilde{\mathcal{H}} := \{sign(|h(\boldsymbol{x}) - h'(\boldsymbol{x})| - t) \mid h, h' \in \mathcal{H}, 0 \leq t \leq 1\}$

The first two terms in the above bound have a similar interpretation as those in (2.6) and (2.10), while the last term is distinguishable. We can observe that unlike the combined risk term in (2.6) and (2.10), the last term in the above bound is free of the hypothesis class, thus intrinsic to the domain itself. Furthermore, the last term serves as another important part of the whole DA problem puzzle: it quantifies

the class-conditional distribution shift of the data between seen and unseen domains, which is not emphasized in previous theoretical works. Though taking both shift into account, it should be noted that Zhao *et al.* directly adopt the labeling rule as $h^{(s)}$ and $h^{(u)}$ for each domain, instead of considering the composition of $g^{(s)} \circ f^{(s)}$ and $g^{(u)} \circ f^{(u)}$, which leads to the same issue as we mentioned in Section 2.2.1. We will discuss the potential issue related to it in the following chapter.

From the three theorems above, we observe that distribution discrepancies between seen and unseen domains are quantified using the $\mathcal{H}$-divergence or its variation. This theoretical foundation has gained wide acceptance and has consequently spurred the development of adversarial algorithms [36, 120, 147]. However, as noted in [118], this adversarial framework can encounter gradient vanishing problems, especially when the domain discriminator can accurately discriminate between the representations of the seen and unseen domains. To mitigate this practical issue, Shen *et al.* introduce an upper bound for the risk in unseen domains, substituting the $\mathcal{H}$-divergence with Wasserstein-1 distance [37, 105].

Theorem 2.2.4. (Theorem 1 in [118]). *Let $\mathcal{H}$ be a hypothesis class such that $h \in \mathcal{H}$ are all K-Lipschitz continuous for some K, then*

$$R^{(u)}(h) \leq R^{(s)}(h) + 2K\mathsf{W}_1(\mu^{(u)}, \mu^{(s)}) + \lambda \tag{2.19}$$

where $\lambda = \inf_{h \in \mathcal{H}} \left(R^{(s)}(h) + R^{(u)}(h) \right)$

Here, a new metric, Wasserstein-1 distance [37, 105, 115] first comes to the stage. We provide a brief overview of the definition of the Wasserstein distance in Appendix A. The authors then propose a theory-driven algorithm called Wasserstein Distance Guided Representation Learning (WDGRL). This algorithm aims to minimize the classification loss and maximize the dual form of the Wasserstein-1 distance between the extracted representations of both seen and unseen domains, simultaneously. Al-

though it also employs an adversarial training scheme in practice, the objective differs. Specifically, the adversarial training part does not aim at fooling the domain discriminator, but rather at maximizing the dual form of the Wasserstein-1 distance, which is equivalent to minimizing the Wasserstein-1 distance between the representation distributions. This ensures that the representations from seen and unseen domains are aligned, thereby achieving "domain-alignment".

The theoretical work above do not pose explicit assumptions on the distribution of seen and unseen domains. We then introduce some works that requires specific relationships to hold on the distributions of the seen and unseen domains.

Johansson *et al.* [68] focus on a special scenario of the DA problem known as the covariate shift. Under this setting, the conditional distribution of both seen and unseen domains are assumed to be stable, *i.e.*, $p^{(s)}(Y|X) = p^{(u)}(Y|X)$, while the marginal distribution of the data $p^{(s)}(X)$, differs from $p^{(u)}(X)$. Specifically, their bound includes three terms: the first term is a weighted risk on the seen domain; the second term is the support sufficiency divergence that measures the lack of overlapping support between seen and unseen domains; and the third term quantified the loss caused by the non-invertible representation function f.

Approaching from a different perspective, Combes and Zhao *et al.* [126] introduce a generalized label shift (GLS) assumption, where they assume $p^{(s)}(f(X)|Y) = p^{(u)}(f(X)|Y)$ while $p^{(s)}(Y) \neq p^{(u)}(Y)$. With such an assumption for a k class classification problem, the joint risk of seen and unseen domains is bounded by $2 \times \max_{j \in [k]} p^{(s)}(g(f(X)) \neq Y|Y = j)$. They further extende their analysis to scenarios without the GLS assumption, bounding the risk of the unseen domain based on the risk in the seen domain, the weighted L^1 distance of the marginal label distributions between seen and unseen domains, and the maximum of $|p^{(s)}(g(f(X)) = y'|Y = y) - p^{(u)}(g(f(X)) = y'|Y = y)|$ for all $y' \neq y$.

The aforementioned theoretical studies have laid the foundation for DA algorithms

and significantly influenced the development of DG. Numerous DG works [85,86,96,117] derive insights from these theories, seeking a representation function f that can extract domain-invariant yet task-sensitive representations [42] from seen domains. With such a function f, models are better equipped to reliably handle unseen test domains. These representation learning-based studies constitute a central vein of DG research. In subsequent sections, while we delve into various DG solutions, our core focus will remain on the domain-invariant representation learning method.

2.3 Domain Generalization

As the rapid development of the DG research, a plethora of algorithms have emerged. Most of these algorithms aim to solve the DG problem from three different directions: data manipulation, representation learning, and learning strategy [137, 152]. In this section, we will mainly review the DG work from these three perspectives.

2.3.1 Data Manipulation

The performance of a model often depends heavily on the quantity and diversity of the training data. Given a limited set of data, data manipulation offers a cost-effective way to generate more samples with controllable diversity [137]. Though commonly incorporated in deep neural network training to mitigate overfitting [53], in the DG context, data manipulation is developed as a distinct approach to address the DG problem. Following [137], we abstract the objective of data manipulating methods as follow:

$$\underset{h\in\mathcal{H}}{\operatorname{argmin}} \frac{1}{S}\sum_{s=1}^{S}\sum_{i=1}^{N^s} \ell(h(\boldsymbol{x}_i^{(s)}), y_i^{(s)}) + \ell(h(\hat{\boldsymbol{x}}_i^{(s)}), y_i^{(s)}) \tag{2.20}$$

where $\hat{\boldsymbol{x}}_i^{(s)}$ is the transformed data of original data $\boldsymbol{x}_i^{(s)}$ and for simplicity, we replace $h^{(s)}(\boldsymbol{x}_i^{(s)})$ with $y_i^{(s)}$. Transformations here are not limited to standard operations like rotation, flipping, color jittering, resizing, *etc.*, but are usually more task specific. Two

main methods will be reviewed here, namely data augmentation and data generation [137].

Data augmentation is widely adopted by machine learning researchers, especially in the field of computer vision. Besides the commonly used augmentation techniques mentioned above, domain randomization has recently been adopted to address the DG problem. For example, in [128], the authors address the object localization problem by generating more diverse data with randomized numbers, positions, textures of objects and surrounding lights from a simulator. Similar techniques are further explore by [71, 144], where the authors randomly map the labeled synthetic images to multiple real-world domains with different style. Domain randomization is powerful technique for increasing the diversity of the training dataset and reduce the gap between the synthetic data and the real-world data. However, due to its randomness, we may lose control of the quality of the augmented data, leading to some useless or repeated augmentations [137]. Adversarial data augmentation is another widely used method, where the input data from seen domains is modified to follows some fictitious distributions guided by an adversarial loss. Following this idea, Volpi *et al.* [134] propose to iteratively augment the training data such that its augmentation is challenging for the current model to discern. Leveraging a domain transformation network (DoTNet), Zhou *et al.* [153] augment the training data such that a domain classifier is unable to tell which domain the augmented images are from. Motivated by the Information Bottleneck (IB) [127] principle, Zhao *et al.* [148] propose to generate "hard" adversarial perturbations by maximizing the IB function, which will ensure a large distribution shift between augmentations and the original input.

The other branch taking advantage of the powerful Generative Adversarial Network (GAN) [55]. Through GAN, diverse data can be easily generated, paving the road for data generation techniques in DG. For example, [154] employs the Learning to Augment by Optimal Transport strategy to generate pseudo-novel domain data based

on the data from seen domain conditioned on the label. [122] uses AdaIN [61] to generate images with varying texture. Other methods diverge from the conventional deep neural network data generation approach. For instance, Mixup [145] proposes to directly generate new samples by a linear interpolation between samples and their corresponding labels from the seen domain. Sharing the same idea, [141, 156] generate new representations instead of data in the representation space, further reducing the computational cost while maintaining good generalization performance.

2.3.2 Representation Learning

As the most popular category in DG, a great amount of work contributes the DG development through the lens of representation learning. Most of the DG studies within this category draw inspiration from the theoretical results from DA [21, 22, 118, 146]. In this part, we will focus on domain-invariant representation learning and representation disentanglement approach.

2.3.2.1 Domain-invariant Representation Learning

As noted in Section 2.2, the concept of learning domain-invariant representation has its origins in the DA problem. Motivated by the theoretical works in DA, which indicating that domain-invariant representations are transferable across domains, DG algorithms encourage the representation function f to extract representations that contain less domain-specific but more domain-invariant and task-related information. This strategy ensures the model remains robust against distribution shifts in the training data, such that it can also generalize to the unseen test data. Embracing a framework analogous to DA, DG algorithms in this category typically decompose the hypothesis into a representation function f and a labeling function g. The abstracted

objective can be viewed as follows:

$$\underset{f\in\mathcal{F},g\in\mathcal{G}}{\operatorname{argmin}} \frac{1}{S}\sum_{s=1}^{S}\sum_{i=1}^{N^s}\ell(g(f(\boldsymbol{x}_i^{(s)}),y_i^{(s)}))) + \alpha \mathsf{L}_{reg}(f) \tag{2.21}$$

where $\mathsf{L}_{reg}(f)$ is the regularization loss designed for the representation function f. Here we mainly review three types algorithms: adversarial-based approaches, representation alignment approaches, and invariant risk minimization-based approaches.

At the early phase of DG evolution, the idea of domain adversarial neural network (DANN) [51] is widely adapted and extensively integrated into DG applications [86, 92, 117]. To fit DG settings, the domain discriminator's input transitions from test data to data from multiple seen domains. Variations involved transforming the domain discriminator into a binary classifier for each domain [117] and deploying a single neural network for multi-domain classification [7, 16]. Subsequently, authors of [126] introduce a conditional invariant adversarial network for extracting class-wise domain-invariant representations. Building on this, [149] leverages an entropy term to measure the dependency between the extracted representation and the label, mitigating the demand for numerous domain discriminators as class numbers grow.

Contrary to the adversarial approach, which often requires one or multiple auxiliary neural networks as domain discriminators, a more direct and simple method is to align the representation distributions across different seen domains directly. The early work like [95] does not work on the distribution level but on samples. It introduces contrastive loss to minimize the distance between samples of the same class from different domains while maximizing the distance between samples of different classes from different domains. Subsequent studies introduce various metrics to measure the discrepancy between representation distributions across domains. For example, [89] and [150] use Wasserstein distance [133], [136] and [85] adopt the maximum mean discrepancy (MMD), [125] and [124] match the second-order moment of representations.

Besides distribution alignment, invariant risk minimization (IRM) [12] provides a innovative angle to tackle the DG problem, taking spurious features into account. In their work [12], Arjovsky *et al.* aim to find a representation function f that elicits an invariant predictor $g \circ f$ across all seen domains, meaning that the optimal linear classifiers, once applied to representations, remain optimal for all domains [4], to get rid of the spurious features. Variations based on the IRM framework are introduced in [29], where information theory is integrated into the IRM framework. Subsequent work, as seen in [6, 111], expands the analysis of IRM to classification problems and non-linear scenarios, pointing out limitations of the original IRM method in these contexts. Building on this, [4] incorporates information bottleneck principle to overcome these constraints, further broadening the utility of IRM-based techniques.

2.3.2.2 Representation Disentanglement

Another prominent approach within DG's representation learning landscape is the representation disentanglement method.

Differing from domain-invariant representation learning, representation disentanglement imposes a more relaxed constraint on the extracted representations, allowing the coexistence of domain-specific information. This approach aim to design a representation function f, in such a way that domain-specific and domain-invariant information occupy different dimensions or segments within extracted features [137]. In the application phase, domain-invariant information is either combined with the domain-specific features [30] or utilized independently [106]. Methods rooted in this principle can be further categorized into two groups, based on the specific disentangling techniques employed. One includes directly decomposing the models weight [106] or adding a structured low-rank constraint [41]. The other adopts the generative models like variational autoencoder to disentangle the extracted features into domain-specific and domain-invariant features [33, 65, 99].

2.3.3 Learning Strategy

In addition to theory-guided DG algorithms, there exists another branch of solutions that employ diverse learning strategies, which are not originally designed for DG but effectively adapted by researchers to address the DG problem. These strategies include meta-learning, where models are trained to iteratively adapt to meta-test domains that are created from the current seen domains [19, 83, 84, 87, 108]; ensemble learning, where either multiple models collaborate to enhance performance [45, 155] or weights of multiple models are averaged to reach the flat minima of the risk [15, 28, 66]; gradient-operation based strategies, where gradients are manipulated to the direction for more robust features [62, 109, 119]; and the self-supervised learning strategy, where models are trained using their own generated supervisory signals [27, 73]. Each of these methodologies offers unique perspectives and solutions, thereby expanding the horizons of DG research.

Chapter 3

Domain Generalization Theory: A Revisit and Refinement

Chapter 3 comprises two part. The first part presents our primary contributions to Domain Generalization (DG) theory. In the second part, we offer a theoretical justification for the use of the reconstruction loss employed in the first part, from an alternative perspective of the DG problem.

3.1 Part I: Barycentric-Alignment and Reconstruction Loss Minimization for Domain Generalization

In this section, we present our main theoretical result for Domain Generalization (DG). Specifically, we consider the typical DG setting where the hypothesis is composed of a representation mapping followed by a labeling function. Note that within this setting, the majority of popular DG methods aim to jointly learn the representation and the labeling functions by minimizing the well-known upper bounds [21, 22, 118] introduced in Chapter 2. However, in practice, methods based on these theoretical upper bounds

either ignore a term that cannot be directly optimized, or approximately apply the bound to the feature space. To bridge this gap between theory and practice, we introduce a new upper bound, resulting in a fully optimizable upper bound for the risk of unseen domain. Our derivation leverages classical and recent transport inequalities that link optimal transport metrics with information-theoretic measures. Compared to previous bounds, our bound introduces two new terms: (i) the Wasserstein-2 barycenter term that aligns distributions between domains, and (ii) the reconstruction loss term that assesses the quality of representation in reconstructing the original data. Based on this new upper bound, we propose a novel DG algorithm named Wasserstein Barycenter Auto-Encoder (WBAE) that simultaneously minimizes the classification loss, the barycenter loss, and the reconstruction loss.

3.1.1 Introduction

As introduced in Chapter 2, many representation learning-based DG work decomposes the hypothesis as a representation function followed by a labeling function [9,42,85,150], and optimizes both jointly by minimizing an upper bound for the classification risk in the unseen domain derived in [22] (shown as 2.2.1). The upper bound consists of three terms: (1) the prediction risk on the mixture of seen domains, (2) the discrepancy or divergence between the data distributions of different domains in the representation space, and (3) a *combined risk* across all domains that implicitly depends on both the representation mapping and the unknown optimal labeling function from the unseen domain. However, most current approaches disregard this dual dependency and treat the third term (*combined risk*) as a constant while developing their algorithms. In fact, the majority of prominent works in DG and DA such as [51, 85, 146] are essentially variations of the following strategy: ignore the *combined risk* term and learn a domain-invariant representation mapping *or* align the domains in the representation space, together with learning a common labeling function controlling the prediction loss

across the seen domains. However, the *combined risk* term is, in fact, a function of the representation mapping and should somehow be accounted for within the optimization process, as we discussed in Chapter 2.

To address these limitations, we revisit the analysis in [21, 22] and derive a new upper bound that is free of terms with the dual dependence mentioned above. Our new bound consists of four terms: (1) the prediction risk across seen domains in the input space; (2) the discrepancy/divergence between the induced distributions of seen and unseen domains in the representation space, which can be approximated via the Wasserstein-2 barycenter [115] of seen domains; (3) the reconstruction loss term that measures how well the input can be reconstructed from its representation; and (4) a combined risk term that is independent of the representation mapping and labeling function to be learned. Our new bound differs from previous ones in two aspects. Firstly, it introduces two new terms: (a) the Wasserstein-2 barycenter term for domain alignment and (b) the reconstruction loss term for assessing the quality of representation in reconstructing the original data. We note that the Wasserstein-2 barycenter term for controlling the domain discrepancy in our bound is built in the representation space, which is better aligned with the practical implementation than previous Wasserstein-based bounds [118] (also shown as 2.2.4), which are built in the data space. Secondly, the combined risk in our bound is independent of the representation mapping and thus can be ignored during the optimization. Motivated by these theoretical results, we propose an Auto-Encoder-based model that interacts with the Wasserstein barycenter loss to achieve domain alignment.

3.1.2 Contributions

The contributions of this work can summarized as follows:

1. *Contributions to Theory*: We propose a new upper bound for the risk of the unseen domain using classical and recent transport inequalities that link optimal

transport metrics with information-theoretic measures. All terms in our new upper bound are optimizable in practice which overcomes the limitations of previous works and bridges the gap between previous theory and practice.

2. *Contributions to Algorithm Development and Practice*: We develop a novel algorithm for domain generalization based on our new upper bound. Our algorithm optimizes a new term that controls the domain discrepancy through Wasserstein-2 barycenter. Unlike previous Wasserstein distance-based bounds that form the domain discrepancy term in the data space but optimize it in the representation space, our domain discrepancy term is constructed and optimized in the representation space, making our practical implementation better aligned with the theory.

3. *Gains over state-of-the-art methods*: Our algorithm consistently outperforms other theory-guided methods on PACS, VLCS, Office-Home, and TerraIncognita datasets, with a noticeable improvement of $1.7-2.8$ percentage points on average across all datasets.

3.1.3 Related Work

This work falls within the DG framework wherein domain-invariant features are learned by decomposing the prediction function into a representation mapping followed by a labeling function. A recent example of this framework is [9], where the authors propose a three-part model consisting of a feature extractor, a classifier, and domain discriminators. The feature extractor learns the task-sensitive, but domain-invariant features via minimizing the cross-entropy loss with respect to the task label and maximizing the sum of domain discriminator losses. The domain discriminator loss is based on an estimate of the $\mathcal{H}$-divergence between all seen domains [21] and has roots in the works [51, 86] on Domain Adaptation. Following a similar idea,

the authors of [85] align the representation distributions from different domains by minimizing their Maximum Mean Discrepancy. In [42], the authors adopt a gradient-based episodic training scheme for DG in which the extracted features are driven to simultaneously preserve global class information and local task-related clusters across seen domains by minimizing an alignment loss comprising soft class confusion matrices and a contrastive loss. In [99], DG is achieved by disentangling style variation across domains from learned features. Among the large body of works on the DG problem, we regard [12, 23, 51, 86], and [78] as recent exemplars of principled algorithms that are guided by theory and compare their performance with our algorithm's.

Our proposed upper bound is based on the Wasserstein barycenters. Related to this context are the works [110, 118], and [150]. In [150], the pairwise Wasserstein-1 distance [105, 115], is used as a measure of domain discrepancy. Using the dual form of the Wasserstein-1 distance, the feature extractor in [150] minimizes a combination of cross-entropy loss, Wasserstein distance loss, and a contrastive loss to achieve DG. The works [110, 118] provide upper bounds for the risk of unseen domain based on the Wasserstein-1 distance. While sharing some similarities with ours, their bounds are constructed in the input space and therefore do not explicitly motivate the use of representation functions. By contrast, our proposed upper bound measures the discrepancy of domains in the representation space, which naturally justifies the decomposition of the hypothesis in the practical implementation. A detailed analysis and comparison of the bounds in [110, 118] and our proposed bound can be found in Section 3.1.4.2.

In addition to the domain-invariant feature learning approach, which is the main focus of this paper, there are other noteworthy and emerging directions in the DG research. For more details, we refer the reader to Chapter 2.

3.1.4 Theoretical Analysis

Here we restate the notations employed throughout this study as a helpful reminder. We consider a *domain* v as a triple $(\mu^{(v)}, f^{(v)}, g^{(v)})$ consisting of a distribution $\mu^{(v)}$ on the input $\boldsymbol{x} \in \mathbb{R}^d$, a representation function $f^{(v)} : \mathbb{R}^d \to \mathbb{R}^{d'}$, from the input space to the representation space, and a stochastic labeling function $g^{(v)} : \mathbb{R}^{d'} \to \mathcal{Y}$ from the representation space to the label space. We denote the unseen domain by $(\mu^{(u)}, f^{(u)}, g^{(u)})$ and S seen domains by $(\mu^{(s)}, f^{(s)}, g^{(s)})$, with $s = 1, \ldots, S$.

Let $\mathcal{F} = \{f | f : \mathbb{R}^d \to \mathbb{R}^{d'}\}$ be the set of *representation functions*, $\mathcal{G} = \{g | g : \mathbb{R}^{d'} \to \mathcal{Y}\}$ the set of stochastic *labeling functions*, $\mathcal{H} := \mathcal{G} \circ \mathcal{F}$ the set of *hypotheses*, with each hypothesis $h : \mathbb{R}^d \to \mathcal{Y}$ obtained by composing a $g \in \mathcal{G}$ with an $f \in \mathcal{F}$, *i.e.*, $h = g \circ f$, and $\Psi = \{\psi | \psi : \mathbb{R}^{d'} \to \mathbb{R}^d\}$ the set of *reconstruction functions* that map from the representation space back to the input space. Here, we limit our theoretical study to binary classification problems, specifically hypothesis functions h such that $h : \mathbb{R}^d \to \mathcal{Y} = [0, 1]$. Note that a similar set-up is also used in [22] where the hypothesis h occurs non-deterministically and maps a data point to a label between zero and one.

The risk of using a hypothesis h in domain v is then defined by:

$$R^{(v)}(h) := \mathbb{E}_{\boldsymbol{x} \sim \mu^{(v)}} \left[\ell(h(\boldsymbol{x}), h^{(v)}(\boldsymbol{x})) \right], \tag{3.1}$$

where $\mathbb{E}[\cdot]$ denotes the expectation, $h^{(v)} = g^{(v)} \circ f^{(v)}$, and $\ell(\cdot, \cdot)$ is a loss function. We make the following assumptions:

A1: The loss function $\ell(\cdot, \cdot)$ is non-negative, symmetric, bounded by a finite positive number L, satisfies the triangle inequality, and Q-Lipschitz continuous, *i.e.*, for any three scalars a, b, c and positive constant Q,

$$|\ell(a, b) - \ell(a, c)| \leq Q \ |b - c|. \tag{3.2}$$

A2: The optimal hypothesis of the unseen domain $h^{(u)} = g^{(u)} \circ f^{(u)}$ is K-Lipschitz continuous. Specifically, we assume that for any two vectors $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$, and positive constant K,

$$|h^{(u)}(\boldsymbol{x}) - h^{(u)}(\boldsymbol{x}')| \leq K \ \|\boldsymbol{x} - \boldsymbol{x}'\|_2, \tag{3.3}$$

where $\|\boldsymbol{x} - \boldsymbol{x}'\|_2$ denotes the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$.

The first four conditions in Assumption A1 can be easily satisfied by any metric or norm truncated by a finite positive number. Concretely, if $d(a, b)$ is a metric, potentially unbounded like Mean Squared Error (MSE), then $loss(a, b) := \min(L, d(a, b))$, where L is a positive constant, will satisfy the first four conditions in A1. The Lipschitz condition in A1 and A2 are also widely used in the theory and practice of DG [23, 118, 139].

One may find our assumptions bear some similarities with the assumptions in [110] and [118], but there are some fundamental differences. Specifically, we assume that the loss function is non-negative, symmetric, bounded, Lipschitz, and satisfies the triangle inequality, whereas the loss function in [110] is required to be convex, symmetric, bounded, obey the triangle inequality, and satisfy a specific form. We only assume that the optimal hypothesis function on the unseen domain is Lipschitz, whereas [118] requires all hypotheses to be Lipschitz.

3.1.4.1 Bound for Unseen Domain Risk

Our analysis starts by considering a single seen domain. Lemma 3.1.1 below upper bounds the risk $R^{(u)}(h)$ of a hypothesis $h = g \circ f$ in the unseen domain u by four terms: (1) the risk of the seen domain s, (2) the L^1 distance between the distributions of the *data representations* from the seen and unseen domain, (3) the reconstruction loss that quantifies how well the representation can reconstruct its original data input, and (4) an intrinsic risk term that is free of h and is intrinsic to the domains and the

loss function. We use the notation $f_{\#}\mu^{(v)}$ to denote the pushforward of distribution $\mu^{(v)}$ under the representation function f, *i.e.*, the distribution of $f(\boldsymbol{x})$ with $\boldsymbol{x} \sim \mu^{(v)}$.

Lemma 3.1.1. *Under assumptions A1 and A2, for any hypothesis $h \in \mathcal{H}$ and any reconstruction function $\psi \in \Psi$, the following bound holds:*

$$R^{(u)}(h) \leq R^{(s)}(h) + L \, \|f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}\|_1 + QK\Big(\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big] + \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big]\Big) + \sigma^{(u,s)}$$

where $\|f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}\|_1 = \int_{\boldsymbol{z}} |f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}| d\boldsymbol{z}$ denotes the L^1 distance between $(f_{\#}\mu^{(u)}, f_{\#}\mu^{(s)})$ in the representation space and:

$$\sigma^{(u,s)} := \min\{\ \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\ell(h^{(u)}(\boldsymbol{x}), h^{(s)}(\boldsymbol{x}))\big], \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\ell(h^{(u)}(\boldsymbol{x}), h^{(s)}(\boldsymbol{x}))\big]\ \}.$$

Proof. Note that in this work, we assume that any hypothesis function $h(\cdot)$ outputs a value in $[0, 1]$, *i.e.*, $h : \mathbb{R}^d \to [0, 1]$, and $\ell(\cdot, \cdot)$ is a bounded distance metric. In addition, we assume that $h^{(u)}(\cdot)$ is K-Lipschitz continuous and $\ell(\cdot)$ is Q-Lipschitz continuous. Particularly, we assume that for any two vectors $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ and any three scalars a, b, and c, the following inequalities hold:

$$|h^{(u)}(\boldsymbol{x}) - h^{(u)}(\boldsymbol{x}')| \leq K\|\boldsymbol{x} - \boldsymbol{x}'\|_2, \tag{3.4}$$

$$|\ell(a, b) - \ell(a, c)| \leq Q|b - c|, \tag{3.5}$$

where $\|\boldsymbol{x} - \boldsymbol{x}'\|_2$ and $|b - c|$ denote the Euclidean distances between $\boldsymbol{x}$ and $\boldsymbol{x}'$, and b and c, respectively.

We want to note that our approach is motivated by the proof of Theorem 1 in [21]. To better demonstrate the relationship between the hypothesis, input distribution, true representation and labeling functions, we use inner product notation $\langle \cdot, \cdot \rangle$ to

denote expectations. Specifically,

$$R^{(v)}(h) := \mathbb{E}_{\boldsymbol{x}\sim\mu^{(v)}}\left[\ell(h(\boldsymbol{x}), h^{(v)}(\boldsymbol{x}))\right] = \langle \ell(h, h^{(v)}), \mu^{(v)} \rangle. \tag{3.6}$$

From the definition of risk,

$$\begin{aligned}
R^{(u)}(h) &= \langle \ell(h, h^{(u)}), \mu^{(u)} \rangle \\
&= \langle \ell(h, h^{(s)}), \mu^{(s)} \rangle - \langle \ell(h, h^{(s)}), \mu^{(s)} \rangle + \langle \ell(h, h^{(u)}), \mu^{(u)} \rangle \\
&= R^{(s)}(h) + \left(\langle \ell(h, h^{(u)}), \mu^{(u)} \rangle - \langle \ell(h, h^{(s)}), \mu^{(u)} \rangle \right) \\
&+ \left(\langle \ell(h, h^{(s)}), \mu^{(u)} \rangle - \langle \ell(h, h^{(s)}), \mu^{(s)} \rangle \right) \\
&\leq R^{(s)}(h) + \langle \ell(h^{(u)}, h^{(s)}), \mu^{(u)} \rangle + \langle \ell(h, h^{(s)}), \mu^{(u)} - \mu^{(s)} \rangle
\end{aligned} \tag{3.7}$$

where the inequality of (3.7) follows from the triangle inequality $\ell(h, h^{(u)}) \leq \ell(h, h^{(s)}) + \ell(h^{(s)}, h^{(u)})$ and $\ell(h^{(s)}, h^{(u)}) = \ell(h^{(u)}, h^{(s)})$. In an analogous fashion, it is possible to show that:

$$R^{(u)}(h) \leq R^{(s)}(h) + \langle \ell(h^{(u)}, h^{(s)}), \mu^{(s)} \rangle + \langle \ell(h, h^{(u)}), \mu^{(u)} - \mu^{(s)} \rangle. \tag{3.8}$$

Next, we will bound the third term in the right-hand-side of (3.8). Specifically,

$$
\begin{aligned}
&\langle \ell(h, h^{(u)}), \mu^{(u)} - \mu^{(s)} \rangle \\
&= \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\Big[\ell\big(h(\boldsymbol{x}), h^{(u)}(\boldsymbol{x})\big)\Big] - \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\Big[\ell\big(h(\boldsymbol{x}), h^{(u)}(\boldsymbol{x})\big)\Big] \\
&\leq \max\Big\{\mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\Big[\ell\big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x}))) + K\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big)\Big], \\
&\qquad\qquad \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\Big[\ell\big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x}))) - K\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big)\Big]\Big\} \\
&- \min\Big\{\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\Big[\ell\big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x}))) + K\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big)\Big], \\
&\qquad\qquad \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\Big[\ell\big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x}))) - K\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big)\Big]\Big\} \qquad (3.9) \\
&\leq \Big(\mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\Big[\ell\big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x})))\big)\Big] + \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\Big[QK\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\Big]\Big) \\
&\qquad\qquad - \Big(\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\Big[\ell\big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x})))\big)\Big] - \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\Big[QK\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\Big]\Big) \qquad (3.10) \\
&= \Big(\mathbb{E}_{\boldsymbol{z}\sim f_{\#}\mu^{(u)}}\Big[\ell\big(g(\boldsymbol{z}), h^{(u)}(\psi(\boldsymbol{z}))\big)\Big] - \mathbb{E}_{\boldsymbol{z}\sim f_{\#}\mu^{(s)}}\Big[\ell\big(g(\boldsymbol{z}), h^{(u)}(\psi(\boldsymbol{z}))\big)\Big]\Big) \\
&\qquad\qquad + \Big(\mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\Big[QK\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\Big] + \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\Big[QK\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\Big]\Big) \qquad (3.11) \\
&= \langle \ell\big(g(\boldsymbol{z}), h^{(u)}(\psi(\boldsymbol{z}))\big), f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)} \rangle \\
&\qquad\qquad + QK\Big(\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big] + \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big]\Big) \\
&\leq L\langle 1, |f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}| \rangle + QK\Big(\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big] + \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big]\Big). \\
& \qquad\qquad (3.12)
\end{aligned}
$$

(3.10) is due to the Lipschitzness of $\ell(\cdot)$:

$$
\begin{aligned}
&\max\Big\{\ell\Big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x}))) + K\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\Big), \ell\Big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x}))) - K\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\Big)\Big\} \\
&\quad \leq \ell\Big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x})))\Big) + QK\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2, \qquad (3.13)
\end{aligned}
$$

$$
\begin{aligned}
&\min\Big\{\ell\Big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x}))) + K\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\Big), \ell\Big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x}))) - K\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\Big)\Big\} \\
&\quad \geq \ell\Big(h(\boldsymbol{x}), h^{(u)}(\psi(f(\boldsymbol{x})))\Big) - QK\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2. \qquad (3.14)
\end{aligned}
$$

Finally, we get (3.11) due to $h = g \circ f$, $f(\boldsymbol{x}) = \boldsymbol{z}$, and (3.12) due to $\ell(\cdot, \cdot)$ is

bounded by L.

The proof of Lemma 3.1.1 follows by combining (3.7), (3.8), (3.12), and note that:

$$\sigma^{(u,s)} = \min\Big\{\langle \ell(h^{(u)}, h^{(s)}), \mu^{(u)}\rangle, \langle \ell(h^{(u)}, h^{(s)}), \mu^{(s)}\rangle\Big\},$$

and

$$\langle 1, |f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}|\rangle = \|f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}\|_1.$$

□

In typical DG applications, training data from multiple seen domains are available and can be combined in various ways. Therefore, Lemma 3.1.2 below extends Lemma 3.1.1 to a convex combination of distributions of multiple seen domains.

Lemma 3.1.2. *For any convex weights $\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(S)}$ (non-negative and summing to one), any reconstruction function $\psi \in \Psi$, and any hypothesis $h \in \mathcal{H}$, the following bound holds:*

$$\begin{aligned} R^{(u)}(h) \leq & \sum_{s=1}^{S} \lambda^{(s)} R^{(s)}(h) \\ & + L \sum_{s=1}^{S} \lambda^{(s)} \|f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}\|_1 \\ & + QK\Big(\sum_{s=1}^{S} \lambda^{(s)} \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big] + \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big]\Big) \\ & + \sum_{s=1}^{S} \lambda^{(s)} \sigma^{(u,s)}. \end{aligned}$$

Proof. Apply Lemma 3.1.1 S times for S seen domains, then for any hypothesis $h \in \mathcal{H}$

and function (decoder) $\psi : \mathbb{R}^{d'} \to \mathbb{R}^d$, the following bound holds:

$$\begin{aligned} R^{(u)}(h) \leq R^{(s)}(h) + L \, \|f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}\|_1 \\ + QK\Big(\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big] \\ + \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big]\Big) + \sigma^{(u,s)}, \forall s = 1, \ldots, S. \end{aligned} \tag{3.15}$$

Next, multiplying (3.15) with its corresponding convex weight $\lambda^{(s)}$, for $s = 1, 2, \ldots, S$, and summing them up, we have:

$$\begin{aligned} \sum_{s=1}^{S} \lambda^{(s)} R^{(u)}(h) \leq \sum_{s=1}^{S} \lambda^{(s)} \Bigg[R^{(s)}(h) + L \, \|f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}\|_1 + QK\Big(\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big] \\ + \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big]\Big) + \sigma^{(u,s)} \Bigg]. \end{aligned} \tag{3.16}$$

Note that $\sum_{s=1}^{S} \lambda^{(i)} = 1$, thus, the left-hand side of (3.16) is $R^{(u)}(h)$, and by re-arranging the terms on the right-hand side of (3.16), the proof follows. □

The upper bound above relies on the L^1 distances between the pushforwards of seen and unseen distributions. However, accurately estimating L^1 distances from samples is hard [21, 72]. To overcome this practical limitation, we upper bound the L^1 distance by the Wasserstein-2 distance under additional regularity assumptions on the pushforward distributions.

Definition 3.1.3. [107] A probability distribution on $\mathbb{R}^d$ is called (c_1, c_2)-regular, with $c_1, c_2 \geq 0$, if it is absolutely continuous with respect to the Lebesgue measure with a differentiable density $p(\boldsymbol{x})$ such that

$$\forall \boldsymbol{x} \in \mathbb{R}^d, \quad \|\nabla \log_2 p(\boldsymbol{x})\|_2 \leq c_1 \|\boldsymbol{x}\|_2 + c_2,$$

where ∇ denotes the gradient and $\|\cdot\|_2$ denotes the Euclidean norm.

Lemma 3.1.4. *If μ and ν are (c_1, c_2)-regular, then:*

$$\|\mu - \nu\|_1 \leq \sqrt{c_1\Big(\sqrt{\mathbb{E}_{\boldsymbol{u}\sim\mu}\big[\|\boldsymbol{u}\|_2^2\big]} + \sqrt{\mathbb{E}_{\boldsymbol{v}\sim\nu}\big[\|\boldsymbol{v}\|_2^2\big]}\Big) + 2c_2} \times \sqrt{\mathsf{W}_2(\mu,\nu)}$$

where the Wasserstein-p metric [105, 115] $\mathsf{W}_p(\mu,\nu)$ *is defined as,*

$$\mathsf{W}_p(\mu,\nu) := (\inf_{\pi\in\Pi(\mu,\nu)} \mathbb{E}_{(\boldsymbol{u},\boldsymbol{v})\sim\pi}[\|\boldsymbol{u}-\boldsymbol{v}\|_2^p])^{1/p}$$

where $\Pi(\mu,\nu)$ is the set of joint distributions with marginals μ and ν.

Proof. From Pinsker's inequality [35], the L^1 distance can be bounded by Kullback–Leibler (KL) divergence as follows:

$$\|\mu - \nu\|_1^2 \leq 2d_{KL}(\mu,\nu) \tag{3.17}$$

where $\|\mu-\nu\|_1$ and $d_{KL}(\mu,\nu)$ denote L^1 distance and Kullback–Leibler divergence between two distributions μ and ν, respectively. Since $\|\mu-\nu\|_1 = \|\nu-\mu\|_1$, applying Pinsker's inequality to (μ,ν) and (ν,μ),

$$2\|\mu-\nu\|_1^2 = \|\mu-\nu\|_1^2 + \|\nu-\mu\|_1^2 \leq 2d_{KL}(\mu,\nu) + 2d_{KL}(\nu,\mu) \tag{3.18}$$

which is equivalent to,

$$\|\mu-\nu\|_1 \leq \sqrt{d_{KL}(\mu,\nu) + d_{KL}(\nu,\mu)}. \tag{3.19}$$

Next, if μ and ν are (c_1, c_2)-regular distributions, their KL divergences can be bounded by their Wasserstein-2 distance as follows (please see equation (10), Proposi-

tion 1 in [107]),

$$d_{KL}(\mu,\nu)+d_{KL}(\nu,\mu)\leq 2\mathsf{W}_2(\mu,\nu)\Big(\frac{c_1}{2}\sqrt{\mathbb{E}_{\boldsymbol{u}\sim\mu}\big[\|\boldsymbol{u}\|_2^2\big]}+\frac{c_1}{2}\sqrt{\mathbb{E}_{\boldsymbol{v}\sim\nu}\big[\|\boldsymbol{v}\|_2^2\big]}+c_2\Big). \tag{3.20}$$

Combining (3.19) and (3.20), we have:

$$\|\mu-\nu\|_1\leq\big[\mathsf{W}_2(\mu,\nu)\big]^{1/2}\sqrt{c_1\Big(\sqrt{\mathbb{E}_{\boldsymbol{u}\sim\mu}\big[\|\boldsymbol{u}\|_2^2\big]}+\sqrt{\mathbb{E}_{\boldsymbol{v}\sim\nu}\big[\|\boldsymbol{v}\|_2^2\big]}\Big)+2c_2}. \tag{3.21}$$

□

One may wonder what conditions would guarantee the regularity of the pushforward distributions. Proposition 2 and Proposition 3 in [107] show that any distribution ν for which $\mathbb{E}_{\boldsymbol{v}\sim\nu}\|\boldsymbol{v}\|_2$ is finite becomes regular when convolved with any regular distribution, including the Gaussian distribution. Since convolution of distributions corresponds to the addition of independent random vectors having those distributions, it is always possible to make the pushforwards regular by adding a small amount of independent spherical Gaussian noise in the representation space.

Combining Lemma 3.1.2, Lemma 3.1.4, and applying Jensen's inequality, we obtain our main result:

Theorem 3.1.5. *If $f_{\#}\mu^{(s)}$, $s=1,2,\ldots,S$, and $f_{\#}\mu^{(u)}$ are all (c_1,c_2)-regular, then for any convex weights $\lambda^{(1)},\lambda^{(2)},\ldots,\lambda^{(S)}$, any reconstruction function $\psi\in\Psi$, and any*

hypothesis $h \in \mathcal{H}$, the following bound holds:

$$
\begin{aligned}
R^{(u)}(h) \leq & \sum_{s=1}^{S} \lambda^{(s)} R^{(s)}(h) \\
& + LC\big[\sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu^{(u)}, f_{\#}\mu^{(s)})\big]^{1/4} \\
& + QK\Big(\sum_{s=1}^{S} \lambda^{(s)} \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big] + \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\big]\Big) \\
& + \sum_{s=1}^{S} \lambda^{(s)} \sigma^{(u,s)}
\end{aligned} \tag{3.22}
$$

where:

$$
C = \max_s \sqrt{c_1\Big(\sqrt{\mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\|f(\boldsymbol{x})\|^2\big]} + \sqrt{\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\|f(\boldsymbol{x})\|^2\big]}\Big) + 2c_2}.
$$

Proof. Under the assumption that $f_{\#}\mu^{(s)}$ and $f_{\#}\mu^{(u)}$ are (c_1, c_2)-regular, $\forall s = 1, 2, \ldots, S$, we can derive the following inequality from Lemma 3.1.4,

$$
\|f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}\|_1 \leq \sqrt{c_1\Big(\sqrt{\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\|f(\boldsymbol{x})\|_2^2\big]} + \sqrt{\mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\|f(\boldsymbol{x})\|_2^2\big]}\Big) + 2c_2} \times \big[\mathsf{W}_2(f_{\#}\mu^{(u)}, f_{\#}\mu^{(s)})\big]^{1/2}. \tag{3.23}
$$

Let:

$$
C := \max_s \sqrt{c_1\Big(\sqrt{\mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\big[\|f(\boldsymbol{x})\|_2^2\big]} + \sqrt{\mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\|f(\boldsymbol{x})\|_2^2\big]}\Big) + 2c_2}. \tag{3.24}
$$

Multiplying (3.23) by $\lambda^{(s)}$ and summing over all s, we get:

$$
\sum_{s=1}^{S} \lambda^{(s)} \|f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)}\|_1 \leq C \sum_{s=1}^{S} \lambda^{(s)} \big[\mathsf{W}_2(f_{\#}\mu^{(u)}, f_{\#}\mu^{(s)})\big]^{1/2}. \tag{3.25}
$$

By Jensen's inequality,

$$
\sum_{s=1}^{S} \lambda^{(s)} \big[\mathsf{W}_2(f_{\#}\mu^{(u)}, f_{\#}\mu^{(s)})\big]^{1/2} \leq \big[\sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu^{(u)}, f_{\#}\mu^{(s)})\big]^{1/4}. \tag{3.26}
$$

From (3.25) and (3.26),

$$\sum_{s=1}^{S} \lambda^{(s)} \| f_{\#}\mu^{(u)} - f_{\#}\mu^{(s)} \|_1 \leq C \big[\sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu^{(u)}, f_{\#}\mu^{(s)}) \big]^{1/4}. \tag{3.27}$$

Finally, combining the upper bound in Lemma 3.1.2 and (3.27), the proof follows. □

The upper bound in Theorem 3.1.5 consists of four terms: the first term is the sum of the risk on seen domains, the second term is the Wasserstein distance between the pushforward of seen and unseen domains in the representation space, the third term indicates how well the input can be reconstructed from its corresponding representation, and the fourth term is a combined risk that is independent of both the representation function and the labeling function and only intrinsic to the domain and loss function.

The form of the upper bound derived above shares some similarities with previous bounds in [22, 110, 118]. However, it differs from previous bounds in the following important aspects:

- Firstly, even though Lemma 1 in [110] and Theorem 1 in [118] employ Wasserstein distance to capture domain divergence, the corresponding term is constructed in the *data space*. By contrast, the corresponding term in our bound is constructed in the *representation space*, which not only provides a theoretical justification when decomposing the hypothesis into a representation mapping and a labeling function, but is also consistent with the algorithm implementation in practice. Moreover, the bounds in [110] and [118] are controlled by the Wasserstein-1 distance, while our upper bound is managed by the square root of the Wasserstein-2 distance. There are regimes where one bound is tighter than the other as discussed in Section 3.1.4.2.

- Secondly, our third term measures how well the input can be reconstructed from its representation. This motivates the use of an encoder-decoder structure in the

proposed algorithm in Section 3.1.7 to minimize the reconstruction loss. This is a novel component absent from [22, 110, 118].

- Finally, the last term in our upper bound is independent of both the representation function f and the labeling function g. This contrasts with the previous results in [22], where the last term in their upper bound (see Theorem 1 in [22]) depends on the representation function f. We make a detailed comparison in Section 3.1.4.2.

The bound proposed in Theorem 3.1.5 can also be used for the DA problem where one can access the unseen/target domain data and estimate its distribution. However, under the DG setting, the second and third term in (3.22) are uncontrollable, leading to an intractable upper bound due to the unavailability of the unseen data. This intractability, which cannot be overcome without making additional specific assumptions on the unseen domain, is widely accepted in the literature as a fundamental limitation for all DG methods and analyses.

As a step toward developing a practical algorithm based on our new bound, we decompose both the second term and the third term in (3.22) into two separate terms where one term completely depends on the unseen distribution and the other fully depends on the seen distributions.

Corollary 3.1.6. *Under the setting and notation of Theorem 3.1.5, for an arbitrary*

pushforward distribution $f_{\#}\mu$, we have:

$$
\begin{aligned}
R^{(u)}(h) \leq & \sum_{s=1}^{S} \lambda^{(s)} R^{(s)}(h) \\
& + LC\Big(\sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu, f_{\#}\mu^{(s)}) \Big)^{1/4} \\
& + LC\Big(\mathsf{W}_2^2(f_{\#}\mu^{(u)}, f_{\#}\mu) \Big)^{1/4} \\
& + QK\Big(\sum_{s=1}^{S} \lambda^{(s)} \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}} \big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2 \big] \Big) \\
& + QK\Big(\mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}} \big[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2 \big] \Big) \\
& + \sum_{s=1}^{S} \lambda^{(s)} \sigma^{(u,s)}. \qquad (3.28)
\end{aligned}
$$

Proof. We begin with the second term in the upper bound of Theorem 3.1.5. Indeed, for any arbitrary pushforward distribution $f_{\#}\mu$, we have:

$$
\begin{aligned}
& \Big[\sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu^{(u)}, f_{\#}\mu^{(s)}) \Big]^{1/4} && (3.29) \\
\leq & \Big[\sum_{s=1}^{S} \lambda^{(s)} \Big(\mathsf{W}_2^2(f_{\#}\mu^{(u)}, f_{\#}\mu) + \mathsf{W}_2^2(f_{\#}\mu, f_{\#}\mu^{(s)}) \Big) \Big]^{1/4} && (3.30) \\
= & \Big[\sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu^{(u)}, f_{\#}\mu) + \sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu, f_{\#}\mu^{(s)}) \Big]^{1/4} && (3.31) \\
= & \Big[\mathsf{W}_2^2(f_{\#}\mu^{(u)}, f_{\#}\mu) + \sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu, f_{\#}\mu^{(s)}) \Big]^{1/4} && (3.32) \\
\leq & \Big[\sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu, f_{\#}\mu^{(s)}) \Big]^{1/4} + \Big[\mathsf{W}_2^2(f_{\#}\mu^{(u)}, f_{\#}\mu) \Big]^{1/4} && (3.33)
\end{aligned}
$$

with (3.30) due to the triangle inequality, (3.32) due to $\sum_{s=1}^{S} \lambda^{(s)} = 1$, (3.33) due to the fact that for any $a, b \geq 0$ and $0 < p \leq 1$, $(a+b)^p \leq a^p + b^p$.

Combining (3.22) in Theorem 3.1.5 and (3.33), the proof of Corollary 3.1.6 follows.

□

3.1.4.2 Comparison between the Proposed Upper Bound and Previous Work

Although the bound in Theorem 1 of [22] was originally constructed for the domain adaptation problem, it has significantly influenced past and recent works in domain generalization as discussed earlier in Chapter 2. To highlight the differences between our work and the bound in Theorem 1 of [22] (Chapter 2, Theorem 2.2.1) and Theorem 4.1 of [146] (Chapter 2, Theorem 2.2.3), we provide a detailed comparison below:

- Firstly, [22] defines the risk induced by labeling function g from the representation space to the label space based on the disagreement between g and the optimal labeling function $g^{(u)}$:

$$R^{(u)}(g) = \mathbb{E}_{\boldsymbol{z}\sim f_{\#}\mu^{(u)}}\big[|g(\boldsymbol{z}) - g^{(u)}(\boldsymbol{z})|\big]. \tag{3.34}$$

  On the other hand, we define the risk induced by using a hypothesis h from the input space to the label space by the disagreement between h and the optimal hypothesis $h^{(u)}$ via a general loss function $\ell(\cdot,\cdot)$:

$$R^{(u)}(h) = \mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\big[\ell(h(\boldsymbol{x}), h^{(u)}(\boldsymbol{x}))\big]. \tag{3.35}$$

  Since the empirical risk measures the probability of misclassification of a hypothesis that maps from the input space to the label space, minimizing $R^{(u)}(g)$ does not guarantee to minimize the empirical risk. Though there are some cases for the causality to hold, for example, if the representation function f is invertible *i.e.*, there is a one-to-one mapping between $\boldsymbol{x}$ and $\boldsymbol{z}$, and the loss function has the form of $\ell(a,b) = |a-b|$, it is possible to verify that $R^{(u)}(g) = R^{(u)}(h)$. In general, the representation mapping might not be invertible. For example, let us consider a representation function f that maps $f(\boldsymbol{x}_1) = f(\boldsymbol{x}_2) = \boldsymbol{z}$, $\boldsymbol{x_1} \neq \boldsymbol{x_2}$,

with corresponding labels as $y_1 = 0$ and $y_2 = 1$. In this case, the risk defined in (3.34) will introduce a larger error than the risk introduced in (3.35) since $g(\boldsymbol{z})$ cannot be mapped to both "0" and "1". That said, the risk defined in (3.35) is more precise to describe the empirical risk. In addition, the risk defined in (3.34) is only a special case of (3.35) when the representation mapping f is invertible and the loss function satisfies $\ell(a, b) = |a - b|$.

- Secondly, using the setting in [22], for a given hypothesis space, the ideal joint hypothesis g^* is defined as the hypothesis which globally minimizes the combined error from seen and unseen domains [21, 22]:

$$g^* = \operatorname*{argmin}_{g \in \mathcal{G}} \big(R^{(s)}(g) + R^{(u)}(g) \big).$$

  In other words, this hypothesis should work well in both domains. The error induced by using this ideal joint hypothesis is called *combined risk*:

$$\lambda = \inf_{g \in \mathcal{G}} \big(R^{(s)}(g) + R^{(u)}(g) \big) = \big(R^{(s)}(g^*) + R^{(u)}(g^*) \big).$$

  Note that the labeling function g is a mapping from the representation space to the label space, therefore, the ideal labeling function g^* depends implicitly on the representation function f, hence, λ depends on f. Simply ignoring this fact and treating λ as a constant may loosen the upper bound. By contrast, our goal is to construct an upper bound with the *combined risk* term $\sigma^{(u,s)}$ independent of both the representation function and the labeling function, which can be seen from Lemma 3.1.1 and Theorem 3.1.5.

- Finally, it is worth comparing our upper bound with the bound in Theorem 4.1 of [146] which also has the *combined risk* term free of the choice of the hypothesis class. However, note that the result in Theorem 4.1 of [146] does

not consider any representation function f, *i.e.*, their labeling function directly maps from the input space to the label space, while our hypothesis is composed of a representation function from the input space to the representation space followed by a labeling function from the representation space to the label space. Since it is possible to pick a representation function f that maps any input to itself, *i.e.*, $f(\boldsymbol{x}) = \boldsymbol{x}$ which leads to $h = g \circ f = g$, the bound in [146] can be viewed as a special case of our proposed upper bound in Lemma 3.1.1.

The form of the proposed upper bound derived in Theorem 3.1.5 shares some similarities with Lemma 1 in [110] and Theorem 1 in [118] (Chapter 2, Theorem 2.2.4), for example, all of them introduce Wasserstein distance between domain distributions. However, they differ in the following key aspects.

1. The term containing Wasserstein distance in our upper bound is constructed in the *representation* space, not in the data (ambient) space, which provides a theoretical justification when decomposing the hypothesis into a representation mapping and a labeling function. This is also consistent with the algorithmic implementation in practice.

2. The bounds in Lemma 1 of [110] and Theorem 1 of [118] are controlled by the Wasserstein-1 distance while our upper bound is managed by the square-root of the Wasserstein-2 distance. There are regimes where one bound is tighter than the other. It is well-known that $\mathsf{W}_1(\mu, \nu) \leq \mathsf{W}_2(\mu, \nu)$, if $\mathsf{W}_2(\mu, \nu) \leq 1$, then $\mathsf{W}_1(\mu, \nu) \leq \sqrt{\mathsf{W}_2(\mu, \nu)}$. However, based on Jensen's inequality, it is possible to show that $\sqrt{\mathsf{W}_2(\mu, \nu)} \leq [Diam(f(\boldsymbol{X}))\mathsf{W}_1(\mu, \nu)]^{1/4}$ where $Diam(f(\boldsymbol{X}))$ denotes the largest distance between two points in the representation space $\mathbb{R}^{d'}$ generated by input $\boldsymbol{X}$ via mapping f. To guarantee $\sqrt{\mathsf{W}_2(\mu, \nu)} \leq \mathsf{W}_1(\mu, \nu)$, a sufficient condition is $[Diam(f(\boldsymbol{X}))\mathsf{W}_1(\mu, \nu)]^{1/4} \leq \mathsf{W}_1(\mu, \nu)$ which is equivalent to $Diam(f(\boldsymbol{X})) \leq \mathsf{W}_1(\mu, \nu)^3$. In fact, for a given $Diam(f(\boldsymbol{X}))$, the larger the value of $\mathsf{W}_1(\mu, \nu)$, the higher the chance that this sufficient condition will hold.

3.1.5 Proposed Method

Motivated by the bound in Corollary 3.1.6, we want to find a suitable representation function f together with a reconstruction function ψ to minimize the second term $\sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu, f_{\#}\mu^{(s)})$ and the fourth term $\sum_{s=1}^{S} \lambda^{(s)} \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\left[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\right]$ in (3.28), while ignoring the third term $\mathsf{W}_2^2(f_{\#}\mu^{(u)}, f_{\#}\mu)$ and the fifth term $\mathbb{E}_{\boldsymbol{x}\sim\mu^{(u)}}\left[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\right]$, as both of them are intractable.

Minimizing the second term $\sum_{s=1}^{S} \lambda^{(s)} \mathsf{W}_2^2(f_{\#}\mu, f_{\#}\mu^{(s)})$ in (3.28) leads to finding the Wasserstein-2 barycenter of the distributions of seen domains in the representation space. Here, we assume a uniform weight of $\lambda^{(s)} = \frac{1}{S}$ for all s, since there is no additional information for selecting these weights. For this choice, the Wasserstein-2 barycenter of the pushforward distributions of seen domains is defined by:

$$f_{\#}\mu_{barycenter} := \underset{f_{\#}\mu}{\operatorname{argmin}} \sum_{s=1}^{S} \frac{1}{S} \mathsf{W}_2^2(f_{\#}\mu^{(s)}, f_{\#}\mu). \tag{3.36}$$

We refer the reader to [2, 38] for the definition and properties (existence, uniqueness) of the Wasserstein barycenter.

On the other hand, minimizing the fourth term $\sum_{s=1}^{S} \lambda^{(s)} \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}\left[\|\psi(f(\boldsymbol{x})) - \boldsymbol{x}\|_2\right]$ in (3.28) naturally leads to an auto-encoder mechanism. With a little abuse of notation, we denote the encoder, namely the representation function as f and the decoder, namely the reconstruction function as ψ. The L^2 reconstruction loss should be optimized over all seen domains.

3.1.6 Objective Functions

As the last term in (3.28) of Corollary 3.1.6 is independent of both the representation function f and the labeling function g, and the third and fifth terms are intractable due to their dependence on unseen domain, we focus on designing f, ψ and g to minimize the first, second, and fourth terms in (3.28) of Corollary 3.1.6.

Following previous works [9,21,22], we optimize the first term by training f together with g using a standard cross-entropy (CE) loss, such that the empirical classification risk on seen domains is minimized. The classification loss function can be written as:

$$\mathsf{L}_c(f, g) = \frac{1}{S}\sum_{s=1}^{S} \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}[\mathsf{CE}(h^{(s)}(\boldsymbol{x}), g(f(\boldsymbol{x})))] \tag{3.37}$$

where $\mathsf{CE}(h^{(s)}(\boldsymbol{x}), g(f(\boldsymbol{x})))$ denotes the cross-entropy (CE) loss between the output of classifier and the ground-truth label of seen domain s.

As discussed in Corollary 3.1.6, we propose to use the Wasserstein-2 barycenter of representation distributions of seen domains to optimize the second term in (3.28). Specifically, the barycenter loss is defined by:

$$\mathsf{L}_{bary}(f) := \sum_{s=1}^{S} \frac{1}{S}\mathsf{W}_2^2(f_{\#}\mu^{(s)}, f_{\#}\mu_{barycenter}) \tag{3.38}$$

where $f_{\#}\mu_{barycenter}$, as defined in (3.36), denotes the Wasserstein barycenter of push-forward distributions of seen domains.

In contrast to the previous Wasserstein distance-based method [150] where pairwise Wasserstein distance loss is employed, we motivate the use of Wasserstein barycenter loss based on our Corollary 3.1.6 and demonstrate its ability in enforcing domain-invariance in the ablation study of Section 3.1.8.4. Notably, the barycenter loss (3.38) only requires computing S Wasserstein distances, whereas using pairwise Wasserstein distance would require $S(S-1)/2$ computations.

Furthermore, to handle the fourth term in (3.28), we utilize the auto-encoder structure. Specifically, a decoder $\psi : \mathbb{R}^{d'} \to \mathbb{R}^d$ is adopted, leading to the following reconstruction loss term:

$$\mathsf{L}_r(f, \psi) := \frac{1}{S}\sum_{s=1}^{S} \mathbb{E}_{\boldsymbol{x}\sim\mu^{(s)}}[\|\boldsymbol{x} - \psi(f(\boldsymbol{x}))\|^2]. \tag{3.39}$$

From the analysis above, we aim to find a representation function f, a classifier g, and a decoder function ψ to optimize the following objective function:

$$\arg\min_{f,g,\psi} \mathsf{L}_c(f,g) + \alpha \mathsf{L}_{bary}(f) + \beta \mathsf{L}_r(f,\psi) \tag{3.40}$$

where weights $\alpha, \beta > 0$ are hyper-parameters. One can observe that the terms in our proposed upper bound are incorporated into our objective function in (3.40). Specifically, the first term in our objective function aims to determine a good classifier g together with a representation mapping f by minimizing the risk of seen domains, which corresponds to the first term of the upper bound in (3.28). The second term in (3.40) acts as a domain alignment tool to minimize the discrepancy between seen domains, aligning with the second term in the proposed bound in (3.28). Note that although L_{bary} itself requires solving an optimization problem, we leverage fast computation methods, which are also discussed in Section 3.1.7, to directly estimate this loss without invoking the Kantorovich-Rubenstein dual characterization of Wasserstein distance [115]. This avoids solving a min-max type problem that is often plagued by unstable numerical dynamics. Finally, the third term in the objective function minimizes the mean squared error between the input and its reconstruction over all seen domains, which directly minimizes the fourth term in (3.28).

3.1.7 Algorithm

Based on the loss function designed above, we propose an algorithm named Wasserstein Barycenter Auto-Encoder (WBAE). The pseudo code of the WBAE algorithm can be found in Algorithm 1 while its block diagram is shown in Fig. 3-1.

As shown in the pseudo code, we use an encoder f and a decoder ψ, which are parameterized by θ_e and θ_d for feature extraction and reconstruction, respectively. Here we denote $\mathcal{X}^{(s)}$ as a set of samples from domain s with empirical distribution $\hat{\mu}^{(s)}$

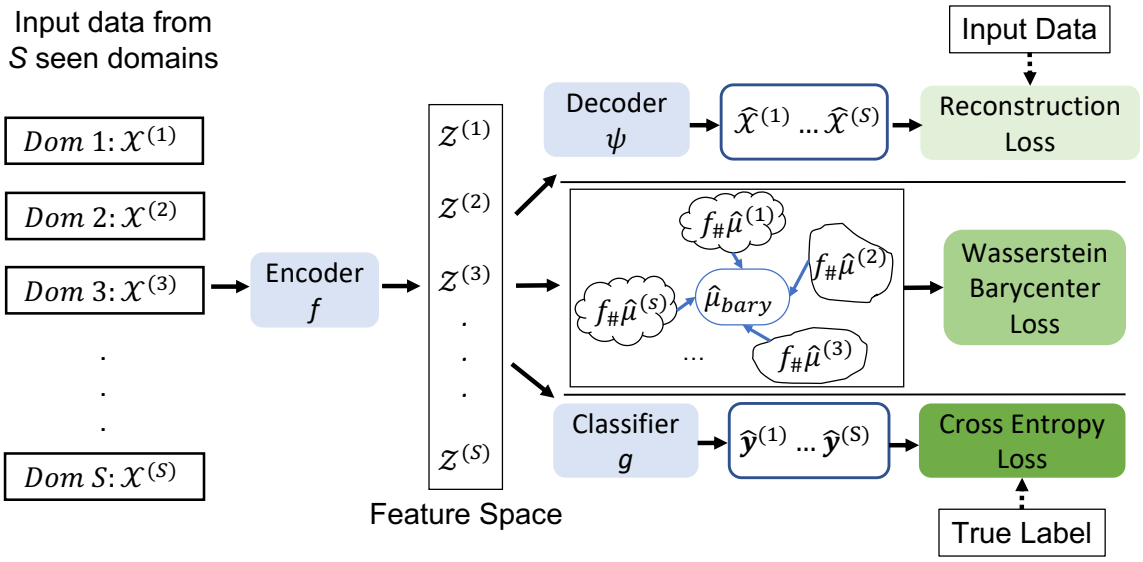


Figure 3-1: An overview of the proposed algorithm. The top, middle, and bottom branches refer to the reconstruction loss term, the Wasserstein barycenter loss term, and the classification risk (from seen domains), respectively.

and with $\boldsymbol{x}_i^{(s)}$ as one of its element. The corresponding label set of $\mathcal{X}^{(s)}$ is denoted as $\boldsymbol{y}^{(s)}$, where $\boldsymbol{y}^{(s)} := \{y_i^{(s)}\}$ with $y_i^{(s)}$ as the label for sample $\boldsymbol{x}_i^{(s)}$. The extracted feature $\boldsymbol{z}_i^{(s)} = f_{\theta_e}(\boldsymbol{x}_i^{(s)})$ in set $\mathcal{Z}^{(s)}$ follows the empirical distribution of $f_{\#}\hat{\mu}^{(s)}$. The decoder takes the extracted features as input and outputs the reconstructions as $\psi_{\theta_d}(\boldsymbol{z}_i^{(s)})$ for domain s. The classifier g, parameterized by θ_c is then applied to the extracted features for label prediction.

The proposed algorithm requires calculating Wasserstein-2 barycenter and its supporting points. Here we use an off-the-shelf python package [50] that implements a free-support Wasserstein barycenter algorithm described in [38]. This algorithm is executed in the primal domain and avoids the use of the dual form of Wasserstein distances, which otherwise would turn the problem into an adversarial (min-max) type setting that we want to avoid due to its instability. The barycenter loss is approximated via an average Sinkhorn divergence [49] between the seen domains and the estimated barycenter. Sinkhorn divergence is an unbiased proxy for the Wasserstein

Algorithm 1 Wasserstein Barycenter Auto-Encoder (WBAE)

Input: Data from S seen domains, m samples from each domain, learning rate η, parameters α, β, ϵ. **Output**: Encoder f_{θ_e}, decoder ψ_{θ_d}, classifier g_{θ_c}

while training is not end **do**
  Randomly choose m samples from each domain, denoted as $\mathcal{X}^{(s)} := \{\boldsymbol{x}_i^{(s)}\}_{i=1}^m \sim \hat{\mu}^{(s)}$ and $\boldsymbol{y}^{(s)} := \{y_i^{(s)}\}_{i=1}^m$
  for $s = 1 : S$ and $i = 1 : m$ **do**
    $\boldsymbol{z}_{\boldsymbol{i}}^{(s)} \leftarrow f_{\theta_e}(\boldsymbol{x}_i^{(s)})$ with set $\mathcal{Z}^{(s)} \sim f_{\#}\hat{\mu}^{(s)}$
  end for
  Calculate the Wasserstein barycenter $\hat{\mu}_{bary}$ of $\{f_{\#}\hat{\mu}^{(s)}\}_{s=1}^S$ and its supporting points with f_{θ_e} detached from automatic backpropagation
  $\mathsf{L}_{bary} \leftarrow \frac{1}{S}\sum_{s=1}^S Sinkhorn_\epsilon(\hat{\mu}_{bary}, f_{\#}\hat{\mu}^{(s)})$
  $\mathsf{L}_c \leftarrow -\frac{1}{mS}\sum_{s=1}^S\sum_{i=1}^m y_i^s \log p(g_{\theta_c}(f_{\theta_e}(\boldsymbol{x}_i^{(s)})))$
  $\mathsf{L}_r \leftarrow \frac{1}{mS}\sum_{s=1}^S\sum_{i=1}^m \|\boldsymbol{x}_i^{(s)} - \psi_{\theta_d}(\boldsymbol{z}_i^{(s)})\|_2^2$
  $\mathsf{L} \leftarrow \mathsf{L}_c + \alpha\mathsf{L}_{bary} + \beta\mathsf{L}_r$
  $\theta_c \leftarrow \theta_c - \eta\nabla_{\theta_c}\mathsf{L}_c$
  $\theta_d \leftarrow \theta_d - \eta\nabla_{\theta_d}\mathsf{L}_r$
  $\theta_e \leftarrow \theta_e - \eta\nabla_{\theta_e}\mathsf{L}$
end while

distance, which leverages entropic regularization [37] for computational efficiency, thereby allowing for integrating automatic differentiation with GPU computation. We incorporate the implementation from [49] into our algorithm for a fast gradient computation and denote it as $Sinkhorn_\epsilon$ in Algorithm 1, where ϵ is the entropic regularization term.

3.1.8 Experiments and Results

The proposed method was evaluated on four benchmark datasets for DG: PACS [82], VLCS [48], Office-Home [131], and TerraIncognita [20] under two different settings: DomainBed setting [57] and Stochastic Weight Averaging Densely (SWAD) setting [28]. In the DomainBed setting, we implemented our method with the widely used DomainBed package and compared it with various *theory-guided* DG algorithms. Additionally, incorporating the recent advancement in DG-specific optimization, we

Table 3.1: Performance of tested methods on PACS dataset in the DomainBed setting, measured by accuracy (%). A, C, P, S are left-out unseen domains.

Algorithm	**A**	**C**	**P**	**S**	**Avg**
theory-guided algorithms					
ERM	84.7 ± 0.4	80.8 ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5
IRM	84.8 ± 1.3	76.4 ± 1.1	96.7 ± 0.6	76.1 ± 1.0	83.5
DANN	86.4 ± 0.8	77.4 ± 0.8	**97.3** ± 0.4	73.5 ± 2.3	83.6
CDANN	84.6 ± 1.8	75.5 ± 0.9	96.8 ± 0.3	73.5 ± 0.6	82.6
MTL	**87.5** ± 0.8	77.1 ± 0.5	96.4 ± 0.8	77.3 ± 1.8	84.6
VREx	86.0 ± 1.6	79.1 ± 0.6	96.9 ± 0.5	77.7 ± 1.7	84.9
WBAE (Ours)	86.9 ± 0.3	**81.3** ± 0.4	97.2 ± 0.2	**80.5** ± 0.4	**86.5**
best-performing heuristic algorithm					
SagNet	87.4 ± 1.0	80.7 ± 0.6	97.1 ± 0.1	80.0 ± 0.4	86.3

conducted separate experiments using the SWAD [28] weight sampling strategy with the same experimental setting described in [28]. Furthermore, to investigate the impact of different components of the proposed loss function, we conducted an ablation analysis on the PACS, VLCS, and Office-Home datasets and reported the results in Section 3.1.8.4.

3.1.8.1 Datasets

The details for the four datasets are described below:

- **PACS dataset [82]:** PACS contains 9,991 images with 7 classes from 4 domains: Art (A), Cartoons (C), Photos (P) and Sketches (S), where each domain represents one type of images.
- **VLCS dataset [48]:** VLCS consists of 10,729 images from 4 different domains: VOC2007 (V), LabelMe (L), Caltech (C), PASCAL (S). A total of 5 classes are shared by all domains.
- **Office-Home dataset [131]:** Office-Home contains 15,500 images from 4

Table 3.2: Performance of tested methods on VLCS dataset in the DomainBed setting, measured by accuracy (%). C, L, S, V are left-out unseen domains.

Algorithm	**C**	**L**	**S**	**V**	**Avg**
theory-guided algorithms					
ERM	97.7 ± 0.4	64.3 ± 0.9	73.4 ± 0.5	74.6 ± 1.3	77.5
IRM	98.6 ± 0.1	64.9 ± 0.9	73.4 ± 0.6	77.3 ± 0.9	78.5
DANN	**99.0** ± 0.3	65.1 ± 1.4	73.1 ± 0.3	77.2 ± 0.6	78.6
CDANN	97.1 ± 0.3	65.1 ± 1.2	70.7 ± 0.8	77.1 ± 1.5	77.5
MTL	97.8 ± 0.4	64.3 ± 0.3	71.5 ± 0.7	75.3 ± 1.7	77.2
VREx	98.4 ± 0.3	64.4 ± 1.4	**74.1** ± 0.4	76.2 ± 1.3	78.3
WBAE (Ours)	98.3 ± 0.2	65.5 ± 1.0	72.8 ± 0.3	**78.6** ± 0.4	**78.8**
best-performing heuristic algorithm					
CORAL	98.3 ± 0.1	**66.1** ± 1.2	73.4 ± 0.3	77.5 ± 1.2	**78.8**

different domains: Artistic (A), Clipart (C), Product (P), and Real-World (R). Each domain has 65 object categories.

- **TerraIncognita dataset [20]:** TerraIncognita contains four domains {L100, L38, L43, L46} with a total of 24,788 pictures of wild animals belonging to 10 classes.

Example images of the above datasets are shown in Fig. 3-2 and Fig. 3-3.

3.1.8.2 Methods for Comparison

In this work, we compare the empirical performance of our method against the state-of-the-art DG methods reported in [57] under the DomainBed setting. Specifically, the competing methods include:

- Empirical Risk Minimization (**ERM**) [130] which aims to minimize the cumulative training error across all seen domains.

- Domain-Adversarial Neural Networks (**DANN**) [51] which is motivated by the theoretical results from [22]. In particular, to minimize the upper bound of the

Table 3.3: Performance of tested methods on Office-Home dataset in the DomainBed setting, measured by accuracy (%). A, C, P, R are left-out unseen domains.

Algorithm	**A**	**C**	**P**	**R**	**Avg**
theory-guided algorithms					
ERM	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5
IRM	58.9 ± 2.3	52.2 ± 1.6	72.1 ± 2.9	74.0 ± 2.5	64.3
DANN	59.9 ± 1.3	53.0 ± 0.3	73.6 ± 0.7	76.9 ± 0.5	65.9
CDANN	61.5 ± 1.4	50.4 ± 2.4	74.4 ± 0.9	76.6 ± 0.8	65.8
MTL	61.5 ± 0.7	52.4 ± 0.6	74.9 ± 0.4	76.8 ± 0.4	66.4
VREx	60.7 ± 0.9	53.0 ± 0.9	75.3 ± 0.1	76.6 ± 0.5	66.4
WBAE (Ours)	63.7 ± 0.5	**56.4** ± 0.8	76.1 ± 0.3	**78.8** ± 0.4	**68.8**
best-performing heuristic algorithm					
CORAL	**65.3** ± 0.4	54.4 ± 0.5	**76.5** ± 0.1	78.4 ± 0.5	68.7

risk in the unseen domain, **DANN** adopts an adversarial network to enforce that features from different domains are indistinguishable.

- Class-conditional DANN (**C-DANN**) [86] is a variant of **DANN** that aims to match the conditional distributions of feature given the label across domains.

- Invariant Risk Minimization (**IRM**) [12] aims to learn features such that the optimal classifiers applied to these features are matched across domains.

- Risk Extrapolation (**VREx**) [78] is constructed on the assumption from [12] which assumes the existence of an optimal linear classifier across all domains. While **IRM** specifically seeks the invariant classifier, **VREx** aims to identify the form of the distribution shift and propose a variance penalty, leading to the robustness for a wider variety of distributional shifts.

- Marginal Transfer Learning (**MTL**) [23,24] is proposed based on an upper bound for the generalization error under the setting of an Agnostic Generative Model. Specifically, **MTL** estimates the mean embedding per domain and uses it as a second argument for optimizing the classifier.

Table 3.4: Performance of tested methods on TerraIncognita dataset in the DomainBed setting, measured by accuracy (%). L100, L38, L43, L46 are left-out unseen domains.

Algorithm	**L100**	**L38**	**L43**	**L46**	**Avg**
theory-guided algorithms					
ERM	49.8 ± 4.4	42.1 ± 1.4	56.9 ± 1.8	35.7 ± 3.9	46.1
IRM	54.6 ± 1.3	39.8 ± 1.9	56.2 ± 1.8	39.6 ± 0.8	47.6
DANN	51.1 ± 3.5	40.6 ± 0.6	57.4 ± 0.5	37.7 ± 1.8	46.7
CDANN	47.0 ± 1.9	41.3 ± 4.8	54.9 ± 1.7	39.8 ± 2.3	45.8
MTL	49.3 ± 1.2	39.6 ± 6.3	55.6 ± 1.1	37.8 ± 0.8	45.6
VREx	48.2 ± 4.3	41.7 ± 1.3	56.8 ± 0.8	38.7 ± 3.1	46.4
WBAE (Ours)	**55.3** ± 0.4	**44.3** ± 0.7	56.4 ± 0.5	39.1 ± 0.6	**48.8**
best-performing heuristic algorithm					
SagNet	53.0 ± 2.9	43.0 ± 2.5	**57.9** ± 0.6	**40.4** ± 1.3	48.6

Table 3.5: Performance of theory-guided methods on four datasets in the DomainBed setting, measured by accuracy (%). The average accuracy is reported over different tasks per dataset.

Algorithm	**PACS**	**VLCS**	**Office-Home**	**TerraIncognita**	**Avg**
ERM	85.5	77.5	66.5	46.1	68.9
IRM	83.5	78.5	64.3	47.6	68.5
DANN	83.6	78.6	65.9	46.7	68.7
CDANN	82.6	77.5	65.8	45.8	67.9
MTL	84.6	77.2	66.4	45.6	68.5
VREx	84.9	78.3	66.4	46.4	69.0
WBAE (Ours)	**86.5**	**78.8**	**68.8**	**48.8**	**70.7**

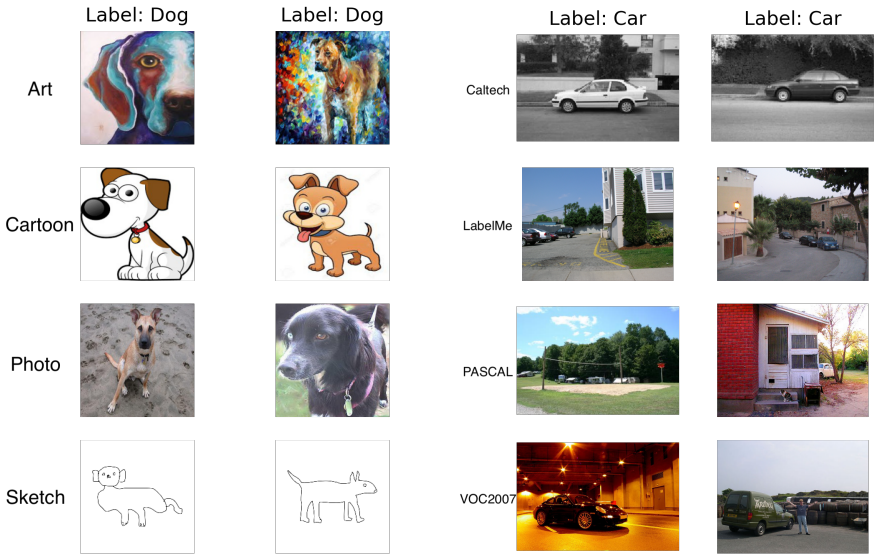
- CORrelation ALignment (**CORAL**) [125] is based on the idea of matching the mean and covariance of feature distributions from different domains.

- Style-Agnostic Networks (**SagNet**) [99] minimizes the style induced domain gap by randomizing the style feature for different domains and train the model mainly on the disentangled content feature.

We can categorize the algorithms provided in [57] into two groups: (1) heuristic algorithms, which lack theoretical analysis, and (2) theory-guided algorithms. As the

Table 3.6: Performance of tested methods on four datasets in the SWAD setting, measured by accuracy (%).

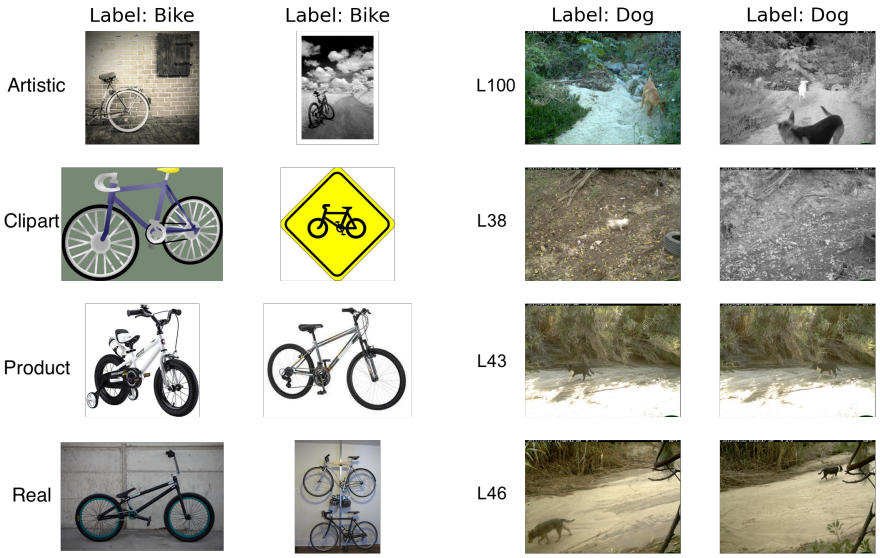
Algorithm	PACS	VLCS	Office-Home	TerraIncognita	Avg
ERM + SWAD	88.1 ± 0.1	79.1 ± 0.1	70.6 ± 0.2	50.0 ± 0.3	72.0
CORAL + SWAD	88.3 ± 0.1	78.9 ± 0.1	71.3 ± 0.1	51.0 ± 0.1	72.4
WBAE + SWAD	**88.4** ± 0.1	**79.5** ± 0.1	**71.4** ± 0.2	**51.8** ± 0.3	**72.8**



(a) PACS (b) VLCS

Figure 3-2: Example images of PACS and VLCS.

proposed method in this work falls into the second category, we primarily compare it with the theory-guided methods. Here, **ERM** acts as the baseline theory-guided model and **DANN**, **C-DANN**, **IRM**, **VREx**, **MTL** are five state-of-the-art theory-guided algorithms. Besides these six methods, for a complete comparison, we also include three heuristic algorithms that achieve the best performances on four evaluated datasets [57]. More specific, **SagNet** [99] is the best-performing algorithm for the PACS and TerraIncognita datasets, and **CORAL** [125] is the best-performing algorithm for both the VLCS and Office-Home datasets. In the SWAD setting, following [28] where



(a) Office-Home (b) TerraIncognita

Figure 3-3: Example images of Office-Home and TerraIncognita.

CORAL was considered as the representative of the previous state-of-the-art methods, we compare our method with both **ERM** and **CORAL**, all of which employed the SWAD strategy. The results for the competing methods above are sourced from [57] and [28].

3.1.8.3 Experiment Settings

Model Structure: We used the same feature extractor and classifier as used in [57] for all four datasets. Specifically, an ImageNet pre-trained ResNet-50 model with the final (softmax) layer removed is used as the feature extractor. The decoder is a stack of 6 ConvTranspose2d layers for all datasets. The detailed structure of the decoder is described in Table 3.7. The classifier is a one-linear-layer model with the output dimension the same as the number of classes.

Hyper-parameters: In the DomainBed setting, we performed a random search of 20

Table 3.7: Model structure of the decoder.

Layer
ConvTranspose2d (in=2048, out=512, kernel_size=4, stride= 1, padding=0)
BatchNorm2d + ReLU
ConvTranspose2d (in=512, out=256, kernel_size=4, stride=2, padding=1)
BatchNorm2d + ReLU
ConvTranspose2d (in=256, out=128, kernel_size=4, stride=2, padding=1)
BatchNorm2d + ReLU
ConvTranspose2d (in=128, out=64, kernel_size=4, stride=2, padding=1)
BatchNorm2d + ReLU
ConvTranspose2d (in=64, out=32, kernel_size=4, stride=2, padding=1)
BatchNorm2d + ReLU
ConvTranspose2d (in=32, out=3, kernel_size=4, stride=2, padding=1)
Tanh + Interpolate (size=(224, 224))

Table 3.8: Hyper-parameters of the proposed method.

Parameters	DomainBed Setting	SWAD Setting
Optimizer	Adam [74]	Adam [74]
Learning rate	5×10^{-5}	$\{10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$
Batch size	32	32
ResNet dropout	0	{0.0, 0.1, 0.5}
Weight decay	0	$\{10^{-4}, 10^{-6}\}$
Training steps	2000	5000
ϵ	20	20
α	$10^{\text{Uniform}(-3.5,-2)}$	$\{10^{-3.5}, 10^{-3}, 10^{-2.5}, 10^{-2}\}$
β	$10^{\text{Uniform}(-3.5,-1.5)}$	$\{10^{-3.5}, 10^{-3}, 10^{-2}, 10^{-1.5}\}$

trials within the joint distribution of $10^{\text{Uniform}[-3.5,-2]}$ for α and $10^{\text{Uniform}[-3.5,-1.5]}$ for β (see (3.40)) with other hyper-parameters (*e.g.*, learning rate, batch size, dropout rate, *etc.*) set as the default values recommended in [57]. In the SWAD setting, following [28], we performed a grid search for α in $\{10^{-3.5}, 10^{-3}, 10^{-2.5}, 10^{-2}\}$ and β in $\{10^{-3.5}, 10^{-3}, 10^{-2}, 10^{-1.5}\}$. We chose the value of ϵ for the Sinkhorn loss (line 7, Algorithm 1) as 20, which is the smallest value that can produce stable training processes.

In the SWAD setting, following [28], we first fixed all algorithm-agnostic hyper-

parameters (HPs) and only tuned the algorithm-specific HPs. Specifically, we first fixed the learning rate as 5×10^{-5}, Resnet dropout rate and weight decay both as 0, and grid searched α, β in $\{10^{-3.5}, 10^{-3}, 10^{-2.5}, 10^{-2}\}$ and $\{10^{-3.5}, 10^{-3}, 10^{-2}, 10^{-1.5}\}$ with the batch size as 32. Then we searched learning rate, Resnet dropout rate, and weight decay in $\{10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$, $\{0.0, 0.1, 0.5\}$ and $\{10^{-4}, 10^{-6}\}$, with the selected α, β, as performed in [28]. We used the same values for SWAD-specific hyper-parameters as those used in [28], without any further tuning. A full list of hyper-parameters can be found in Table 3.8.

Model Selection: We adopted the commonly used training-domain validation strategy in [57, 78] for hyper-parameter tuning and model selection. Specifically, we split the data from each domain into training and validation sets in the proportion 80% and 20%, respectively. During training, we aggregated together the training/validation samples from each seen domain to form the overall training/validation set and selected the model with the highest validation accuracy for testing.

All models were trained on a single NVIDIA Tesla V100 16GB GPU. Experiments on each dataset are repeated three times with different random seeds and the average accuracy together with its standard error are reported.

3.1.8.4 Results and Ablation Study

As shown in Table 3.1, 3.2, 3.3, and 3.4, the proposed method (WBAE) performs comparably or better than the state-of-the-art methods. In particular, WBAE achieves the highest accuracy in three out of the four datasets compared to all methods, with a moderate improvement over all theory-guided methods on all four datasets. Additionally, the proposed method performs equally well as, or slightly better than, the best-performing heuristic DG methods.

In Table 3.1, it is demonstrated that WBAE outperforms other theory-guided methods by 0.5% and 1.2% points in both Cartoons (C) and Sketches (S) domains,

respectively, and by at least 1% point on average on the PACS dataset. Similarly, Table 3.2 shows that WBAE achieves a performance gain of at least 0.2% points over all theory-guided comparison methods on the VLCS dataset. The effectiveness of the proposed method is further highlighted in Tables 3.3 and 3.4, which present results on the larger and more challenging Office-Home and TerraIncognita datasets. Specifically, compared to all theory-guided methods on Office-Home, WBAE boosts the average accuracy by at least 2.3% points on average and at least 2.2%, 3.4%, 0.3%, and 1.9% points on each task. Regarding the TerraIncognita dataset, the proposed algorithm still exhibits superiority by outperforming all theory-guided methods by at least 1.2% points, as shown in Table 3.4. A summary of evaluation results in the DomainBed setting is reported in Table 3.5. The proposed method outperforms all theory-guided methods with a noticeable improvement of 1.7-2.8 percentage points on average across all tested datasets.

Table 3.6 presents the results obtained by applying SWAD, a DG-specific optimizer and weight-averaging technique, in combination with our proposed algorithm WBAE. It can be observed that this combination outperforms all comparison methods on all four evaluated datasets, with an average improvement of 0.4% point over the previous best-performing method **CORAL** as reported in [28].

Based on the results above, it is evident that the proposed algorithm has a more significant impact on the PACS, Office-Home, and TerraIncognita datasets compared to the VLCS dataset. One possible explanation for this, as also suggested in [149], is that three out of four domains in the VLCS dataset contain a greater proportion of scenery contents rather than object information. Unlike the scenery background in TerraIncognita dataset, the scenery contents in the VLCS dataset are usually more intricate and sometimes include multiple objects, making it challenging for the feature extractor to obtain useful object information for the downstream classification.

To study the impact of different components of the loss function in (3.40), we

Table 3.9: Ablation study for the proposed algorithm (WBAE) on PACS, VLCS, and Office-Home datasets.

Dataset	no L_{bary}	no L_r	WBAE
PACS	85.3 ± 0.3	86.0 ± 0.1	86.5 ± 0.2
VLCS	77.9 ± 0.1	78.4 ± 0.2	78.8 ± 0.2
Office-Home	65.7 ± 0.2	67.7 ± 0.1	68.8 ± 0.1

conducted an ablation study for WBAE on all datasets except TerraIncognita due to our limited computational resources. In particular, we consider the following variants of our method: (1) no L_{bary}: using the WBAE loss function without the Wasserstein barycenter term L_{bary}; (2) no L_r: using the WBAE loss function without the reconstruction term L_r. We re-ran all the experiments three times using the same model architectures, hyper-parameter tuning, and validation method.

Table 3.9 demonstrates the performance of the model with different loss terms removed from the original WBAE loss function. It can be observed that removing L_r from the WBAE loss function leads to a decrease in the accuracy of 0.5%, 0.4%, and 1.1% points for PACS, VLCS, and Office-Home datasets, respectively. The performance deterioration is more significant when removing L_{bary} from the WBAE loss function, leading to a drop of 1.2%, 0.9%, and 3.1% points for PACS, VLCS, and Office-Home datasets, respectively. Our ablation study demonstrates the importance of the Wasserstein barycenter loss and also highlights the auxiliary role of the reconstruction loss. Specifically, removing the Wasserstein barycenter loss (L_{bary}) will result in diminished performance, and a similar, though less significant, decrease will occur if the reconstruction loss (L_r) is removed.

3.1.9 Limitations

In terms of algorithm and numerical implementation, it should be noted that while our theory-guided method is effective in addressing the DG problem, it may become

computationally expensive if one wants to use a larger batch size for a more accurate estimation of the Wasserstein-2 barycenter. To alleviate this constraint, the future work should focus on leveraging the recently proposed large-scale-barycenter and mapping estimators [47, 76] to enable the calculation of barycenters with a larger number of samples.

3.1.10 Conclusion

In this section, we revisited the theory and methods for DG and provided a new upper bound for the risk in the unseen domain. The proposed upper bound contains four terms: (1) the empirical risk of the seen domains in the input space; (2) the discrepancy between the induced representation distribution of seen and unseen domains, which can be further represented by the Wasserstein-2 barycenter of representation in the seen domains; (3) the reconstruction loss term that measures how well the data can be reconstructed from its representation; and (4) a combined risk term. Our upper bound provides valuable insights in three aspects. Firstly, we observed that the combined risk term in previous bounds relied on the representation function, which made optimization challenging. By contrast, our combined risk term in the proposed upper bound is a constant with respect to both the representation and the labeling function, making optimization straightforward, thus bridging the previous gap between theory and practice. Secondly, compared with other upper bounds using Wasserstein distance to measure the domain discrepancy, the proposed bound constructs the discrepancy term in the representation space rather than in the data space. This approach offers a theoretical justification for the decomposition of the hypothesis when bounding the risk and for practical implementation when designing the algorithm. Lastly, motivated by the proposed upper bound, our practical algorithm WBAE demonstrates competitive performance over state-of-the-art DG algorithms, validating the usefulness of the proposed theoretical bound for addressing the DG problem.

Notably, our bound encourages minimizing the reconstruction loss term, aligning with the recent findings in [68] that encourage (nearly) invertible representation mappings. A more comprehensive discussion on the utilization of the reconstruction loss will be provided in the forthcoming section.

3.2 Part II: Complement for Current Domain Generalization Theory

In the second part, we show that designing models based on domain-invariant feature alone is necessary but insufficient for DG. We theoretically prove the necessity of imposing an information preserving constraint on the representation function. In particular, a reconstruction loss induced by the representation function is desired for preserving most of the relevant information about the label in the representation space. More importantly, we advances previous works [68, 89] by pointing out the trade-off between minimizing the reconstruction loss and achieving domain alignment in DG. Our theoretical results motivate a new DG framework that jointly optimizes the reconstruction loss and the domain discrepancy.

3.2.1 Introduction

Domain-invariant or domain-alignment representation learning and is widely considered as one of the most promising and efficient approaches in DG [137, 151]. Without any knowledge about the unseen domains, if one can learn the domain-invariant feature, *i.e.*, features are general and transferable between domains, the corresponding classifier trained based on these features is likely to perform well on the unseen domains. However, recent work [68] has shown that domain-invariant representation learning may not fully address the information loss caused by non-invertible representation maps. This finding has led to the motivation for exploring the use of (nearly) invertible

representation maps [68]. Similarly, in [89], the authors developed a novel upper bound for the risk of the unseen domain by encouraging a small reconstruction loss induced by the representation function.

In this section, we study domain-invariant representation learning from the information-theoretic perspective. We point out the necessity of imposing a constraint on the representation function to retain the relevant information about the label in extracted features, aligning with the results of previous works [68, 89]. Furthermore, we demonstrate that there is a trade-off between minimizing the reconstruction loss and minimizing the discrepancy between domains.

3.2.2 Contributions

In this work, our contributions include:

1. We derive a lower bound on mutual information between the latent representation and their labels to demonstrate the necessity of imposing a information preserving constraint on the representation function in DG [Proposition 3.2.5].

2. We characterize the trade-off between minimizing the reconstruction loss *vs.* minimizing the discrepancy of joint distributions between domains. In other words, we show that it is impossible to perfectly accomplish these two objectives at the same time [Proposition 3.2.7].

3. We propose a new DG learning framework that directly accounts for both the reconstruction loss and the discrepancy between domains and demonstrate the efficiency of our proposed framework on several datasets.

3.2.3 Related work

As discussed in Chapter 2, domain-invariant representation learning is the most common approach in DG. It aims to extract domain-invariant features and subsequently

design a classifier based on these features [4, 5, 12, 43, 51, 80, 88, 93, 150]. However, the domain-invariant approach relies on two key assumptions: (a) the domain-invariant features must exist and be shared between domains [12], and (b) the domain-invariant features must be strongly correlated with labels [26, 97, 101]. Moreover, obtaining precise domain-invariant features often requires a sufficiently large number of seen domains during training [32, 101].Therefore, if (a) the invariant features are neither existent nor strongly correlated with the label, or (b) the number of observed (seen) domains is not large enough, domain-invariant methods may fail [17, 57, 69, 111].

Domain-invariant representation learning can be categorized into two main branches: (a) marginal distribution-invariant methods, *i.e.*, learning the features such that their distributions are unchanged according to domains, and (b) conditional distribution-invariant methods, *i.e.*, learning the features such that the conditional distributions of labels given features are stable from domain to domain. The first branch includes works such as [23, 46, 52, 60, 67, 85, 96, 112, 118]. For instance, in [96] and [46], deep neural networks are employed to learn transformations that minimize the distributional variances of transformed features over the seen domains. Similarly, Li *et al.* [85] propose a method that minimizes the Maximum Mean Discrepancy (MMD) between marginal distributions in seen domains. Sun and Saenko [125] introduce a method that not only matches the mean but also synchronizes the covariance of feature distributions across different domains. Shen *et al.* [118] minimize the Wasserstein distance between marginal distributions of representation variables from different seen domains in latent space to extract invariant features. Bui *et al.* [26] aim to learn domain-invariant features (with marginal distributions unchanged according to domains) along with domain-specific features to enhance generalization performance. Works going to the second branch includes [4, 5, 12, 80, 88, 114, 138, 149]. Particularly, linear/non-linear Invariant Risk Minimization algorithms are proposed in [12, 88] with the goal to find a common optimal linear/non-linear classifier over all observed domains. These

methods are based on a key assumption that a common optimal classifier exists if the conditional distributions of the label given the learned feature are stable from domain to domain. Additionally, Li *et al.* [80] propose to extract domain-invariant features via information bottleneck scheme together with minimizing the mutual information between label and domain information given extracted feature. Wang *et al.* [138] minimize the Kullback–Leibler (KL) divergence between conditional distributions in each class to obtain domain-invariant features.

It is worth noting that domain-invariant methods may fail under some specific settings, such as cases when the labels is more correlated with the spurious features than with the true invariant features [4,101]. To address the potential failure of learning models in these scenarios, Ahuja *et al.* propose to add a constraint on the entropy of extracted features for capturing the true invariant features [4]. Similarly, Nguyen *et al.* utilize the principle of conditional entropy minimization to eliminate the influence of spurious-invariant features [101]. While the above works have achieved promising results, we theoretically show that learning a domain-invariant representation function itself is necessary but insufficient for DG in the following sections.

3.2.4 Problem Formulation

3.2.4.1 Notations

Following the notations we defined in Chapter 1 and 2, let $\mathcal{X}$, $\mathcal{Z}$, $\mathcal{Y}$ denote the input space, the representation space, and the label space, respectively. For a given family of domains $\mathcal{D}$, suppose that the data from S observed (seen) domains $\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots, \mathcal{D}^{(S)} \in \mathcal{D}$ is accessible, DG tasks aim to learn a representation function $f : \mathcal{X} \to \mathcal{Z}$ followed by a classifier $g : \mathcal{Z} \to \mathcal{Y}$ that generalizes well on an unseen domain $\mathcal{D}^{(u)} \in \mathcal{D}$, $u \neq 1, 2, \ldots, S$.

We denote the input random variable as X, the extracted feature random variable as $Z = f(X)$, and the label random variable as Y, in the input space, representation

space, and label space, respectively. Superscription i is used to denote the variables and functions specified on domain $\mathcal{D}^{(i)}$. Specifically, $p^{(i)}(\boldsymbol{x})$, $p^{(i)}(\boldsymbol{z})$, and $p^{(i)}(\boldsymbol{x}, \boldsymbol{z})$ represent the distribution of input sample $\boldsymbol{x}$, the distribution of feature sample $\boldsymbol{z} = f(\boldsymbol{x})$, and their joint distribution on $\mathcal{D}^{(i)}$, respectively. Moreover, we utilize $p^{(i)}(X, Z)$ and $p^{(i)}(Y, Z)$ to denote the joint distribution between the input random variable X and its representation random variable Z, and the joint distribution between the label random variable Y and the representation random variable Z within $\mathcal{D}^{(i)}$. Finally, we adopt $H(A|B)$ and $I(A; B)$ to represent the conditional entropy and mutual information between two random variables A and B, respectively.

3.2.4.2 Problem Formulation

For given S seen domains, a DG task aims to find an optimal representation function f^* by solving the following optimization problem:

$$\min_{f:\mathcal{X}\to\mathcal{Z}} \quad \mathrm{R}^{(u)}(g_f \circ f) \tag{3.41}$$

Here, $\mathrm{R}^{(u)}(g_f \circ f)$ denotes the risk (classification error) introduced by the representation map f followed by an optimal classifier g_f on the unseen domain $D^{(u)}$. It is important to note that for a given $f : \mathcal{X} \to \mathcal{Z}$, the optimal classifier $g_f : \mathcal{Z} \to \mathcal{Y}$ completely depends on f.

In this work, from the information-theoretic point of view, we aim to solve the following optimization problem:

$$\max_{f:\mathcal{X}\to\mathcal{Z}} \quad I^{(u)}(Y; Z) \tag{3.42}$$

where $I^{(u)}(Y; Z)$ denotes mutual information between the labels and representation features on unseen domain $D^{(u)}$. It is worth noting that a higher mutual information between features and corresponding labels is likely to result in a higher classification

accuracy. Hence, solving (3.42) serves as a proxy for minimizing the classification risk on unseen domains, which ultimately aligns with the primary objective of DG task as defined in (3.41).

3.2.5 Preliminary

This section provides some definitions and preliminary results that support our main results in Section 3.2.6.

3.2.5.1 Measure of Domain Discrepancy

As we discussed above, learning domain-invariant features is a widely adopted strategy to tackle the DG problem. This prevalent approach typically involves two steps. Firstly, one learns domain-invariant features using data and labels from the seen domains, and subsequently, a classifier is designed based on these extracted features [4, 5, 12, 43, 51, 80, 88, 93, 150]. For a given divergence measure $D(\cdot||\cdot)$ and a seen domain $\mathcal{D}^{(s)}$, previous studies on DG usually aim to (a) align marginal distributions of the representation, *i.e.*, minimize $D(p^{(u)}(Z)||p^{(s)}(Z))$, or (b) enforce the conditional distribution to be the same , *i.e.*, minimizing $D(p^{(u)}(Y|Z)||p^{(s)}(Y|Z))$. Under DG settings, one cannot access the distribution of the unseen domain, thus, as a practical workaround, aligning the distribution across all seen domains is commonly adopted as a proxy to achieve this goal. In contrast to the aforementioned approaches, our focus is on learning a mapping f that minimizes the discrepancy between the joint distributions of the seen and unseen domains. While this condition may seem restrictive, we provide the following examples to demonstrate that aligning either the marginal distribution or the conditional distribution alone is insufficient to guarantee a small classification risk on the unseen domain.

Example 3.2.1 (Alignment of marginal distribution alone is not enough)**.** *Suppose that there exits a mapping $f : \mathcal{X} \to \mathcal{Z}$ such that the marginal distributions of seen*

and unseen domains in the latent space are perfectly aligned. Particularly, we assume that $\mathcal{Z} = \{0, 1\}$*, and* $p^{(s)}(Z = 0) = p^{(u)}(Z = 0) = p^{(s)}(Z = 1) = p^{(u)}(Z = 1) = 0.5$*. Now, suppose that there is a mismatch between the conditional distribution between two domains, for example,*

$$p^{(s)}(Y = 0|Z = 0) = 0.9,$$
$$p^{(s)}(Y = 1|Z = 0) = 0.1,$$
$$p^{(s)}(Y = 0|Z = 1) = 0.1,$$
$$p^{(s)}(Y = 1|Z = 1) = 0.9,$$

and

$$p^{(u)}(Y = 0|Z = 0) = 0.1,$$
$$p^{(u)}(Y = 1|Z = 0) = 0.9,$$
$$p^{(u)}(Y = 0|Z = 1) = 0.9,$$
$$p^{(u)}(Y = 1|Z = 1) = 0.1.$$

If one trains a maximum likelihood classifier $g : \mathcal{Z} \to \mathcal{Y}$ *on seen domain, then the obtained classifier will produce:* $g(Z = 0) = 0$ *and* $g(Z = 1) = 1$*. The classification error on the seen domain induced by* f *and* g *is thus:*

$$\begin{aligned} \mathrm{R}^{(s)}(g \circ f) &= p^{(s)}(Z = 0)\big[1 - p^{(s)}(Y = 0|Z = 0)\big] \\ &+ p^{(s)}(Z = 1)\big[1 - p^{(s)}(Y = 1|Z = 1)\big] \\ &= 0.1. \end{aligned}$$

If one applies this classifier to the unseen domain, the classification error is:

$$\begin{aligned} \mathrm{R}^{(u)}(g \circ f) &= p^{(u)}(Z=0)\big[1 - p^{(u)}(Y=0|Z=0)\big] \\ &+ p^{(u)}(Z=1)\big[1 - p^{(u)}(Y=1|Z=1)\big] \\ &= 0.9. \end{aligned}$$

Therefore, only aligning the marginal distribution of the representations alone is not sufficient to guarantee a low classification error on unseen domain.

Example 3.2.2 (Conditional distribution alignment alone is not enough)**.** *Suppose that there exits a mapping $f : \mathcal{X} \to \mathcal{Z}$ such that the conditional distributions of seen and unseen domains in the latent space are perfectly aligned. Particularly, we assume that $\mathcal{Z} = \{0, 1\}$, and*

$$\begin{aligned} p^{(s)}(Y=0|Z=0) &= p^{(u)}(Y=0|Z=0) = 0.9, \\ p^{(s)}(Y=1|Z=0) &= p^{(u)}(Y=1|Z=0) = 0.1, \\ p^{(s)}(Y=0|Z=1) &= p^{(u)}(Y=0|Z=1) = 0.49, \\ p^{(s)}(Y=1|Z=1) &= p^{(u)}(Y=1|Z=1) = 0.51. \end{aligned}$$

Now, suppose that there is a mismatch between the marginal distribution of two domains. Specifically, we have $p^{(s)}(Z=0) = 0.9$, $p^{(s)}(Z=1) = 0.1$ while $p^{(u)}(Z=0) = 0.1$, $p^{(u)}(Z=1) = 0.9$.

If one trains a maximum likelihood classifier $g : \mathcal{Z} \to \mathcal{Y}$ on seen domain, then we will have a classifier g such that $g(Z=0) = 0$ and $g(Z=1) = 1$. The classification

error on seen domain induced by f and g is:

$$\begin{aligned} \mathrm{R}^{(s)}(g \circ f) &= p^{(s)}(Z=0)\big[1 - p^{(s)}(Y=0|Z=0)\big] \\ &+ p^{(s)}(Z=1)\big[1 - p^{(s)}(Y=1|Z=1)\big] \\ &= 0.139. \end{aligned}$$

If one applies this classifier to the unseen domain, the classification error is:

$$\begin{aligned} \mathrm{R}^{(u)}(g \circ f) &= p^{(u)}(Z=0)\big[1 - p^{(u)}(Y=0|Z=0)\big] \\ &+ p^{(u)}(Z=1)\big[1 - p^{(u)}(Y=1|Z=1)\big] \\ &= 0.451. \end{aligned}$$

Therefore, solely targeting the conditional distribution alignment will not ensure a low classification error on unseen domain.

As shown from examples above, we demonstrate that either aligning the marginal distribution or the conditional distribution alone is not enough. Therefore, we propose to align the joint distribution of representation and label. To quantify the discrepancy between the distributions, we introduce the following measure:

Definition 3.2.1 (Domain discrepancy induced by a representation function)**.** For a representation function $f : \mathcal{X} \to \mathcal{Z}$, unseen domain $\mathcal{D}^{(u)}$, and seen domain $\mathcal{D}^{(s)}$, the domain-discrepancy between $\mathcal{D}^{(u)}$ and $\mathcal{D}^{(s)}$ induced by f is:

$$K(f) = D(p^{(u)}(Y, Z)||p^{(s)}(Y, Z)) \tag{3.43}$$

where $D(\cdot||\cdot)$ is a divergence measure that quantifies the mismatch between two distributions.

If the representation function f induces $K(f) = 0$, the distributions between seen

and unseen domains are perfectly aligned. In practice, enforcing $K(f) = 0$ is usually too strict, one may want to release such constraint to $K(f) \leq \epsilon$ where ϵ is a positive number.

Definition 3.2.2. Define $W(\epsilon)$ as the maximum discrepancy between mutual information of unseen domain $\mathcal{D}^{(u)}$ and seen domain $\mathcal{D}^{(s)}$ while the domain discrepancy $K(f)$ does not exceed a positive number ϵ. Formally,

$$W(\epsilon) = \max_{\substack{f:\mathcal{X}\to\mathcal{Z}, \\ K(f)\leq\epsilon}} \left| I^{(u)}(Y;Z) - I^{(s)}(Y;Z) \right| \tag{3.44}$$

where $I^{(u)}(Y;Z)$ and $I^{(s)}(Y;Z)$ are mutual information between label Y and representation feature Z in unseen domain and seen domain, respectively.

If $\epsilon = 0$, we will have $K(f) = 0$ and $I^{(u)}(Y;Z) = I^{(s)}(Y;Z)$, thus $W(0) = 0$. In addition, it is possible to verify that $W(\epsilon)$ is a monotonically increasing function of ϵ.

3.2.5.2 Measure of the Reconstruction Loss

Note that by Data Processing Inequality [34] and the fact that $Y \to X \to Z$ forms a Markov chain, for any representation function f:

$$I^{(u)}(Y;X) \geq I^{(u)}(Y;Z) \tag{3.45}$$

where $I^{(u)}(Y;X)$ denotes mutual information between label and input and $I^{(u)}(Y;Z)$ stands for mutual information between label and feature on unseen domain, respectively. The equality holds in (3.45) if f is invertible. It is worth noting that there may exist non-invertible representation functions that make the equality happens. Indeed, if the label information can be precisely preserved under mapping f, *i.e.*, using Z to predict Y is as good as using X to predict Y, for example, if $H^{(u)}(Y|X) = H^{(u)}(Y|Z)$, then $I^{(u)}(Y;X) = I^{(u)}(Y;Z)$. However, under DG settings, there is no information

about the data, nor the label from unseen domains. Thus, it is impossible to design such non-invertible mappings that perfectly preserve the label information on unseen domain. On the other hand, (nearly) invertible mappings can be constructed without any knowledge of the domains. This can be achieved by minimizing the reconstruction loss between the input data and its representation, providing a feasible way to retain the useful information of the data. In other words, this approach allows us to make $I^{(u)}(Y;Z)$ close to $I^{(u)}(Y;X)$.

Definition 3.2.3 (Reconstruction loss)**.** For a representation function $f : \mathcal{X} \to \mathcal{Z}$, and a function $\psi : \mathcal{Z} \to \mathcal{X}$, the reconstruction loss (on unseen domain) induced by f and ψ is defined by:

$$\begin{aligned} R(f,\psi) &= \int_{\boldsymbol{x}\in\mathcal{X}} p^{(u)}(\boldsymbol{x})\,\ell(\boldsymbol{x},\psi(f(\boldsymbol{x})))\,d\boldsymbol{x} \\ &= \int_{\boldsymbol{x}\in\mathcal{X}}\int_{\boldsymbol{z}\in\mathcal{Z}} p^{(u)}(\boldsymbol{x},\boldsymbol{z})\,\ell(\boldsymbol{x},\psi(\boldsymbol{z}))\,d\boldsymbol{x}\,d\boldsymbol{z} \end{aligned} \tag{3.46}$$

where $\ell(\cdot,\cdot)$ is a distortion function.

Definition 3.2.4. Let $Q(\gamma)$ denote the maximum mutual information loss (on unseen domain) when the reconstruction loss induced by the representation function f and the reconstruction function ψ does not exceed a positive number γ. Formally,

$$Q(\gamma) = \max_{\substack{f:\mathcal{X}\to\mathcal{Z},\\ \psi:\mathcal{Z}\to\mathcal{X},\\ R(f,\psi)\le\gamma}} I^{(u)}(Y;X) - I^{(u)}(Y;Z). \tag{3.47}$$

Note that $\gamma = 0$ implies f is invertible, leading to $I^{(u)}(Y;X) = I^{(u)}(Y;Z)$ and therefore, $Q(0) = 0$. In addition, it is possible to show that $Q(\gamma)$ is a monotonic increasing function of γ.

3.2.6 Main Results

In this section, we present our main results, showing the necessity of employing the representation functions such that a small reconstruction loss is induced in order to solve the optimization problem in (3.42). More interestingly, we show that there is a trade-off between minimizing the reconstruction loss and aligning the joint distributions between domains.

Proposition 3.2.5 (Main result)**.** *For unseen domain $\mathcal{D}^{(u)}$, seen domain $\mathcal{D}^{(s)}$, and any representation function f and reconstruction function ψ:*

$$I^{(u)}(Y;Z) \geq \max\Big[\, I^{(s)}(Y;Z) - W\big(K(f)\big);\; I^{(u)}(Y;X) - Q\big(R(f,\psi)\big)\Big]. \tag{3.48}$$

Proof. First, from Definition 3.2.2, for a given f:

$$W(K(f)) \geq I^{(s)}(Y;Z) - I^{(u)}(Y;Z), \tag{3.49}$$

which is equivalent to:

$$I^{(u)}(Y;Z) \geq I^{(s)}(Y;Z) - W(K(f)). \tag{3.50}$$

Next, from Definition 3.2.4, for given f and ψ:

$$Q(R(f,\psi)) \geq I^{(u)}(Y;X) - I^{(u)}(Y;Z) \tag{3.51}$$

which is equivalent to:

$$I^{(u)}(Y;Z) \geq I^{(u)}(Y;X) - Q(R(f,\psi)). \tag{3.52}$$

Combine (3.50) and (3.52), the proof follows. □

Proposition 3.2.5 points out a possible way to solve the optimization problem proposed in (3.42). Particularly, to maximize $I^{(u)}(Y;Z)$, one needs to (a) maximize $I^{(s)}(Y;Z) - W(K(f))$, and (b) maximize $I^{(u)}(Y;X) - Q(R(f,\psi))$, simultaneously. Since $I^{(u)}(Y;X)$ is a constant and $W(\cdot)$ and $Q(\cdot)$ are monotonically increasing functions, to maximize $I^{(u)}(Y;Z)$, we need to find a representation function f and a reconstruction function ψ to (i) maximize the mutual information on seen domain $I^{(s)}(Y;Z)$, (ii) minimize the domain discrepancy $K(f)$, and (iii) minimize the reconstruction loss $R(f,\psi)$, at the same time.

In practice, if the invariant features exist, strongly correlate with the label, and can be precisely learned, there may exist a mapping f such that $I^{(s)}(Y;Z)$ is large and $K(f)$ is small which make the first lower bound $I^{(s)}(Y;Z) - W(K(f))$ is tighter than the second lower bound $I^{(u)}(Y;X) - Q(R(f,\psi))$. However, certain failure cases reported in the literature [17, 57, 69, 111] reveal scenarios where the invariant feature is not strongly correlated with the label due to the interference of some spurious features, leading to a large $K(f)$ and a small $I^{(s)}(Y;Z)$, thus making the second lower bound $I^{(u)}(Y;X) - Q(R(f,\psi))$ the tighter one. Traditional approaches that aim to learn the invariant features (minimizing $K(f)$) and minimize the empirical risk (a proxy for maximizing the mutual information on the seen domain $I^{(s)}(Y;Z)$) for optimizing the first lower bound can fail in such cases.

Motivated by Proposition 3.2.5, it is natural to pursue a representation function f and a reconstruction function ψ that can simultaneously minimize both $K(f)$ and $R(f,\psi)$. However, we demonstrate below that it is impossible to optimize $K(f)$ and $R(f,\psi)$ at the same time.

Definition 3.2.6 (Reconstruction-alignment function)**.** For unseen domain $\mathcal{D}^{(u)}$, seen domain $\mathcal{D}^{(s)}$, and a given reconstruction function ψ, the reconstruction-alignment

function $T(\gamma)$ is defined by:

$$\begin{aligned} T(\gamma) &= \min_{f:\mathcal{X}\to\mathcal{Z}} K(f) = \min_{f:\mathcal{X}\to\mathcal{Z}} D(p^{(u)}(Y,Z)||p^{(s)}(Y,Z)) \\ \text{s.t.} \quad R(f,\psi) &= \int_{\boldsymbol{x}\in\mathcal{X}} \int_{\boldsymbol{z}\in\mathcal{Z}} p^{(u)}(\boldsymbol{x},\boldsymbol{z})\ell(\boldsymbol{x},\psi(\boldsymbol{z}))\, d\boldsymbol{x}\, d\boldsymbol{z} \leq \gamma \end{aligned} \tag{3.53}$$

where γ is a positive number, $\ell(\cdot,\cdot)$ is a distortion measure, and $D(\cdot||\cdot)$ is a divergence measure.

The reconstruction-alignment function $T(\gamma)$ is the minimal discrepancy between the joint distributions of the unseen domain $\mathcal{D}^{(u)}$ and seen domain $\mathcal{D}^{(s)}$ that can be obtained if the reconstruction loss (on unseen domain) does not exceed a positive number γ. We formally characterize the trade-off between minimizing reconstruction loss and achieving domain alignment as below.

Proposition 3.2.7 (Main result)**.** *If the divergence measure $D(a||b)$ is convex (in both variables a and b), then $T(\gamma)$ defined in (3.53) is (a) monotonically non-increasing, and (b) convex.*

Proof. Our proof closely follows to the proof of rate-distortion theory in [34]. Specifically, consider two positive numbers γ_1 and γ_2, and assume that $\gamma_1 \leq \gamma_2$. For a given reconstruction function ψ, let $\mathcal{F}_{\gamma_1}$ and $\mathcal{F}_{\gamma_2}$ denote the sets of representation functions f such that $R(f,\theta) \leq \gamma_1$ and $R(f,\theta) \leq \gamma_2$, respectively. Since $\gamma_1 \leq \gamma_2$, $\mathcal{F}_{\gamma_1} \subset \mathcal{F}_{\gamma_2}$, we have:

$$T(\gamma_1) = \min_{f\in\mathcal{F}_{\gamma_1}} K(f) \geq \min_{f\in\mathcal{F}_{\gamma_2}} K(f) = T(\gamma_2). \tag{3.54}$$

Thus, $T(\gamma)$ is a monotonically non-increasing function of γ. Next, let:

$$f_1 = \operatorname*{argmin}_{f:\mathcal{X}\to\mathcal{Z}} K(f) \quad \text{s.t.} \qquad R(f,\theta) \leq \gamma_1, \tag{3.55}$$

$$f_2 = \operatorname*{argmin}_{f:\mathcal{X}\to\mathcal{Z}} K(f) \quad \text{s.t.} \qquad R(f,\theta) \leq \gamma_2. \tag{3.56}$$

$p_1^{(u)}(Y,Z)$, $p_1^{(s)}(Y,Z)$ be the corresponding joint distributions of Y and Z on unseen and seen domain introduced by f_1, and $p_2^{(u)}(Y,Z)$, $p_2^{(s)}(Y,Z)$ be the corresponding joint distributions of Y and Z on unseen and seen domain introduced by f_2, respectively. Similarly, let $p_1^{(u)}(X,Z)$, $p_1^{(s)}(X,Z)$ be the corresponding joint distributions of X and Z on unseen and seen domain introduced by f_1, and $p_2^{(u)}(X,Z)$, $p_2^{(s)}(X,Z)$ be the corresponding joint distributions of X and Z on unseen and seen domain introduced by f_2, respectively.

Note that for any representation function f, we have $p^{(u)}(Y,Z) = p^{(u)}(Y|X)\, p^{(u)}(X,Z)$ and $p^{(s)}(Y,Z) = p^{(s)}(Y|X)\, p^{(s)}(X,Z)$ where $p^{(u)}(Y|X)$ and $p^{(s)}(Y|X)$ denote the conditional distribution between label and input data of unseen and seen domain, respectively. Since $p^{(u)}(Y|X)$ and $p^{(s)}(Y|X)$ do not depend on the representation function f. Thus,

$$p_1^{(u)}(Y,Z) = p^{(u)}(Y|X)\, p_1^{(u)}(X,Z), \tag{3.57}$$

$$p_2^{(u)}(Y,Z) = p^{(u)}(Y|X)\, p_2^{(u)}(X,Z), \tag{3.58}$$

and,

$$p_1^{(s)}(Y,Z) = p^{(s)}(Y|X)\, p_1^{(s)}(X,Z), \tag{3.59}$$

$$p_2^{(s)}(Y,Z) = p^{(s)}(Y|X)\, p_2^{(s)}(X,Z). \tag{3.60}$$

Next, to prove the convexity of $T(\gamma)$, we will show that:

$$\lambda T(\gamma_1) + (1-\lambda)T(\gamma_2) \geq T(\lambda\gamma_1 + (1-\lambda)\gamma_2), \tag{3.61}$$

for any $\lambda \in [0,1]$. Let:

$$p_\lambda^{(u)}(X,Z) = \lambda p_1^{(u)}(X,Z) + (1-\lambda) p_2^{(u)}(X,Z), \tag{3.62}$$

$$p_\lambda^{(s)}(X,Z) = \lambda p_1^{(s)}(X,Z) + (1-\lambda) p_2^{(s)}(X,Z). \tag{3.63}$$

By definition, the left hand side of (3.61) can be rewritten by:

$$
\begin{aligned}
& \lambda T(\gamma_1) + (1-\lambda) T(\gamma_2) \\
= \; & \lambda D(p_1^{(u)}(Y,Z) \,||\, p_1^{(s)}(Y,Z)) + (1-\lambda) D(p_2^{(u)}(Y,Z) \,||\, p_2^{(s)}(Y,Z)) \\
= \; & \lambda D(p^{(u)}(Y|X) p_1^{(u)}(X,Z) \,||\, p^{(s)}(Y|X) p_1^{(s)}(X,Z)) && (3.64) \\
+ \; & (1-\lambda) D(p^{(u)}(Y|X) p_2^{(u)}(X,Z) \,||\, p^{(s)}(Y|X) p_2^{(s)}(X,Z)) && (3.65) \\
\geq \; & D(p^{(u)}(Y|X) p_\lambda^{(u)}(X,Z) \,||\, p^{(s)}(Y|X) p_\lambda^{(s)}(X,Z)) && (3.66)
\end{aligned}
$$

where (3.64) and (3.65) are due to (3.57), (3.58), (3.59), and (3.60); (3.66) is due to (3.62), (3.63), and the convexity of $D(\cdot||\cdot)$.

Let f_λ is the corresponding function that induces the joint distribution $p_\lambda^{(u)}(X,Z)$ and $p_\lambda^{(s)}(X,Z)$[1], the reconstruction loss corresponding to f_λ is:

$$\gamma_\lambda = \int_{\boldsymbol{x}\in\mathcal{X}} \int_{\boldsymbol{z}\in\mathcal{Z}} p_\lambda^{(u)}(\boldsymbol{x},\boldsymbol{z}) \ell(\boldsymbol{x}, \psi(\boldsymbol{z}))\, d\boldsymbol{x} d\boldsymbol{z}. \tag{3.67}$$

By Definition 3.2.6,

$$D(p^{(u)}(Y|X)\, p_\lambda^{(u)}(X,Z) \,||\, p^{(s)}(Y|X)\, p_\lambda^{(s)}(X,Z)) \geq T(\gamma_\lambda). \tag{3.68}$$

Combine (3.66) and (3.68):

$$\lambda T(\gamma_1) + (1-\lambda) T(\gamma_2) \geq T(\gamma_\lambda), \tag{3.69}$$

or the left hand side of (3.61) is larger or at least equal to $T(\gamma_\lambda)$. Next, we show that $T(\gamma_\lambda)$ is at least as large as the right hand side of (3.61). Particularly, we want to show:

$$T(\gamma_\lambda) \geq T(\lambda\gamma_1 + (1-\lambda)\gamma_2). \tag{3.70}$$

[1]Indeed, we always can construct f_λ that induces $p_\lambda^{(u)}(X,Z)$ and $p_\lambda^{(s)}(X,Z)$ by linear interpolating between f_1 and f_2.

Since $T(\gamma)$ is a monotonically non-increasing function, we want to show that:

$$\gamma_\lambda \leq \lambda\gamma_1 + (1-\lambda)\gamma_2. \tag{3.71}$$

Since,

$$\begin{aligned}
\gamma_\lambda &= \int_{\boldsymbol{x}}\int_{\boldsymbol{z}} p_\lambda^{(u)}(\boldsymbol{x},\boldsymbol{z})\ell(\boldsymbol{x},\psi(\boldsymbol{z}))d\boldsymbol{x}d\boldsymbol{z} && (3.72)\\
&= \int_{\boldsymbol{x}}\int_{\boldsymbol{z}} \Big(\lambda p_1^{(u)}(\boldsymbol{x},\boldsymbol{z}) + (1-\lambda)p_2^{(u)}(\boldsymbol{x},\boldsymbol{z})\Big)\ell(\boldsymbol{x},\psi(\boldsymbol{z}))d\boldsymbol{x}d\boldsymbol{z} && (3.73)\\
&= \lambda\int_{\boldsymbol{x}}\int_{\boldsymbol{z}} p_1^{(u)}(\boldsymbol{x},\boldsymbol{z})\ell(\boldsymbol{x},\psi(\boldsymbol{z}))d\boldsymbol{x}d\boldsymbol{z} && (3.74)\\
&+ (1-\lambda)\int_{\boldsymbol{x}}\int_{\boldsymbol{z}} p_2^{(u)}(\boldsymbol{x},\boldsymbol{z})\ell(\boldsymbol{x},\psi(\boldsymbol{z}))d\boldsymbol{x}d\boldsymbol{z} && (3.75)\\
&\leq \lambda\gamma_1 + (1-\lambda)\gamma_2 && (3.76)
\end{aligned}$$

with (3.72) due to (3.67); (3.73) due to (3.62); (3.74) and (3.75) due to a bit of algebra; (3.76) due to (3.55) and (3.56), respectively.

From (3.71) and (3.76), (3.70) follows. Finally, from (3.69) and (3.70), (3.61) follows. The proof is complete. □

Sharing some similarities with rate-distortion theory [34], Proposition 3.2.7 characterizes the trade-off between minimizing the domain discrepancy $K(f)$ and minimizing the reconstruction loss $R(f,\psi)$. Since Proposition 3.2.7 holds for any reconstruction function ψ, there are no representation function f and reconstruction function ψ that can minimize the domain discrepancy and the reconstruction loss together.

Though the proof of Proposition 3.2.7 is constructed by considering the reconstruction loss on unseen domain, a similar proof applies to seen domains, *i.e.*, there exists a universal trade-off between minimizing the domain discrepancy and minimizing the reconstruction loss, regardless of domains. Lastly, it is important to note that the assumption regarding the convexity of the divergence $D(\cdot||\cdot)$ is not overly restrictive

in practice. Indeed, most of the divergence measures, such as the Kullback-Leibler divergence, are convex [34].

3.2.7 Practical Approach

Motivated by Proposition 3.2.5, our objective is to simultaneously maximize the mutual information $I^{(s)}(Y;Z)$, minimize the reconstruction loss $R(f,\psi)$ induced by the representation function, and reduce the domain discrepancy $K(f)$ between the seen and unseen domains. Since the unseen domain is inaccessible during training in the DG setting, we approximate the minimization of the domain discrepancy term by minimizing the discrepancy between all seen domains. Similarly, we approximate the maximization of the mutual information between label and representation features by minimizing the empirical risk on seen domains, leaving the direct maximization of the mutual information term for future work. Specifically, we objective function is written as:

$$\min_{f,g_f,\psi} \sum_{i=1}^{S} \mathrm{R}^{(i)}(g_f \circ f) + \alpha L_{\text{discrepancy}}(f) + \beta L_{\text{reconstruction}}(f,\psi), \tag{3.77}$$

where the first term is the empirical classification risk over S seen domains, the second term denotes the domain discrepancy, and the third term represents the reconstruction loss. α, and β are two positive hyper-parameters that control the trade-off between minimizing these three loss terms.

Compared to most existing DG works, the main difference of our objective function in (3.77) lies in the inclusion of the reconstruction loss term, which is motivated by Proposition 3.2.5 to retain the information between the latent representation and its labels on the unseen domain. As a result, (3.77) can be practically optimized by incorporating a decoder (to optimize the reconstruction loss) into well-established existing DG models that already handle the empirical risk and domain discrepancy terms. Practically, we employ the following DG methods: Invariant Risk Minimization (IRM)

algorithm [12], Maximum Mean Discrepancy (MMD) algorithm [85], CORrelation ALignment (CORAL) algorithm [125], Invariant Risk Minimization-Maximum Mean Discrepancy (IRM-MMD) algorithm [58], Information Bottleneck-Invariant Risk Minimization (IB-IRM) algorithm [4], Empirical Risk Minimization (ERM) algorithm [130], and Conditional Entropy Minimization (CEM) algorithm [101] to minimize the first two terms in (3.77). To minimize the reconstruction loss term in (3.77), we train a representation function $f : \mathcal{X} \to \mathcal{Z}$ together with a reconstruction function $\psi : \mathcal{Z} \to \mathcal{X}$ to minimize:

$$L_{\text{reconstruction}}(f, \psi) = \sum_{i=1}^{S} \int_{\boldsymbol{x} \in \mathcal{X}} p^{(i)}(\boldsymbol{x}) \ell(\boldsymbol{x}, \psi(f(\boldsymbol{x}))) \, d\boldsymbol{x}, \tag{3.78}$$

where the squared-Euclidean distance is selected as the distortion measure, *i.e.*, $\ell(a, b) = (a - b)^2$, and $p^{(i)}(\boldsymbol{x})$ denotes the input distribution on domain $\mathcal{D}^{(i)}$, $i = 1, 2, \ldots, S$.

By integrating the reconstruction loss term into IRM, MMD, CORAL, IRM-MMD, IB-IRM, ERM, and CEM algorithms, we propose the following variations: IRM-Rec, MMD-Rec, CORAL-Rec, IRM-MMD-Rec, IB-IRM-Rec, ERM-Rec, and CEM-Rec, respectively. Employing multiple algorithms to handle the first two terms in (3.77) offers the advantage of evaluating the effectiveness of combining the reconstruction-loss term with a variety of DG methods. Our numerical results in the next section demonstrate that incorporating the reconstruction loss term leads to improvements in the accuracy of existing DG methods.

3.2.8 Experiments

3.2.8.1 Datasets

Colored-MNIST (CMNIST) [12]. The CMNIST dataset is a common DG dataset which was first proposed in [12]. The learning task is to classify a colored digit into

two classes: the digit is less than or equal to four or the digit is strictly greater than four. There are three domains in CMNIST, two domains contain 25,000 images each and one domain contains 20,000 images. Here, the color is considered as a spurious feature which is added in a way such that the label is more correlated with the color than with the digit. Due to a strong spurious correlation between colors and labels, any algorithm simply aims to minimize the training error will tend to discover the color rather than the shape of the digit (on seen domains) and therefore fail in the test on unseen domains. More details about the CMNIST dataset can be found in [12].

Covariate-Shift-CMNIST (CS-CMNIST) [6]. The CS-CMNIST dataset is a dataset derived from CMNIST dataset which was first introduced in [6]. There are 10 classes in CS-CMNIST dataset where each class corresponds to a digit from zero to nine and each digit is associated with a single color. There are three domains in CS-CMNIST: two training domains and one test domain, each containing 20,000 images. The color is considered the spurious feature and is added in a way such that the color is more correlated to digits on seen domains than on unseen domains. More detail about CS-CMNIST can be found in [4], [6].

3.2.8.2 Implementation Details

For the CMNIST dataset, we utilize the well-established implementation in Domainbed [57] that employs the MNIST-ConvNet with four convolutional layers as the learning model. 20 trials corresponding to 20 pairs of hyper-parameters α and β are randomly selected in $[10^{-1}, 10^{4}]$. For each trial, the learning rate is randomly picked in $[10^{-4.5}, 10^{-3.5}]$ while the batch size is randomly selected in $[2^{3}, 2^{9}]$.

Since the CS-CMNIST dataset is not available in Domainbed [57], we follow the implementation proposed in [4] where the learning model is composed of three convolutional layers with feature map dimensions of 256, 128, and 64, respectively. The last layer (linear layer) is used to classify the colored digit back to 10 classes

corresponding to 10 digits from zero to nine. We use an SGD optimizer for training with a batch size fixed to 128, the learning rate fixed to 10^{-1} and decay every 600 steps with the total number of steps set to 2,000. In contrast to CMNIST, a grid search is performed in CS-CMNIST with $\alpha, \beta \in \{0.1, 1, 10, 10^2, 10^3, 10^4\}$.

The training-domain validation set procedure is used for model selection, *i.e.*, selecting the hyper-parameters (the models) that induce the highest validation accuracy on the validation set sampled from seen domains [4, 57].

We repeat the whole experiment three times for CMNIST and five times for CS-CMNIST via selecting different random seeds[2]. For each selected random seed, the whole process of hyper-parameters tuning and model selection is repeated. After the whole process is completed, only the average accuracy and its corresponding standard deviation are reported. Our code can be found at this link[3].

3.2.9 Results and Discussion

Table 3.10: Average accuracy (%) of compared methods on CS-CMNIST dataset.

Algorithm	IRM [12]	IB-IRM [4]	MMD-IRM [58]	CEM [101]
Accuracy	61.5 ± 1.5	71.8 ± 0.7	77.2 ± 0.9	85.7 ± 0.9
Algorithm	IRM-Rec	IB-IRM-Rec	MMD-IRM-Rec	CEM-Rec
Accuracy	71.0 ± 0.8	75.6 ± 1.1	79.7 ± 0.6	87.1 ± 1.3

Table 3.11: Average accuracy (%) of compared methods on CMNIST dataset.

Algorithm	IRM [12]	MMD [85]	ERM [130]	CORAL [125]
Accuracy	52.0 ± 0.1	51.5 ± 0.2	51.5 ± 0.1	51.5 ± 0.1
Algorithm	IRM-Rec	MMD-Rec	ERM-Rec	CORAL-Rec
Accuracy	51.7 ± 0.2	51.7 ± 0.1	51.8 ± 0.1	52.0 ± 0.1

[2]We follow the settings in [4, 57]. In particular, in [57], the experiment is repeated three times while in [4], the experiment is repeated five times.

[3]`https://github.com/thuan2412/tradeoff_between_domain_alignment_and_reconstruction_loss`

Tables 3.10 and 3.11 present the accuracy of the original DG methods and their variations on the CS-CMNIST dataset and CMNIST dataset, respectively. The numerical results for IRM, MMD, CORAL, and ERM on the CMNIST dataset are gathered from [57] while the numerical results for IRM, IB-IRM, and CEM on the CS-CMNIST dataset are gathered from [101]. Since the source code of MMD-IRM [58] was not released, we implemented this algorithm ourselves to construct the MMD-IRM-Rec algorithm.

For the CS-CMNIST dataset, the accuracy of all four tested algorithms has improved when the reconstruction loss term is added. In particular, the lowest improvement of 1.4% is observed from the CEM algorithm [101], while the largest improvement of 9.5% appears in the IRM algorithm [12]. We believe this variation in improvement can be attributed to Proposition 3.2.5. It seems that the original CEM algorithm [101] already performs well on CS-CMNIST, and hence, the first lower bound in Proposition 3.2.5 induced by CEM is relatively tight, resulting in a small accuracy improvement when the reconstruction loss term is added. In contrast, since the IRM algorithm [12] exhibits poorer performance on CS-CMNIST, we suspect that the first lower bound in Proposition 3.2.5 induced by IRM is the looser one, leading to a significant improvement when optimizing the second bound by adding the reconstruction loss term.

In comparison, CMNIST is a more challenging dataset, where all algorithms tested perform poorly due to the strong spurious correlation between colors and the labels of digits [57]. However, as observed in Table 3.11, three out of four tested algorithms show improvement when the reconstruction loss term is added. Although the improvement is not substantial, with the largest margin being only 0.5% observed in the CORAL algorithm, this still demonstrates the usefulness of optimizing the reconstruction loss term in DG.

Finally, our future work will focus on integrating the reconstruction loss into other

state-of-the-art DG algorithms, and using mutual information as a direct objective function instead of empirical risk.

3.2.10 Conclusions

In this part of work, we have demonstrated that while learning domain-invariant representation features is necessary in DG, it is not sufficient to preserve the mutual information between labels and representation features on unseen domains. To address this limitation, we introduce a constraint on the representation function by adding a reconstruction loss between the input and its reconstruction from the extracted feature, which helps retain essential information about the labels. Additionally, we highlight the inherent trade-off between minimizing the reconstruction loss and achieving domain alignment in DG. This observation implies that simultaneously minimizing both the reconstruction loss and the domain discrepancy is not feasible. Building on these theoretical insights, we present a new practical framework that jointly considers both the reconstruction loss and the domain discrepancy to learn representation features. Importantly, our proposed framework can be easily adapted to different DG algorithms. Moreover, it demonstrates improved performance compared to state-of-the-art DG methods in practice. These findings provide valuable guidance for the development of more effective DG approaches with enhanced generalization performance.

Chapter 4

Spurious Domain-invariant Features and Domain Generalization

Building upon the insights presented in Chapter 2, it is clear that invariant features play a pivotal role in tackling the DG problem. However, as highlighted in various domain generalization studies [5,6,12], algorithms seeking to learning domain-invariant features may still fail in real-world scenario, despite the promising guidance offered by theoretical works [21, 146]. In this chapter, we take a closer look at the failure cases of the DG algorithms and analyze their underlying causes. In particular, we identify that the algorithms' downfall lies in their inability to discriminate the true invariant feature from the spurious invariant features. To address this problem, we introduce a novel framework grounded in conditional entropy minimization (CEM). This framework serves to effectively filter out spurious invariant features, ultimately enhancing the robustness of model.

4.1 Introduction

In previous chapters, we can see that learning domain-invariant features is generally used as a primary solution to the DG problem. Following this idea, various DG

algorithms have been proposed for domain-invariant feature learning [137, 151] over the past decade. Among these algorithms, Invariant Risk Minimization (IRM) [12, 88] provides an innovative perspective for formulating and solving the domain generalization problem. Under the widely used assumptions that representations are general and transferable if the feature representations remain invariant from domain to domain, IRM casts fresh light on the interplay between classifier performance on each domain and feature transferability. However, despite its effectiveness, IRM has been found to fail in some simple settings where spurious invariant features exist [17, 57, 69, 111]. A widely known example is the cow and camel classification problem [26, 97], where the label is a deterministic function of the invariant features like the shape of the animals, and independent of spurious attributes, such as background. Although cows commonly appear in grassy settings and camels mostly in desert landscapes with yellow backgrounds, no matter which domain they come from, the background color could inadvertently be treated as an invariant feature and captured by the model. This may not pose an issue when test data aligns with these conditions, but it can significantly increase the classification error when cows appear in a yellow background or camels are placed on a green field. Therefore, even though an invariance principle-based approach can effectively learn invariant features, its success in DG classification tasks can be compromised if the extracted features contain not only the true invariant features but also spurious invariant features. Although these spurious features could be removed if one can observe a sufficiently large number of domains [32, 111], for example, if the seen domain contains a picture of a cow walking in a desert, it is impossible to exhaust all possible domains.

In response to this challenge of spurious features, Ahuja *et al.* propose an approach that leverages feature entropy minimization to effectively eliminate these spurious features [5]. This technique draws inspiration from the Information Bottleneck (IB) framework [127]. Nonetheless, it is important to note that their method is constrained

to linear classifiers, thereby limiting its applicability. Additionally, while the approach is inspired by the IB framework, it doesn't directly integrate the IB principles into the algorithm. Subsequently, two alternative approaches [43, 81] embrace IB objectives directly to combat the presence of spurious invariant features. However, it is worth noting that their methods are largely heuristically motivated and lack theoretical justification.

In this chapter, we introduce an innovative framework grounded on conditional entropy minimization to effectively filter out spurious domain-invariant features. Moreover, we establish a direct correlation between our objective function and the Deterministic Information Bottleneck (DIB) principle [123].

4.1.1 Main Contributions

The key contributions of this chapter can be outlined as follows:

- We propose a new objective function motivated by the conditional entropy minimization (CEM) principle. Moreover, we establish a direct link between the proposed objective and the Deterministic Information Bottleneck (DIB) principle [123].

- We theoretically show that under some assumptions, minimizing the proposed objective function can effectively filter out spurious features.

- Our proposed framework exhibits a broad scope of applicability. It can accommodate non-linear classifiers, extending its usability beyond linear scenarios. Additionally, though we choose IRM as an example to demonstrate the proposed framework, it can be seamlessly integrated into other domain generalization algorithms based on the learning domain-invariant feature principle.

4.2 Related Work

4.2.1 Domain Generalization

Detailed review for DG has been introduced in chapter 2, here we only provide the background information on the information bottleneck [127] and invariant risk minimization [12].

4.2.2 Information Bottleneck and Invariant Risk Minimization

4.2.2.1 Information Bottleneck

Information Bottleneck (IB) framework [127] is a generalization of the rate distortion theory initially introduced by Tishby *et al.*. This framework aims to identify the representation variable Z that retain the information about the label variable Y as much as possible, while simultaneously achieving maximum compression of X. Before diving into the details, we introduce some notations, in line with the conventions set by previous chapters. Consider $f : \mathcal{X} \to \mathcal{Z}$ as a (potentially stochastic) representation mapping from the input data space $\mathcal{X}$ to the representation space $\mathcal{Z}$, and let $g : \mathcal{Z} \to \mathcal{Y}$ denote a labeling function from the representation space $\mathcal{Z}$ to the label space $\mathcal{Y}$. The IB framework aims to find a good representation function f^* by solving the following problem:

$$f^* = \underset{f}{\operatorname{argmin}}\, I(X;Z) - \theta I(Y;Z), \tag{4.1}$$

where $I(X;Z)$ denotes the mutual information between the random variable X, corresponding to the input data, and its representation $Z = f(X)$. $I(Y;Z)$ denotes the mutual information between the random variable Y and Z, corresponding to the label and representation, respectively. θ is a positive hyper-parameter that controls the trade-off between maximizing $I(Y;Z)$ and minimizing $I(X;Z)$.

Mutual information is a non-negative statistical measure of the dependence between

random variables, where a larger value corresponds to stronger dependence and zero means two random variables are independent. By controlling the two mutual information terms, the IB framework aims to find a representation Z that is weakly dependent on input X, but strongly dependent on the prediction label Y. From the information theory point of view, the IB objective can be likened to the concept of indirect rate-distortion source coding. Here Z can be viewed as a "compressed" code of X, where $I(X;Z)$ quantifies the number of "bits" required for compressing X to Z. On the other hand, $I(Y;Z)$ serves as a measure of how well the label Y can be decoded from Z, reflecting the prediction accuracy or "inverse-distortion". Effectively, the IB problem can be reformulated as a Lagrangian optimization, with the aim of minimizing the number of bits needed to compress X, while ensuring accurate recovery of Y from Z to the desired precision. Following a similar idea, Strouse *et. al* propose Deterministic Information Bottleneck (DIB) [123] as an alternative formulation of the IB framework, where a direct restriction of the resources need to represent Z is posed by replacing the mutual information between X and Z with the entropy of Z. The objective of DIB is shown below.

$$f^* = \underset{f}{\operatorname{argmin}}\, H(Z) - \theta I(Y;Z) \tag{4.2}$$

A special case for the DIB is when $\theta = 1$, the objective function becomes $H(Z) - \theta I(Y;Z) = H(Z|Y)$, implying the minimization of the conditional entropy $H(Z|Y)$.

4.2.2.2 Invariant Risk Minimization Algorithm

In this chapter, we choose Invariant Risk Minimization (IRM) [12] algorithm for initial invariant feature extraction due to its promising performance on learning domain-invariant features. The IRM algorithm aims to find the representations that lead to the optimal classifiers applied to these features are also optimal for all domains, as

stated below:

$$\min_{\substack{f\in\mathcal{F}\\ g\in\mathcal{G}}} \sum_{s=1}^{S} R^{(s)}(g\circ f)$$
$$s.t. \quad g\in \operatorname*{argmin}_{\bar{g}} R^{(s)}(\bar{g}\circ f), \quad \text{for all } s\in S \tag{4.3}$$

where $\mathcal{F}$ is a family of representation functions (typically parameterized by weights of a neural network with a given architecture), $\mathcal{G}$ a family of *linear* classifiers (typically the last fully connected classification layer of a neural network), $R^{(s)}(g\circ f) := \mathbb{E}_{(X,Y)\sim\mathcal{D}^{(s)}}[\ell(g(f(X)),Y)]$ denotes a classification risk of using a representation function f followed by a classifier g in domain s under the loss function ℓ. Note that the implicit assumption of the IRM algorithm is that such representations and optimum domain-invariant classifiers exist. In practice, this challenging bi-level optimization problem is approximately realized by solving the following optimization problem [12]:

$$\min_{h\in\mathcal{G}\circ\mathcal{F}} \mathsf{L}_{IRM}(h,\alpha) := \sum_{i=1}^{S}\left[R^{(s)}(h) + \alpha\,\|\nabla_{t|t=1.0}R^{(s)}(t\cdot h)\|^2\right], \tag{4.4}$$

where α is a hyper-parameter associated with the squared Euclidean norm of the gradients (denoted by ∇) of the risks in different domains, $t=1$ is a scalar and fixed "dummy" classifier. When restricted to the family of linear classifiers and convex differentiable risk functions, Theorem 4 of [12] shows (under certain technical assumptions) that minimizing L_{IRM} will produce a predictor that not only (approximately) minimizes cumulative risk across all domains (the first term in L_{IRM}), but is also approximately optimum across all domains, that is, approximately invariant, with the help of the sum of squared gradients of risk across all domains.

Though in this chapter, we ground our framework on the IRM algorithm [12] due to its promising performance on domain-invariant feature extraction and the shared focus on removing spurious features. We note, however, that our approach is

applicable to any method that can learn invariant features.

4.3 Problem Formulation

In this section, we formulate the minimum conditional entropy principle, a special case of the DIB principle, for spurious feature filtration. This formulation is underpinned by three foundational modeling assumptions, which encapsulate two essential ideas: (i) that the learned features are a linear mixture (superposition) of "true" domain-invariant and "spurious" domain-invariant attributes (domain-specific feature), and (ii) that, given the label, the invariant features are conditionally independent of spurious features.

4.3.1 Notation

Consider a classification task where the learning algorithm has access to *i.i.d.* data from the set of S seen domains $\mathbb{D} = \{D_1, D_2, \ldots, D_S\}$. The DG task is to learn a representation function $f : \mathcal{X} \to \mathcal{Z}$ from the input data space $\mathcal{X}$ to the representation space $\mathcal{Z}$, and a classifier $g : \mathcal{Z} \to \mathcal{Y}$ from the representation space $\mathcal{Z}$ to the label space $\mathcal{Y}$ that generalizes well to an unseen domain $D_u \notin \mathbb{D}$.

We use X for the data random variable in input space, Y for the label random variable in label space, and Z for the extracted feature random variable in representation space. The invariant and spurious features are denoted as Z_{inv} and Z_{sp}. We use $\mathbb{E}[\cdot]$, $Var(\cdot)$, $H(\cdot)$, and $I(\cdot)$ for expectation, variance, discrete/differential entropy, and mutual information, respectively.

4.3.2 Assumptions

Ideally, we want the representation function f, such that $f(X) = Z_{\text{inv}}$. However, with a finite number of observed domains, even with well-designed DG algorithms, learned

features may still contain spurious invariant features that remain invariant across observed domains but vary in unseen domains [32, 111]. We model this scenario by assuming that the representation function extracts features that are a combination of the (true) invariant features and the spurious invariant features.

$$f(X) = Z = \Theta(Z_{\text{inv}}, Z_{\text{sp}}).$$

Next, we state three assumptions on $Z_{\text{inv}}, Z_{\text{sp}}$ and Θ that we will use in Section 4.4 to derive our theoretical results.

Assumption 1. The (true) invariant features Z_{inv} are independent of the spurious invariant features Z_{sp} for a given label Y. Formally, $Z_{\text{inv}} \perp\!\!\!\perp Z_{\text{sp}}|Y$.

Assumption 1 is widely accepted in the DG literature [4, 97, 100, 111]. For example, in the construction of the binary-MNIST dataset [97], the label (class) is first chosen, and then color (a spurious feature) is independently added to the hand-written digit (invariant feature) picked from the selected class, ensuring $Z_{\text{inv}} \perp\!\!\!\perp Z_{\text{sp}}|Y$. For more details, we refer the reader to the third constraint in Section 3 of [97]. In [111], [4] and [100], this assumption is used but not explicitly stated. It is, however, implicit in Figure 2 in [4], Figure 3.1 in [111], and the discussion below Figure 2 in [100].

Assumption 2. The uncertainty of the invariant features is lower than the uncertainty of the spurious features when the label is known. Formally, we assume $H(Z_{\text{inv}}|Y) < H(Z_{\text{sp}}|Y)$.

Assumption 2 has the following interesting clustering interpretation: invariant features are better clustered together in each class (have smaller variability) than spurious features. If additionally, $H(Z_{\text{inv}}) = H(Z_{\text{sp}})$, then $I(Z_{\text{inv}}; Y) = H(Z_{\text{inv}}) - H(Z_{\text{inv}}|Y) > H(Z_{\text{sp}}) - H(Z_{\text{sp}}|Y) = I(Z_{\text{sp}}; Y)$, implying that the invariant features Z_{inv} have a stronger connection to the label Y than the spurious features Z_{sp}.

Assumption 3. $f(X) = Z = \Theta(Z_{\text{inv}}, Z_{\text{sp}}) = aZ_{\text{inv}} + bZ_{\text{sp}}$ and $Var(Z|Y) = Var(Z_{\text{inv}}|Y) = Var(Z_{\text{sp}}|Y) = 1$.

Assumption 3 posits that the derived features are a linear combination of invariant and spurious features, *i.e.*, $Z = \Theta(Z_{\text{inv}}, Z_{\text{sp}}) = aZ_{\text{inv}} + bZ_{\text{sp}}$. This concept aligns with the frameworks in [4, 12], which are grounded in Blind Source Separation (BSS) techniques such as Independent Component Analysis (ICA) [63, 98, 104]. Specifically, the objective resembles that of ICA, which seeks to disentangle statistically independent latent components, denoted S_1 and S_2 with $S_1 \perp\!\!\!\perp S_2$, from observations of their *linear combination* $M = a_1S_1 + a_2S_2$.

Our emphasis on the simple linear combination model allows us to derive some insightful theoretical results in the next section. These insights are then translated into an effective algorithm for filtering out spurious features in domain generalization. Note that the more general non-linear dependence relationship between Z and $Z_{\text{inv}}, Z_{\text{sp}}$ could be potentially handled using techniques such as non-linear ICA [64] or non-linear IRM [88]. But we leave this to future work.

The assumption $Var(Z|Y) = Var(Z_{\text{inv}}|Y) = Var(Z_{\text{sp}}|Y) = 1$ is also motivated by the constraint in ICA, which is essential for overcoming the so-called *scaling* ambiguity: if $S_1 \perp\!\!\!\perp S_2$ and $M = a_1S_1 + a_2S_2$, then both (S_1, S_2) and (a_1S_1, a_2S_2) are pairs of independent component sources whose linear combination is M. Finally, it is worth noting that Assumption 1 and Assumption 3 together imply that $a^2 + b^2 = 1$ (see proof of Lemma 4.4.1).

The assumption $Var(Z|Y) = Var(Z_{\text{inv}}|Y) = Var(Z_{\text{sp}}|Y) = 1$ also draws inspiration from constraints in ICA. Such constraint address the *scaling* ambiguity. Specifically, given $S_1 \perp\!\!\!\perp S_2$ and $M = a_1S_1 + a_2S_2$, both pairs (S_1, S_2) and (a_1S_1, a_2S_2) represent independent component sources that yield the linear combination M. Without the assumption on the variance, the ICA algorithm will not be able to identify the amplitude of the components. Importantly, combining Assumption 1 and Assumption 3

imply that $a^2 + b^2 = 1$, as demonstrated in the proof of Lemma 4.4.1.

4.4 Main Results

Our method consists of two core steps. Firstly, we extract invariant features Z from the source domains, which may encompass both true domain-invariant features Z_{inv} and spurious ones Z_{sp}. Then, we filter out these spurious features in order to construct a classifier that purely relies on the true invariant features Z_{inv}. For example, in the "cow-camel setting", the first step identifies all invariant features, potentially including the background color, a spurious feature that will be eliminated in the subsequent step. We now demonstrate that the CEM principle, *i.e.*, minimizing $H(Z|Y)$, supports filtering out the spurious invariant features.

Assumption 4. Let

$$\begin{aligned} f^* &= \operatorname*{argmin}_f \mathsf{L}_{\text{invariant}}(f) \\ s.t. \quad & H(f(X)|Y) \leq \gamma. \end{aligned}$$

where $\mathsf{L}_{\text{invariant}}$ is the loss function of an invariant representation learning algorithm. We assume that $\mathsf{L}_{\text{invariant}}$ is such that for all γ, $Z = f^*(X)$ is a linear superposition of both the invariant feature Z_{inv} and the spurious feature Z_{sp}.

Given Assumption 4, our primary strategy is to "eliminate" Z_{sp} from Z. This is achieved by minimizing $\mathsf{L}_{\text{invariant}}$, while imposing a suitable constraint on the uncertainty of Z conditional on Y, *i.e.*, determining an appropriate value for γ. As we will demonstrate in the following lemma, there exists an optimal choice of γ that allows f^* to exclusively extract the true invariant feature Z_{inv}, while effectively filtering out Z_{sp}.

Lemma 4.4.1. *If Assumptions 1, 2, 3 hold, then*

$$H(Z|Y) = H(aZ_{\text{inv}} + bZ_{\text{sp}}|Y) \geq H(Z_{\text{inv}}|Y) \tag{4.5}$$

and the equality holds in (4.5) if and only if $a = 1$ *and* $b = 0$.

Proof. Our proof of Lemma 4.4.1 is for differential entropy, but it can be easily extended to discrete entropy (recall that we use $H(\cdot)$ to denote discrete or differential entropy). Under Assumptions 1 and 3, we first show that $a^2 + b^2 = 1$.

$$\begin{aligned} 1 &= Var(Z|Y) = Var(aZ_{\text{inv}} + bZ_{\text{sp}}|Y) \\ &= a^2\ Var(Z_{\text{inv}}|Y) + b^2\ Var(Z_{\text{sp}}|Y) \qquad (4.6) \\ &= a^2 + b^2, \qquad (4.7) \end{aligned}$$

where (4.6) is due to $Z_{\text{inv}} \perp\!\!\!\perp Z_{\text{sp}}|Y$ and (4.7) is due to the assumption that $Var(Z_{\text{inv}}|Y) = Var(Z_{\text{sp}}|Y) = 1$.

Next, we utilize the result in Lemma 1 of [132], which states that for any two random variables R_1, R_2, and any two scalars a, b, if $R_1 \perp\!\!\!\perp R_2$ and $a^2 + b^2 = 1$, then:

$$H(aR_1 + bR_2) \geq a^2 H(R_1) + b^2 H(R_2). \tag{4.8}$$

Now, for a given $Y = y \in \mathcal{Y}$, we have:

$$\begin{aligned} &H(aZ_{\text{inv}} + bZ_{\text{sp}}|Y = y) \\ &\geq a^2 H(Z_{\text{inv}}|Y = y) + b^2 H(Z_{\text{sp}}|Y = y) \qquad (4.9) \\ &= a^2 H(Z_{\text{inv}}|Y = y) + b^2 H(Z_{\text{inv}}|Y = y) + b^2 H(Z_{\text{sp}}|Y = y) - b^2 H(Z_{\text{inv}}|Y = y) \\ &= H(Z_{\text{inv}}|Y=y) + b^2\big(H(Z_{\text{sp}}|Y=y) - H(Z_{\text{inv}}|Y=y)\big), \qquad (4.10) \end{aligned}$$

where (4.9) arises from (4.8) and $a^2 + b^2 = 1$. Meanwhile, (4.10) is directly attributed

to $a^2 + b^2 = 1$. Next,

$$
\begin{aligned}
H(Z|Y) &= H(aZ_{\text{inv}} + bZ_{\text{sp}}|Y) \\
&= \int_{y\in\mathcal{Y}} p(y) H(aZ_{\text{inv}} + bZ_{\text{sp}}|Y = y)\, dy \\
&\geq \int_{y\in\mathcal{Y}} p(y) H(Z_{\text{inv}}|Y = y)\, dy + \int_{y\in\mathcal{Y}} p(y) b^2 \big(H(Z_{\text{sp}}|Y=y) - H(Z_{\text{inv}}|Y=y)\big)\, dy \qquad (4.11) \\
&= H(Z_{\text{inv}}|Y) + b^2\big(H(Z_{\text{sp}}|Y) - H(Z_{\text{inv}}|Y)\big) \qquad (4.12) \\
&\geq H(Z_{\text{inv}}|Y) \qquad (4.13)
\end{aligned}
$$

where (4.11) follows from (4.10). (4.13) is derived from $H(Z_{\text{sp}}|Y) > H(Z_{\text{inv}}|Y)$ (Assumption 2). If $a = 1$ and $b = 0$, then $Z = Z_{\text{inv}}$ and the equality holds. On the contrary, if equality holds, then $a = 1$ and $b = 0$ must hold, because otherwise we would have $b^2 > 0$ which together with $H(Z_{\text{sp}}|Y) > H(Z_{\text{inv}}|Y)$ and (4.12) would imply that $H(Z|Y)$ is strictly larger than $H(Z_{\text{inv}}|Y)$. Thus, the equality $H(Z|Y) = H(Z_{\text{inv}}|Y)$ occurs if and only if $a = 1$ and $b = 0$, or equivalently, if and only if $Z = Z_{\text{inv}}$. □

Lemma 4.4.1 shows that $H(Z|Y)$ is always lower bounded by $H(Z_{\text{inv}}|Y)$ and equality occurs if and only if $Z = Z_{\text{inv}}$. We use Lemma 4.4.1 to prove Theorem 4.4.2 which states that the CEM principle can be used to extract the (true) invariant features Z_{inv}.

Theorem 4.4.2. *If Assumptions 1, 2, 3, and 4 hold, then there exits a γ^* such that $f^*(X) = Z_{\text{inv}}$.*

Proof. Given Assumption 4, minimizing $\mathsf{L}_{\text{invariant}}$ for any value of γ results in the relation $Z = aZ_{\text{inv}} + bZ_{\text{sp}}$, where both a and b are functions of γ. Additionally, the inequalities $\gamma \geq H(Z|Y) \geq H(Z_{\text{inv}}|Y)$ hold. The first inequality arises from Assumption 4, while the second is derived from Lemma 4.4.1. Setting $\gamma = \gamma^* := H(Z_{\text{inv}}|Y)$ ensures $H(Z|Y) = H(Z_{\text{inv}}|Y)$. As indicated by Lemma 4.4.1, this equality

holds true if and only if $b = 0$. Therefore, choosing $\gamma^* = H(Z_{\text{inv}}|Y)$ gives us a representation function f^* such that $f^*(X) = Z = Z_{\text{inv}}$. □

4.5 Practical Approach

Given the analysis above, we propose our CEM objective function for extracting the true invariant features as shown below.

$$\min_{h \in \mathcal{G} \circ \mathcal{F}} \mathsf{L}_{CE-IRM}(h, \alpha, \beta) = \mathsf{L}_{IRM}(h, \alpha) + \beta H(f(X)|Y). \tag{4.14}$$

Here, Y denotes the label, $h = g \circ f$ acts as an invariant predictor with $f \in \mathcal{F}$, $g \in \mathcal{G}$, and $Z = f(X)$ is the output of the penultimate layer of the end-to-end neural network. We note that Z and Y represent the latent representations and the labels corresponding to the input data X from all seen domains combined. Indeed, this expression can be seen as the Lagrangian form of the optimization problem described in Assumption 4, where $\mathsf{L}_{\text{invariant}}$ is substituted by the IRM loss function L_{IRM} from (4.4). Moreover, the conditional entropy constraint from Assumption 4 is integrated as the second term, scaled by the Lagrange multiplier β.

To solve the optimization problem in (4.14), we leverage the implementations in [4] and [10]. Since

$$H(Z|Y) = H(Z) + H(Y|Z) - H(Y) \tag{4.15}$$

and $H(Y)$ is a data-dependent constant that is independent of $h = g \circ f$, the CEM optimization problem in (4.14) is equivalent to

$$\min_{h \in \mathcal{G} \circ \mathcal{F}} \mathsf{L}_{IRM}(h, \alpha) + \beta H(f(X)) + \beta H(Y|f(X)). \tag{4.16}$$

The first two terms of the objective function in (4.16) are identical to the objective

function proposed in [4]. We therefore adapt the implementation[1] in [4] to minimize the first two terms. To optimize the third conditional entropy term $H(Y|Z)$, we adopt the variational characterization of conditional entropy described in [10] and the corresponding implementation[2] of the variational method for the minimization of the conditional entropy term.

4.6 Experiments

In this section, we evaluate the efficacy of the proposed method on DG datasets that contain spurious features.

4.6.1 Datasets

CMNIST [12]. The Colored-MNIST or in some literature, referred as Anti-causal-CMNIST dataset, is a synthetic binary classification dataset derived from the MNIST [79] dataset. Initially introduced in [12], it comprises three domains: two seen domains with 25,000 digit images each and one unseen domain with 20,000 test images. The goal is to identify whether the digit is < 5 or ≥ 5 (binary label). Unlike the original MNIST, CMNIST images are colored either red or green, introducing a spurious correlation with the digit labels. This color-label relationship varies across domains: two seen domains have high correlations (0.9 and 0.8), while the unseen test domain has a low correlation (0.1), making color a deliberately spurious invariant feature. For consistency in comparisons, our CMNIST dataset construction aligns with those in [4, 12].

CS-CMNIST [6]. Derived from the CMNIST dataset, the Covariate-Shift-CMNIST is a synthetic classification dataset with three domains: two for training and one unseen for testing, each containing 20,000 images. Following the methodology

[1] https://github.com/ahujak/IB-IRM

[2] `https://github.com/1Konny/VIB-pytorch`

from [4], we define a ten-class classification task. These classes represent digits 0 through 9. In the two training domains, each digit class is associated with a color that exhibits a strong correlation with the label. By contrast, in the unseen test domain, the color remains independent of the label.

Linear unit dataset (LNU-3/3S) [17]. The Linear Unit (LNU) dataset, a synthetic dataset, was crafted based on a linear low-dimensional model. This dataset is designed to test the DG algorithms, especially when influenced by spurious invariant features [17]. Comprising six sub-datasets, each one encompasses either three or six domains, with each domain contains 10,000 samples. For our evaluation, we choose LNU-3 and LNU-3S sub-datasets, as indicated in the numerical results from [4], these are the most challenging two sub-datasets.

4.6.2 Methods for Comparison

We assess the performance of our proposed Conditional Entropy and Invariant Risk Minimization (CE-IRM) method against several benchmark algorithms, including: (i) Empirical Risk Minimization (ERM) [130], serving as a baseline, (ii) the original Invariant Risk Minimization (IRM) algorithm [12], (iii) Information Bottleneck Empirical Risk Minimization (IB-ERM) algorithm [4], and (iv) Information Bottleneck Invariant Risk Minimization (IB-IRM) algorithm [4]. A comparison with the algorithm from [81] was not conducted as its implementation was inaccessible during our study. Moreover, except for the CS-CMNIST dataset, where our method outperforms theirs by roughly 10% points, they do not report results for the other datasets that we used. In addition to the aforementioned algorithms used for comparison, we also integrate the CEM framework with the WBAE [89] method, as detailed in Chapter 3. Note WBAE method achieves DG via distribution alignment. Here we add this method for assessing the framework's adaptability and flexibility for different kinds of DG algorithms. We refer this new combination as Wasserstein Barycenter Auto-encoder

with Conditional Entropy Minimization (WBAE-CE).

4.6.3 Implementation Details

We use the training-domain validation set tuning procedure in [4] for tuning all hyper-parameters. To construct the validation set, we split the seen data into a training set and a validation set in the ratio of 95% to 5% and select the model that maximizes classification accuracy on the validation set.

For CMNIST, we utilize the learning model in [4] which is based on a simple Multi-Layer Perceptron (MLP) with two fully connected layers each having an output size 256 followed by an output layer of size two which aims to identify whether the digit is less than 5 or more than 5. The Adam optimizer is used for training with a learning rate of 10^{-4}, batch size of 64, and the number of epochs set to 500. To find the best representation, we search for the best values of weights of the Invariant Risk term and the Conditional Entropy term, *i.e.*, α, β, respectively, among the following choices: $0.1, 1, 10, 10^2, 10^3, 10^4$.

For CS-CMNIST, we follow the learning model in [4] which is composed of three convolutional layers with feature map dimensions of 256, 128, and 64. Each convolutional layer is followed by a ReLU activation and batch normalization layer. The last layer is a linear layer that aims to classify the digit to 10 classes. We use the SGD optimizer for training with a batch size of 128, learning rate of 10^{-1} with decay over every 600 steps, and the total number of steps set to 2,000. Similarly to CMNIST, we perform a search for the weights of Invariant Risk and Conditional Entropy terms with $\alpha, \beta \in \{0.1, 1, 10, 10^2, 10^3, 10^4\}$.

For the LNU dataset, we follow the procedure described in [4]. Particularly, 20 pairs of α in the range $[1 - 10^{-0.3}, 1 - 10^{-3}]$, β in the range $[1 - 10^{0}, 1 - 10^{-2}]$, learning rate in the range $[10^{-4}, 10^{-2}]$, and weight of decay in the range $[10^{-6}, 10^{-2}]$ are randomly sampled and trained. The best model is selected based on the training-

domain validation set tuning procedure. All experimental settings described above are also applied to the WBAE-CE algorithm.

We repeat the whole experiment five times by selecting five random seeds, where for each random seed, the whole process including hyper-parameters tuning and model selection is repeated. The average accuracy and standard deviation values are reported. The source code of our proposed algorithm is available at here[3].

Table 4.1: Average accuracy in percentage (%) of compared methods. The LNU-3/3S and CMNIST datasets have 2 classes, while the CS-CMNIST dataset has 10 classes. "#Doms" represents the number of domains in the dataset. The highest test accuracy is highlighted in bold, and the second highest accuracy is indicated with an underline.

Datasets	#Doms	ERM [130]	IRM [12]	IB-ERM [4]	IB-IRM [4]	WBAE-CE (ours)	CE-IRM (ours)
CS-CMNIST	3	60.3 ± 1.2	61.5 ± 1.5	71.8 ± 0.7	71.8 ± 0.7	<u>78.4</u> ± 0.3	**85.7** ± 0.9
LNU-3	6	67.0 ± 18.0	**86.0** ± 18.0	74.0 ± 20.0	81.0 ± 19.0	73.0 ± 12.0	<u>84.0</u> ± 19.0
LNU-3S	6	64.0 ± 19.0	<u>86.0</u> ± 18.0	73.0 ± 20.0	81.0 ± 19.0	70.0 ± 14.0	**90.0** ± 17.0
LNU-3	3	<u>52.0</u> ± 7.0	<u>52.0</u> ± 7.0	51.0 ± 6.0	<u>52.0</u> ± 7.0	**60.0** ± 12.0	<u>52.0</u> ± 7.0
LNU-3S	3	51.0 ± 6.0	51.0 ± 7.0	51.0 ± 6.0	51.0 ± 7.0	**60.0** ± 13.0	<u>52.0</u> ± 7.0
CMNIST	3	17.2 ± 0.6	16.5 ± 2.5	<u>17.7</u> ± 0.5	**18.4** ± 1.4	17.5 ± 0.6	17.5 ± 1.3

4.6.4 Results and Discussion

The experimental outcomes are summarized in Table 4.1. Numerical results for ERM, IRM, IB-ERM, and IB-IRM are sourced from [4].

For the CS-CMNIST dataset, the classification accuracy of the four benchmark algorithms ranges between 60% and 72%. Remarkably, our CE-IRM algorithm significantly outperforms the best alternative, enhancing performance by nearly 14% points. This performance gap stems from the construction routine of CS-CMNIST dataset where colors (spurious features) are integrated independently of the digits (invariant features) for specific labels. Therefore, our assumption $Z_{\text{sp}} \perp\!\!\!\perp Z_{\text{inv}}|Y$ holds for this dataset.

Regarding the LNU dataset, we follow the methodologies detailed in [4] to calculate the classification errors (or equivalently, accuracy) of the evaluated algorithms. The

[3] `https://github.com/thuan2412/Conditional_entropy_minimization_for_Domain_generalization`

mean accuracy and its standard deviation are reported in Table 3.10. Following [4], comparisons are made on both LNU-3 and LNU-3S datasets, considering either 6 or 3 domains (aligning with the same 3 domains mentioned in [4]).

With six domains in consideration, our CE-IRM method surpassed the other four algorithms by over 4% points on the LNU-3S dataset. However, it fells behind the IRM method by approximately 2% points on the LNU-3 dataset. In scenarios with three domains, the performance of all the algorithms is similar to each other for both LNU-3 and LNU-3S datasets. The analysis drawn from the LNU-3 and LNU-3S results underscores that increasing the number of training domains can boost the test accuracy across all methods.

Compared with the CS-CMNIST and LNU-3/3S datasets, the CMNIST stands out as the most challenging dataset, with no algorithm works well. This can be attributed to the inherent design of CMNIST which exhibits pronounced spurious correlations between the data and label, leading to the failure of all evaluated methods. These findings are consistent with the observations reported in [4], and [81].

Additionally, the combination of CEM principle and WBAE algorithm achieves the best or the second best performance on 3 out of 6 datasets, demonstrating the adaptability of our framework on different kinds of DG algorithms.

4.7 Conclusions

In this chapter, we have introduced a novel DG strategy grounded on the CEM principle, targeting the elimination of spurious features. By combining the well-known IRM algorithm with the CEM principle, the proposed algorithm achieve competitive or better performance compared to the state-of-the-art DG algorithms. Theoretically, we have demonstrated the intrinsic relationship of our objective function and the DIB method, and proved that under particular conditions, our method can extract the

true domain-invariant features. We focused on the simple model where the features learned by an IRM algorithm are considered as a linear combination of true and spurious invariant features. Our future work will focus on combining the non-linear IRM algorithm [88] with a nonlinear Blind Source Separation method, *e.g.*, non-linear ICA [64], to accommodate non-linear mixture models.

Chapter 5

Model Selection for Domain Generalization

In the previous chapters, we have discussed both theoretical work and algorithms aimed at addressing the DG problem. In this chapter, our focus shifts from specific algorithms to the entire workflow for DG. As we introduced earlier, state-of-the-art domain generalization methods commonly train a representation function followed by a classifier jointly to minimize both the classification risk and the domain discrepancy. However, when it comes to model selection, most of these methods rely on traditional validation routines that select models based solely on the lowest classification risk on the validation set [26, 67, 119]. In this chapter, we theoretically demonstrate that there exists a trade-off between minimizing classification risk and mitigating domain discrepancy, *i.e.,* it is impossible to achieve the minimum of these two objectives simultaneously. Motivated by this theoretical result, we propose a novel model selection method suggesting that the validation process should consider both the classification risk and the domain discrepancy.

5.1 Introduction

Seeking domain-invariant features is a popular method to address the DG problem. A large number of methods aim to learn the domain-invariant features by minimizing the domain discrepancy in the representation space [12, 81, 85, 89, 101, 125]. Though the domain discrepancy has been accounted for at the training step, few works considered it for model selection at the validation step [152]. Indeed, following traditional machine learning settings, most of the state-of-the-art DG methods form a validation set using a small portion of data from all seen domains and select the model that achieves the lowest classification risk or highest classification accuracy on it. However, unlike traditional machine learning settings where a model with lower classification risk on the validation set is likely to perform better on the test set, we theoretically show that for the DG problem, where the *i.i.d.* assumption does not hold, selecting the model with minimum classification risk may enlarge the domain discrepancy, subsequently leading to a non-optimal model on the unseen domain. Therefore, we argue that model selection in DG requires considering both the classification risk and the domain discrepancy to identify models that perform well on unseen domains.

5.1.1 Contributions

We summarize our contributions as follows:

1. We theoretically show that there is a trade-off between minimizing classification risk and domain discrepancy. This trade-off leads to the conclusion that targeting only a model with the lowest classification risk on the validation set can encourage a distribution mismatch between domains (enlarging domain discrepancy) and reduce the generalizability of the model.

2. Based on our theoretical result and considering the limited attention given to DG-specific validation processes, we propose a simple yet effective validation/model

selection method that integrates both the classification risk and the domain discrepancy as the validation criterion. We further demonstrate the effectiveness of this approach on various DG benchmark datasets.

5.2 Related Work

The trade-off between minimizing the classification risk and domain discrepancy has been mentioned in the literature [22, 146]. As introduced in Chapter 2, Shai-Ben David *et al.* [22] construct an upper bound on the risk of the target domain, composed of the risk from the source domain and the discrepancy between the target and source domains. The authors suggest that there must be a trade-off between minimizing the domain discrepancy and minimizing the risk of the seen domain, but do not propose any further details on how this trade-off is determined and characterized. Zhao *et al.* [146] show that the sum of risks from the source and target domains is lower bounded by the distribution discrepancy between domains. If the discrepancy between domains is large, one can not simultaneously achieve small risks on both domains. Although sharing some similarities, our theoretical result differs from [146] since Zhao *et al.* consider the trade-off between minimizing the risks of different domains rather than the trade-off between optimizing the classification risk and the domain discrepancy. On the other hand, most DG works adopt the model selection methods following traditional machine learning settings, *i.e.*, a validation set is first formed by combining small portions of data from all seen domains and the model that produces the lowest classification risk or highest classification accuracy on the validation set is then selected.

Only a limited number of studies have explored novel model selection methods in the context of DG [9, 14, 57, 113, 135, 143]. Among them, authors of [57] provide a benchmark DG package, DomainBed, with three model selection methods. These

three selection methods, namely Training-domain validation, Leave-one-domain-out validation, and Test-domain validation, have been widely adopted by previous studies and also subsequent DG work built on the DomainBed package [80, 89, 93]. Below, we provide a brief summary of these three model selection methods along with their drawbacks.

The Training-domain validation, which serves as a traditional machine learning validation method, involves splitting the training data into a training set and a validation set. Hyper-parameters and models are subsequently selected based on the validation accuracy. Training-domain validation method essentially assumes that the test data shares a similar distribution with the training data, which does not generally hold for the DG problem. As pointed out in [135], such validation may also undermine the advantages of DG algorithms.

Leave-one-domain-out validation, as indicated by its name, leaves one domain out of the training data to mimic the real DG scenario. Models with candidate hyper-parameters will be repeatedly validated on the remaining training domains, with one domain left out each time. An averaged validation accuracy is then used for the hyper-parameter selection. However, this method is unsuitable for datasets with multiple domains, such as Colored MNIST [12], and it can become computationally expensive.

Test-domain validation uses the data from the unseen test domain for model selection, thus is typically used to assess the potential of DG algorithms [57] rather than being employed as a model selection method, and thus is outside the scope of this chapter.

Recognizing the limitations of these traditional methods for DG, several works have proposed novel validation algorithms to overcome this DG-specific framework issue. For example, [135] considers model stability by measuring the average Expected Calibration Error (ECE) [40] on the training domains as a supplement to the training

domain validation accuracy. Among a set of candidate models that achieve a validation accuracy above a certain threshold, the one with the lowest average ECE is selected as the optimal model.

In a similar vein, [143] designs a validation algorithm that simultaneously achieves high validation accuracy and low feature variation. Viewing DG from the worst-case generalization perspective, [113] balance the validation set and use the worst-group validation accuracy as the selection criterion. Additionally, [14] shows that a model may produce an unstable test domain accuracy during the training process, even with a stable training domain validation performance. To address this issue, they propose a model averaging protocol to stabilize the test performance with respect to the model's validation accuracy.

The most related work of this study is [9], where the authors mentioned that they use the training loss (including both classification risk and adversarial domain discrepancy loss) on the validation set for model selection. However, it is not clear from their paper and their released code how the classification risk and the adversarial domain discrepancy loss are used to validate the model and how these two terms are balanced. On the contrary, we propose an alternative approach for combining classification risk and domain discrepancy loss in a meaningful way in light of our theoretical results.

5.3 Problem Formulation

5.3.1 Notations

Let $\mathcal{X}$, $\mathcal{Z}$, $\mathcal{Y}$ denote the input space, the representation space, and the label space, $\mathcal{D}^{(s)}$ and $\mathcal{D}^{(u)}$ represent the seen and unseen domain, respectively. $f : \mathcal{X} \to \mathcal{Z}$ and $g : \mathcal{Z} \to \mathcal{Y}$ are the representation function and the classifier. We use capital letters for the random variables in different spaces and lowercase letters for samples. Specifically,

we denote X as the input random variable, Z as the extracted feature random variable, and Y as the label random variable. The input samples, the feature samples and the labels of the input samples are denoted as $\boldsymbol{x}, \boldsymbol{z}$, and $y(\boldsymbol{x})$, respectively. Finally, we use $p^{(s)}(\cdot)$ and $p^{(u)}(\cdot)$ to denote the distributions or joint distributions corresponding to the variables inside the bracket on seen domain and unseen domain, respectively.

5.3.2 Problem Formulation

For a representation function f and a classifier g, the classification risk induced by f and g on seen domain is:

$$\begin{aligned} C^{(s)}(f,g) &= \int_{\boldsymbol{x}\in\mathcal{X}} p^{(s)}(\boldsymbol{x})\ell(g(f(\boldsymbol{x})), y^{(s)}(\boldsymbol{x}))d\boldsymbol{x} \\ &= \int_{\boldsymbol{x}\in\mathcal{X}}\int_{\boldsymbol{z}\in\mathcal{Z}} p^{(s)}(\boldsymbol{x},\boldsymbol{z})\ell(g(\boldsymbol{z}), y^{(s)}(\boldsymbol{x}))d\boldsymbol{x}d\boldsymbol{z} \end{aligned} \tag{5.1}$$

where $\ell(\cdot,\cdot)$ is a distance measure that quantifies the mismatch between the label outputted by classifier g and the true label.

For a representation function f, the distribution discrepancy between seen and unseen domains induced by f is:

$$D(f) = d(p^{(u)}(Y,Z)||p^{(s)}(Y,Z)) \tag{5.2}$$

where $d(\cdot||\cdot)$ is a divergence measure between two distributions. Indeed, to deal with the "distribution-shift", one usually looks for a mapping f such that the discrepancy between distributions of seen and unseen domains $D(f)$ is small [39, 102].

A large number of DG works focus on training a model that minimizes both the classification risk $C^{(s)}(f,g)$ and the discrepancy $D(f)$ using data from seen domains [12, 81, 85, 89, 101, 125]. Note that while $C^{(s)}(f,g)$ can be directly minimized, one usually need to approximately/heuristically optimize $D(f)$ by optimizing the

distribution discrepancy between several seen domains. Since there are already well-established theoretical and empirical works on minimizing the classification risk and domain discrepancy, our work aims to highlight the trade-off between these two objectives (Sec 5.4) and argues that taking both objectives into account during model selection can improve model's performance on unseen domains (Sec. 5.5).

5.4 Trade-off between Classification Risk and Domain Discrepancy

We first begin with a definition.

Definition 5.4.1 (Classification risk-domain discrepancy function)**.** For any representation function f and classifier g, define:

$$\begin{aligned} T(\Delta) &= \min_{f:\mathcal{X}\to\mathcal{Z}} D(f) = \min_{f:\mathcal{X}\to\mathcal{Z}} d(p^{(u)}(Y,Z)||p^{(s)}(Y,Z)) \\ \text{s.t.} \quad C^{(s)}(f,g) &= \int_{\boldsymbol{x}\in\mathcal{X}} p^{(s)}(\boldsymbol{x})\ell(g(f(\boldsymbol{x})), y^{(s)}(\boldsymbol{x}))d\boldsymbol{x} \leq \Delta \end{aligned} \tag{5.3}$$

where Δ is a positive number, $\ell(\cdot,\cdot)$ is a distance measure, and $d(\cdot||\cdot)$ is a divergence measure.

$T(\Delta)$ is the minimal discrepancy between the joint distribution of the unseen domain and seen domain if the classification risk on seen domain $C^{(s)}(f,g)$ does not exceed a positive threshold Δ. Next, we formally show that there is a trade-off between minimizing the distribution discrepancy $D(f)$ and minimizing the classification risk $C^{(s)}(f,g)$.

Theorem 5.4.2 (Main result)**.** *If the divergence measure $d(a||b)$ is convex (in both a and b), for a fixed classifier g, $T(\Delta)$ defined in (5.3) is monotonically non-increasing, and convex.*

Proof. The proof of this theorem is mainly based on the proposed approach in Rate-Distortion theory [34]. In particular, consider two positive numbers Δ_1 and Δ_2, and assume $\Delta_1 \leq \Delta_2$. For a given classifier g, we use $\mathcal{F}_{\Delta_1}$ and $\mathcal{F}_{\Delta_2}$ to denote the sets of mappings f such that $C^{(s)}(f,g) \leq \Delta_1$ and $C^{(s)}(f,g) \leq \Delta_2$, respectively. First, we show that $T(\Delta)$ is non-increasing. Indeed, since $\Delta_1 \leq \Delta_2$, $\mathcal{F}_{\Delta_1} \subset \mathcal{F}_{\Delta_2}$, we have:

$$\begin{aligned} T(\Delta_1) &= \min_{f \in \mathcal{F}_{\Delta_1}} d(p^{(u)}(Y,Z)||p^{(s)}(Y,Z)) \\ &\geq \min_{f \in \mathcal{F}_{\Delta_2}} d(p^{(u)}(Y,Z)||p^{(s)}(Y,Z)) = T(\Delta_2). \end{aligned}$$

Next, to prove the convexity of $T(\Delta)$, we need to show that:

$$\lambda T(\Delta_1) + (1-\lambda)T(\Delta_2) \geq T(\lambda\Delta_1 + (1-\lambda)\Delta_2), \forall \lambda \in [0,1]. \tag{5.4}$$

To prove (5.4), we need some additional notations. Here we define:

$$f_1 = \operatorname*{argmin}_{f:\mathcal{X}\to\mathcal{Z}} D(f) \quad \text{s.t. } C^{(s)}(f,g) \leq \Delta_1, \tag{5.5}$$

$$f_2 = \operatorname*{argmin}_{f:\mathcal{X}\to\mathcal{Z}} D(f) \quad \text{s.t. } C^{(s)}(f,g) \leq \Delta_2. \tag{5.6}$$

Note that for any f, $Y \to X \to Z$ forms a Markov chain, thus:

$$p^{(u)}(Y,Z) = p^{(u)}(Y|X)\, p^{(u)}(X,Z), \tag{5.7}$$

$$p^{(s)}(Y,Z) = p^{(s)}(Y|X)\, p^{(s)}(X,Z), \tag{5.8}$$

where $p^{(u)}(Y|X)$ and $p^{(s)}(Y|X)$ are independent of f and only depend on the conditional distributions of label and data on seen and unseen domains.

Let $p_1^{(u)}(Y,Z)$, $p_1^{(s)}(Y,Z)$ be the joint distributions of Y and Z on unseen and seen domain produced by f_1, and similarly $p_2^{(u)}(X,Z)$, $p_2^{(s)}(X,Z)$ be the joint distributions

produced by f_2. Let

$$p_\lambda^{(u)}(X,Z) = \lambda p_1^{(u)}(X,Z) + (1-\lambda)p_2^{(u)}(X,Z), \tag{5.9}$$

$$p_\lambda^{(s)}(X,Z) = \lambda p_1^{(s)}(X,Z) + (1-\lambda)p_2^{(s)}(X,Z). \tag{5.10}$$

By definition, the left hand side of (5.4) can be rewritten by:

$$\begin{aligned}
&\lambda T(\Delta_1) + (1-\lambda)T(\Delta_2) \\
=&\lambda d\big(p_1^{(u)}(Y,Z) \,||\, p_1^{(s)}(Y,Z)\big) + (1-\lambda) d\big(p_2^{(u)}(Y,Z) \,||\, p_2^{(s)}(Y,Z)\big) \\
=&\lambda d\big(p^{(u)}(Y|X)p_1^{(u)}(X,Z) \,||\, p^{(s)}(Y|X)p_1^{(s)}(X,Z)\big) && (5.11) \\
&+ (1-\lambda) d\big(p^{(u)}(Y|X)p_2^{(u)}(X,Z) \,||\, p^{(s)}(Y|X)p_2^{(s)}(X,Z)\big) && (5.12) \\
\geq& d\big(p^{(u)}(Y|X)p_\lambda^{(u)}(X,Z) \,||\, p^{(s)}(Y|X)p_\lambda^{(s)}(X,Z)\big) && (5.13)
\end{aligned}$$

where (5.11) and (5.12) are due to (5.7) and (5.8); (5.13) is due to (5.9), (5.10), and the convexity of $d(\cdot||\cdot)$.

Let f_λ be the corresponding function that induces the joint distribution $p_\lambda^{(u)}(X,Z)$ and $p_\lambda^{(s)}(X,Z)$. Define:

$$\Delta_\lambda = \int_{\boldsymbol{x}\in\mathcal{X}} \int_{\boldsymbol{z}\in\mathcal{Z}} p_\lambda^{(s)}(\boldsymbol{x},\boldsymbol{z}) \ell(g(\boldsymbol{z}), y^{(s)}(\boldsymbol{x}))\, d\boldsymbol{x} d\boldsymbol{z}. \tag{5.14}$$

By definition of $T(\Delta)$ in Definition 5.4.1, we have:

$$d\big(p^{(u)}(Y|X)\, p_\lambda^{(u)}(X,Z) \,||\, p^{(s)}(Y|X)\, p_\lambda^{(s)}(X,Z)\big) \geq T(\Delta_\lambda). \tag{5.15}$$

Combine (5.13) and (5.15):

$$\lambda T(\Delta_1) + (1-\lambda)T(\Delta_2) \geq T(\Delta_\lambda). \tag{5.16}$$

Thus, the left-hand side of (5.4) is greater or equal to $T(\Delta_\lambda)$. Next, we show that:

$$T(\Delta_\lambda) \geq T\big(\lambda\Delta_1 + (1-\lambda)\Delta_2\big). \tag{5.17}$$

Since $T(\Delta)$ is non-increasing, proving (5.17) is equivalent to prove:

$$\Delta_\lambda \leq \lambda\Delta_1 + (1-\lambda)\Delta_2. \tag{5.18}$$

Indeed, we have:

$$\begin{aligned}
\Delta_\lambda &= \int_{\boldsymbol{x}}\int_{\boldsymbol{z}} p_\lambda^{(s)}(\boldsymbol{x},\boldsymbol{z})\ell(g(\boldsymbol{z}), y^{(s)}(\boldsymbol{x}))d\boldsymbol{x}d\boldsymbol{z} && (5.19)\\
&= \lambda\int_{\boldsymbol{x}}\int_{\boldsymbol{z}} p_1^{(u)}(\boldsymbol{x},\boldsymbol{z})\ell(g(\boldsymbol{z}), y^{(s)}(\boldsymbol{x}))d\boldsymbol{x}d\boldsymbol{z} && (5.20)\\
&+ (1-\lambda)\int_{\boldsymbol{x}}\int_{\boldsymbol{z}} p_2^{(u)}(\boldsymbol{x},\boldsymbol{z})\ell(g(\boldsymbol{z}), y^{(s)}(\boldsymbol{x}))d\boldsymbol{x}d\boldsymbol{z} && (5.21)\\
&\leq \lambda\Delta_1 + (1-\lambda)\Delta_2 && (5.22)
\end{aligned}$$

where (5.19) follows from (5.14), (5.20) and (5.21) follow from (5.9), and (5.22) follows from (5.5) and (5.6), respectively. From (5.18) and (5.22), (5.17) follows. Finally, by combining (5.16) with (5.17), (5.4) is obtained. This completes the proof. □

Theorem 5.4.2 shows that only enforcing a small distribution discrepancy between domains will increase the classification risk and vice-versa.

5.5 A New Validation Method

Based on Theorem 5.4.2, we argue that to select a good model for unseen domains, one should account for both the classification risk and the domain discrepancy not only in the training process but also in the validation phase. It's worth noting that prevailing methods for model evaluation and selection in the domain generalization

context are largely centered on the classification risk or equivalently, the classification accuracy [57, 152]. Given this fact, we propose to select a model that minimizes the following objective function on the validation set:

$$L_{\text{Validation loss}} = \beta(1-\alpha)L_{\text{Classification risk}} + \alpha L_{\text{Domain-discrepancy loss}} \tag{5.23}$$

where α is the convex combination hyper-parameter and β is the scale hyper-parameter that supports the combination of objectives with different scales.

The utility of the cross-entropy loss as a good approximation of classification risk is quite evident. Yet, the challenge lies in selecting an appropriate metric to quantify the domain-discrepancy loss, which stems from the diversity of definitions associated with domain discrepancy. Several studies define domain discrepancy through the difference in marginal distributions [85, 125], while others measure it by the mismatch in conditional distributions [12]. We believe that finding a good measure for domain discrepancy is still an open problem. Therefore, within the scope of this chapter, we opt to employ the widely accepted Maximum Mean Discrepancy (MMD) loss in the feature space [85] as our choice for quantifying domain discrepancy. Nonetheless, we acknowledge that although the MMD measure is extensively employed, it may not necessarily be the optimal choice.

In practice, we observed that the MMD loss generally aligns with the cross-entropy loss in terms of scale when the training process is stable. As a result, we set β to 1. For the hyper-parameter α, we prioritize classification performance and thus, heuristically choose α as 0.2. From our experiments, we found that the performance of our validation method is robust to small values of α within the range of $[0.1, 0.3]$. Theorem 5.4.2 also provides another insight for model selection that one should avoid extreme points in Δ (classification error) for a balance between the model's generalization and prediction capabilities. In fact, this suggests that the classification error should neither be

excessively small nor overly large. Thus, for each hyper-parameter configuration, we sort the validation cross-entropy loss in ascending order and only pick the models that generate 5% to 50% percentile of the validation cross-entropy loss as a subset of candidates for model selection. Our implementation is released at this link[1].

5.6 Numerical Results

We compare the proposed model selection method with the Training-domain validation method described in [57] on three datasets: PACS [82], VLCS [48], and Colored-MNIST (C-MNIST) [12] using DomainBed package and 12 different DG algorithms provided there [57]. Recall that the Training-domain validation method chooses the model that produces the highest validation accuracy, while our method selects the model that minimizes the objective function in (5.23). For PACS and VLCS datasets, we report the average test accuracy over 4 different tasks with each time leaving one domain out as the unseen domain. For the C-MNIST dataset, we only focus on the most difficult domain, where the correlation between the label and the color of the unseen domain is completely different from the seen domains and no algorithm can achieve more than 10.5% points accuracy [57].

The validation set is formed using 20% data from each seen domain, denoted as the training-domain validation set in [57]. We follow exactly the same settings and training routine used in DomainBed and conduct 20 trials of random search over a joint distribution of hyper-parameters for each task per algorithm. For the MMD loss implementation, we directly use the code provided in DomainBed package. We train each model for 5000 steps. The validation cross-entropy loss, MMD loss, and validation accuracy are recorded every 100 steps for VLCS dataset and every 300 steps for PACS and C-MNIST datasets.

[1]https://github.com/thuan2412/A-principled-approach-for-model-validation-for-domain-generalization

Table 5.1: Classification accuracy of 12 tested algorithms on PACS, VLCS, and C-MNIST datasets using the Training-domain validation method (Traditional) proposed in [57] *vs.* using our new validation method.

Algorithm	Fish [119]	IRM [12]	GDRO [113]	Mixup [140]	CORAL [125]	MMD [85]	DANN [51]	CDANN [86]	MTL [23]	VREx [78]	RSC [62]	SagNet [99]	Wins
PACS (Traditional)	**84.6**	84.9	84.2	83.3	**85.1**	83.6	84.6	**86.4**	83.0	**84.5**	**85.2**	83.7	
PACS (Ours)	82.0	**85.3**	**84.3**	**85.3**	84.9	**85.0**	**84.9**	82.0	**84.2**	84.2	81.3	**85.1**	7/12
VLCS (Traditional)	**79.4**	76.0	78.1	77.4	76.8	**78.5**	77.8	79.2	77.3	76.4	**78.6**	**80.5**	
VLCS (Ours)	77.5	**79.2**	**79.6**	**77.6**	**78.8**	78.0	**78.5**	**80.3**	**78.2**	**78.6**	76.1	79.3	8/12
CMNIST (Traditional)	**10.0**	10.0	10.2	**10.4**	9.7	**10.4**	10.0	9.9	10.5	10.2	10.2	10.4	
CMNIST (Ours)	9.7	**10.9**	**12.6**	10.3	**11.2**	9.9	**11.1**	**10.2**	**11.5**	**15.6**	**13.8**	**10.5**	9/12

With $\alpha = 0.2, \beta = 1$, the performance of each algorithm under different validation methods on PACS, VLCS and Colored-MNIST datasets is shown in Table 5.1. We refer to the Training-domain validation method as "Traditional" and the proposed method as "Ours". For the PACS dataset, the proposed validation method can select slightly better models for seven out of twelve DG algorithms. For the remaining five DG algorithms, our method achieves comparable performance with the "Traditional" method on CORAL [125] and VREx [78]. However, for Fish [119], CDANN [86] and RSC [62], we observe a performance deterioration. The effectiveness of the proposed method can be more easily observed on VLCS dataset, where eight out of twelve DG algorithms get an improved model selected, with the improvement varying from 0.2% to 3.2%. For the C-MNIST dataset, the proposed validation method consistently selects models with better performance compared with the "Traditional" validation method. Accuracy improves for nine out of twelve tested algorithms with the most significant improvement for VREx [78] method by 5.4%.

5.7 Conclusion

By showing the trade-off between minimizing the classification risk and domain discrepancy, we highlight that the traditional model selection methods may not be suitable for DG problem. We then propose an alternative model selection approach

that considers both objectives. While our approach outperforms traditional methods on several DG algorithms and datasets, it lacks an automatic hyper-parameter tuning strategy. Given that domain discrepancies can differ across datasets, expecting the same optimal values of α and β for all datasets might not be realistic. Determining the "optimal" values could be challenging both practically and theoretically. Therefore, we leave this as an open problem for future investigation. Despite this limitation, we believe our approach offers valuable insights and initial outcomes for developing novel model selection methods tailored to the DG problem.

Chapter 6

Conclusions

In this thesis, we reviewed different paths paved by the researchers for addressing the DG problem. Specifically, we revisited the common path that is shared with the domain adaptation problem, especially in the theory part, and the new avenues built specifically for the DG problem. Our contributions to DG research, covering theory, algorithm and workflow design, are shown in Chapters 3, 4, and 5.

In Chapter 3.1, we studied the DG problem through a theoretical lens, presenting a novel upper bound for the risk of unseen domains. This proposed bound encompasses four components: empirical risk in the input space, the discrepancy between seen and unseen domain representation distributions, a reconstruction loss quantifying the quality of data recovery from its representation; and a combined risk term that is intrinsic to the domain itself. We demonstrated that our bound bridges the gap between the previous bounds and the existing practical algorithms, addressing the limitations of previous theoretical work in three aspects. Firstly, our bound addresses the optimization challenges stemming from the dependency of combined risk on the representation function. Our upper bound achieves this by making the combined risk constant relative to both representation and labeling function. Secondly, unlike bounds using the Wasserstein distance for measuring domain discrepancy, our proposed upper

bound constructs the discrepancy term in the representation space rather than the data space. This approach supports the decomposition of hypothesis when bounding the risk and designing practical algorithms. Lastly, drawing inspiration from the proposed upper bound, our WBAE algorithm shows competitive performance against other theory-guided state-of-the-art DG algorithms, underscoring the effectiveness of the proposed bound. Importantly, our bound encourages the minimization of the reconstruction loss arising from the representation function, which was proved to be important in Chapter 3.2. Serving as a complementary extension to Chapter 3.1, Chapter 3.2 demonstrates that although domain-invariant representation is crucial for DG, it does not guarantee the preservation of high mutual information between the label and the representation in unseen domains. To overcome this, we imposed a constraint on the representation function by adding a reconstruction loss to guarantee that the extracted feature preserves essential label information. We further showed the trade-off between this reconstruction loss and domain alignment in DG. Specifically, minimizing both simultaneously may not be possible. Grounded on these theoretical insights, we chose not to design a new algorithm but a versatile framework that can be seamlessly integrated to various DG algorithms. We assessed this framework using various DG algorithms and datasets, demonstrating its efficacy in boosting the robustness of DG models.

In Chapter 4, we offered an in-depth exploration of domain-invariant features, with a method to mitigate the detrimental effects of spurious features on model performance. Utilizing the Conditional Entropy Minimization (CEM) principle, we demonstrated that spurious domain-invariant feature can be filtered out if the data satisfies some assumptions. Our analysis uncovers the link between our objective function and the deterministic information bottleneck (DIB) method. Additionally, we provided a theoretical result confirming our method's capability to remove spurious features under certain conditions.

In Chapter 5, we shifted our attention from algorithm design to the whole work flow of the DG problem. In particular, by demonstrating the inherent trade-off between minimizing the classification risk and domain discrepancy, we suggested that conventional model selection methods may not be suitable for the DG problem. As an alternative, we proposed a model selection/validation method that accounts for both objectives. An extensive evaluation, spanning twelve DG algorithms on three benchmark DG datasets, attests to the ability of the proposed method to consistently select models that outperform conventional methods.

Limitations and Future Work

The DG problem is a realistic yet challenging problem, with no single algorithm capable of resolving it perfectly. In this section, we reflect on the shortcomings of our methods discussed in this thesis and suggest potential avenues for future exploration.

In Chapter 3.1, while our WBAE algorithm demonstrates efficacy in tackling the DG problem, its computational demands grow with the increase in batch size, especially when aiming for a more accurate estimation of the Wasserstein-2 barycenter. To mitigate this, future research could integrate recent innovations in large-scale barycenter and mapping estimators [47,76]. Such integration could speed up barycenter computations over larger sample sizes.

In Chapter 4, our CEM-based algorithm operates under a simplified condition, treating the learned domain-invariant feature as a linear mixture of the true and spurious invariant features. This linear assumption might not be universally applicable. Consequently, future research should focus on addressing the challenges posed by non-linear mixture cases.

In Chapter 5, though showing enhanced results, our DG-focused validation method does not provide an automated mechanism for hyper-parameter tuning. As domain discrepancies vary by dataset, assuming uniform optimal values for the trade-off terms

between empirical risk and domain discrepancy term is impractical. Determining the "optimal" values could be challenging both practically and theoretically. Looking ahead, research could focus on investigating alternative discrepancy metrics and setting thresholds based on classification accuracy rather than directly referencing empirical risk values.

Appendix A

Appendix

Wasserstein Distance: Before introducing the general version of the Wasserstein distance with respect to continuous probability distribution, let us first consider its discrete case, which usually comes together with Optimal Transport (OT) problem. To facilitate this brief introduction, we introduce some temporary notations: Consider two discrete sets of points $\{\boldsymbol{x}_i\}_{i=1}^n, \boldsymbol{x}_i \in \mathbb{R}^d$, and $\{\boldsymbol{y}_j\}_{j=1}^n, \boldsymbol{y}_j \in \mathbb{R}^d$, both $\boldsymbol{x}, \boldsymbol{y}$ are under the same metric space and we treat them as two empirical distributions,

$$\boldsymbol{a} = \sum_{i=1}^{n} a_i \delta_{\boldsymbol{x}_i}, \boldsymbol{b} = \sum_{j=1}^{m} b_j \delta_{\boldsymbol{y}_j} \tag{A.1}$$

where $\delta_{\boldsymbol{x}_i}$ and $\delta_{\boldsymbol{y}_j}$ are Dirac functions at the position of $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$, a_i and b_j are the corresponding probabilities. Without further information, a_i and b_j will be set as $\frac{1}{n}$ and $\frac{1}{m}$ respectively. $\boldsymbol{C} \in \mathbb{R}^{n \times m}$ with the i, j-th element $\boldsymbol{C}_{i,j}$ being the cost of associating the point $\boldsymbol{x}_i$ to the point $\boldsymbol{y}_j$.

Here, we borrow the "mine and factory" metaphor from [59,105]. Imagine a scenario where we own n warehouses and m factories situated at different locations. Each warehouse contains valuable mines required by the respective factories. In this setup, assume that warehouse i holds a_i units of mines, while factory j requires b_j units of mines to operate effectively, and all mines must be moved from the warehouse to the

factories. A transportation company is available to provide this service, charging a fee denoted by $\boldsymbol{C}_{i,j}$ for moving one unit of mine from warehouse i to factory j. If we need to transport a_i units of mines, the cost will be calculated as $a_i * C_{i,j}$. As the owner, we of course want the task done with the lowest cost. Thus, we decide to ask our friend, a mathematician, to design the transport plan for us. The problem is formulated as finding the optimal transportation plan such that we can spend the least money to move all mines to our factories with the demanded amount. Formally: find a plan $\boldsymbol{T} \in \mathbb{R}^{n \times m}$ that is the solution to

$$\arg \min_{\boldsymbol{T} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle \tag{A.2}$$

where $\langle \boldsymbol{C}, \boldsymbol{T} \rangle = \sum_{i,j} \boldsymbol{C}_{i,j} \boldsymbol{T}_{i,j}$, $U(\boldsymbol{a}, \boldsymbol{b}) = \{\boldsymbol{T} \in \mathbb{R}_+^{n \times m} : \sum_{j=1}^{m} \boldsymbol{T}_{i,j} = \boldsymbol{a}, \sum_{i=1}^{n} \boldsymbol{T}_{i,j} = \boldsymbol{b}\}$, This is also known as the Kantorovich's relaxation [70] for the original Monge problem [94]. To reduce the computational cost of solving the linear program (A.2), an entropic regularization term is usually added to (A.2), leading to:

$$\min_{\boldsymbol{T} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle - \lambda H(\boldsymbol{T}) \tag{A.3}$$

where $H(\boldsymbol{T}) = -\sum_{i,j} \boldsymbol{T}_{i,j} (\log \boldsymbol{T}_{i,j} - 1)$. This entropic OT problem [37] can be solved efficiently using the Sinkhorn Algorithm [121] or its variations such as the Greenkhorn algorithm [1], both of which can achieve a near-linear time complexity [11].

Closely related to the OT problem, when we quantify $\boldsymbol{C}$ by the p_{th}-power of a distance metric, where $p \geq 1$, then the Wasserstein distance between the above two discrete probability distributions is written as:

$$\mathsf{W}_p(\boldsymbol{a}, \boldsymbol{b}) = \big(\min_{\boldsymbol{T} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle \big)^{1/p} \tag{A.4}$$

For a more general case, we define the Wasserstein-p [105, 115] metric between two

Borel probability measures μ, ν as:

$$\mathsf{W}_p(\mu,\nu) = (\inf_{\pi\in\Pi(\mu,\nu)} \mathbb{E}_{(\boldsymbol{u},\boldsymbol{v})\sim\pi}[\|\boldsymbol{u}-\boldsymbol{v}\|_2^p])^{1/p}$$

where $\Pi(\mu,\nu)$ is the set of joint distributions with marginals μ and ν.

Wasserstein distance quantifies the distance between two probability distribution under a given metric space. It is favored in modern machine learning works [13,56] since it can provide meaningful gradient even if support of two distributions do not overlap. When $p = 1$, Wasserstein-1 distance has another name called "Earth Mover Distance". By Kantorovich-Rubinstein theorem [133], the dual form of the Wasserstein-1 distance can be written as:

$$\mathsf{W}_p(\mu,\nu) = \sup_{\|h(x)\|_L\leq 1} \mathbb{E}_{\boldsymbol{x}\sim\mu}[h(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim\nu}[h(\boldsymbol{x})] \tag{A.5}$$

where the h is required to be 1-Lipschitz function and $\|h\|_L = \sup \frac{|f(x)-f(y)|}{|x-y|}$. If we change the upper bound of $\|h(x)\|_L$ from 1 to K, we will then obtain $K\mathsf{W}_p(\mu,\nu)$.

The content above serves as a brief overview of the Wasserstein distance. For a comprehensive review of Optimal Transport and Wasserstein distance, we refer readers to the work by Peyré [105].

Bibliography

[1] Brahim Khalil Abid and Robert Gower. Stochastic algorithms for entropy-regularized optimal transport problems. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1505–1512, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[2] Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[3] Aakash Ahmad, Muhammad Waseem, Peng Liang, Mahdi Fahmideh, Mst Shamima Aktar, and Tommi Mikkonen. Towards human-bot collaborative software architecting with chatgpt. In *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, pages 279–285, 2023.

[4] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:3438–3450, 2021.

[5] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pages 145–155. PMLR, 2020.

[6] Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R Varshney. Empirical or invariant risk minimization? a sample complexity perspective. In *International Conference on Learning Representations*, 2021.

[7] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. Adversarial invariant feature learning with accuracy constraint for domain generalization. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part II*, pages 315–331. Springer, 2020.

[8] Isabela Albuquerque. *On Robust and Generative Neural Networks with Applications to Brain-Computer Interfaces and Object Recognition*. PhD thesis, Institut National de la Recherche Scientifique (Canada), 2021.

[9] Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*, 2019.

[10] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[11] Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems*, pages 1964–1974, 2017.

[12] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *stat*, 1050:27, 2020.

[13] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[14] Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *Advances in Neural Information Processing Systems*, 2021.

[15] Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *Advances in Neural Information Processing Systems*, 35:8265–8277, 2022.

[16] Shahab Aslani, Vittorio Murino, Michael Dayan, Roger Tam, Diego Sona, and Ghassan Hamarneh. Scanner invariant multiple sclerosis lesion segmentation from mri. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 781–785. IEEE, 2020.

[17] Benjamin Aubin, Agnieszka Słowik, Martin Arjovsky, Leon Bottou, and David Lopez-Paz. Linear unit-tests for invariance discovery. *arXiv preprint arXiv:2102.10867*, 2021.

[18] Ayca Aygun, Boyang Lyu, Thuan Nguyen, Zachary Haga, Shuchin Aeron, and Matthias Scheutz. Cognitive workload assessment via eye gaze and eeg in an interactive multi-modal driving task. In *Proceedings of the 2022 International Conference on Multimodal Interaction*, pages 337–348, 2022.

[19] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 31:998–1008, 2018.

[20] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[21] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[22] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

[23] Gilles Blanchard, Aniket Anand Deshmukh, Ürun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *The Journal of Machine Learning Research*, 22(1):46–100, 2021.

[24] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24:2178–2186, 2011.

[25] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. *Advances in neural information processing systems*, 20, 2007.

[26] Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. Exploiting domain-specific features to enhance domain generalization. *Advances in Neural Information Processing Systems*, 34, 2021.

[27] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[28] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. SWAD: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

[29] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR, 2020.

[30] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 301–318. Springer, 2020.

[31] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7181–7190, 2022.

[32] Yining Chen, Elan Rosenfeld, Mark Sellke, Tengyu Ma, and Andrej Risteski. Iterative feature matching: Toward provable domain generalization with logarithmic environments. *Advances in Neural Information Processing Systems*, 35:1725–1736, 2022.

[33] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11580–11590, 2021.

[34] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[35] Imre Csiszár and János Körner. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

[36] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12455–12464, 2020.

[37] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

[38] Marco Cuturi and Arnaud Doucet. Fast computation of Wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR, 2014.

[39] Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 129–136. JMLR Workshop and Conference Proceedings, 2010.

[40] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

[41] Zhengming Ding and Yun Fu. Deep domain generalization with structured low-rank constraint. *IEEE Transactions on Image Processing*, 27(1):304–313, 2017.

[42] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32, 2019.

[43] Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees GM Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 200–216. Springer, 2020.

[44] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3937–3946, 2021.

[45] Antonio D'Innocente and Barbara Caputo. Domain generalization with domain-specific aggregation modules. In *Pattern Recognition: 40th German Conference, GCPR 2018, Stuttgart, Germany, October 9-12, 2018, Proceedings 40*, pages 187–198. Springer, 2019.

[46] Sarah Erfani, Mahsa Baktashmotlagh, Masud Moshtaghi, Xuan Nguyen, Christopher Leckie, James Bailey, and Rao Kotagiri. Robust domain generalisation by enforcing distribution invariance. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pages 1455–1461. AAAI Press, 2016.

[47] Jiaojiao Fan, Amirhossein Taghvaei, and Yongxin Chen. Scalable computations of Wasserstein barycenter via input convex neural networks. In *International Conference on Machine Learning*, pages 1571–1581. PMLR, 2021.

[48] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

[49] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. Interpolating between optimal transport and MMD using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690, 2019.

[50] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[51] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[52] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430, 2016.

[53] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[55] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[56] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[57] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

[58] Ruocheng Guo, Pengchuan Zhang, Hao Liu, and Emre Kiciman. Out-of-distribution prediction with invariant risk minimization: The limitation and an effective fix. *arXiv preprint arXiv:2101.07732*, 2021.

[59] Frank L Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of mathematics and physics*, 20(1-4):224–230, 1941.

[60] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1498–1507, 2018.

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

[62] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 124–140. Springer, 2020.

[63] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

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

[65] Maximilian Ilse, Jakub M Tomczak, Christos Louizos, and Max Welling. Diva: Domain invariant variational autoencoders. In *Medical Imaging with Deep Learning*, pages 322–348. PMLR, 2020.

[66] P Izmailov, AG Wilson, D Podoprikhin, D Vetrov, and T Garipov. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 876–885, 2018.

[67] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Feature alignment and restoration for domain generalization and adaptation. *arXiv preprint arXiv:2006.12009*, 2020.

[68] Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 527–536. PMLR, 2019.

[69] Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pages 4069–4077. PMLR, 2021.

[70] L Kantorovich. On the transfer of masses (in russian). In *Doklady Akademii Nauk*, volume 37, page 227, 1942.

[71] Rawal Khirodkar, Donghyun Yoo, and Kris Kitani. Domain randomization for scene-specific car detection and pose estimation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1932–1940. IEEE, 2019.

[72] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *VLDB*, volume 4, pages 180–191. Toronto, Canada, 2004.

[73] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9619–9628, 2021.

[74] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[75] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[76] Alexander Korotin, Vage Egiazarian, Lingxiao Li, and Evgeny Burnaev. Wasserstein iterative networks for barycenter estimation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[77] Wouter M Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785, 2019.

[78] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

[79] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[80] Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Colorado Reed, Dongsheng Li, Kurt Keutzer, and Han Zhao. Invariant information bottleneck for domain generalization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7399–7407, Jun. 2022.

[81] Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Colorado J Reed, Jun Zhang, Dongsheng Li, Kurt Keutzer, and Han Zhao. Invariant information bottleneck for domain generalization. *arXiv preprint arXiv:2106.06333*, 2021.

[82] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[83] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[84] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1446–1455, 2019.

[85] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018.

[86] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018.

[87] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, pages 3915–3924. PMLR, 2019.

[88] Chaochao Lu, Yuhuai Wu, Jośe Miguel Hernández-Lobato, and Bernhard Schölkopf. Nonlinear invariant risk minimization: A causal approach. *arXiv preprint arXiv:2102.12353*, 2021.

[89] Boyang Lyu, Thuan Nguyen, Prakash Ishwar, Matthias Scheutz, and Shuchin Aeron. Barycentric-alignment and reconstruction loss minimization for domain generalization. *IEEE Access*, 2023.

[90] Boyang Lyu, Thuan Nguyen, Matthias Scheutz, Prakash Ishwar, and Shuchin Aeron. A principled approach to model validation in domain generalization. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[91] Boyang Lyu, Thao Pham, Giles Blaney, Zachary Haga, Angelo Sassaroli, Sergio Fantini, and Shuchin Aeron. Domain adaptation for robust workload level alignment between sessions and subjects using fNIRS. *Journal of Biomedical Optics*, 26(2):1 – 21, 2021.

[92] Bo-Qun Ma, He Li, Yun Luo, and Bao-Liang Lu. Depersonalized cross-subject vigilance estimation with adversarial domain generalization. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[93] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR, 2021.

[94] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

[95] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5715–5725, 2017.

[96] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

[97] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations*, 2021.

[98] Ganesh R Naik and Dinesh K Kumar. An overview of independent component analysis and its applications. *Informatica*, 35(1), 2011.

[99] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8690–8699, 2021.

[100] Elias Chaibub Neto. Causality-aware counterfactual confounding adjustment for feature representations learned by deep models. *arXiv preprint arXiv:2004.09466*, 2020.

[101] Thuan Nguyen, Boyang Lyu, Prakash Ishwar, Matthias Scheutz, and Shuchin Aeron. Conditional entropy minimization principle for learning domain invariant representation features. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3000–3006. IEEE, 2022.

[102] Thuan Nguyen, Boyang Lyu, Prakash Ishwar, Matthias Scheutz, and Shuchin Aeron. Joint covariate-alignment and concept-alignment: a framework for domain generalization. In *2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2022.

[103] Thuan Nguyen, Boyang Lyu, Prakash Ishwar, Matthias Scheutz, and Shuchin Aeron. Trade-off between reconstruction loss and feature alignment for domain generalization. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 794–801, 2022.

[104] Erkki Oja and Aapo Hyvarinen. Independent component analysis: A tutorial. *Helsinki University of Technology, Helsinki*, 2004.

[105] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[106] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *International Conference on Machine Learning*, pages 7728–7738. PMLR, 2020.

[107] Yury Polyanskiy and Yihong Wu. Wasserstein continuity of entropy and outer bounds for interference channels. *IEEE Transactions on Information Theory*, 62(7):3992–4002, 2016.

[108] Fengchun Qiao and Xi Peng. Uncertainty-guided model generalization to unseen domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6790–6800, 2021.

[109] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. PMLR, 2022.

[110] Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II 10*, pages 737–753. Springer, 2017.

[111] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, volume 9, 2021.

[112] Paolo Russo, Fabio M Carlucci, Tatiana Tommasi, and Barbara Caputo. From source to target and back: symmetric bi-directional adaptive gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8099–8108, 2018.

[113] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[114] Olawale Elijah Salaudeen and Oluwasanmi O Koyejo. Exploiting causal chains for domain generalization. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[115] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[116] Matthias Scheutz, Shuchin Aeron, Ayca Aygun, JP de Ruiter, Sergio Fantini, Cristianne Fernandez, Zachary Haga, Thuan Nguyen, and Boyang Lyu. Estimating systemic cognitive states from a mixture of physiological and brain signals. *Topics in Cognitive Science*, 2023.

[117] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10023–10031, 2019.

[118] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[119] Yuge Shi, Jeffrey Seely, Philip Torr, Siddharth N, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, 2022.

[120] Anthony Sicilia, Xingchen Zhao, and Seong Jae Hwang. Domain adversarial neural networks for domain generalization: When it works and how to improve. *arXiv preprint arXiv:2102.03924*, 2021.

[121] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. II. *Proceedings of the American Mathematical Society*, 45(2):195–198, 1974.

[122] Nathan Somavarapu, Chih-Yao Ma, and Zsolt Kira. Frustratingly simple domain generalization via image stylization. *arXiv preprint arXiv:2006.11207*, 2020.

[123] DJ Strouse and David J Schwab. The deterministic information bottleneck. *Neural computation*, 29(6):1611–1630, 2017.

[124] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Mar. 2016.

[125] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 443–450. Springer, 2016.

[126] Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33:19276–19289, 2020.

[127] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[128] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[129] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

[130] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[131] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[132] Sergio Verdú and Dongning Guo. A simple proof of the entropy-power inequality. *IEEE Transactions on Information Theory*, 52(5):2165–2166, 2006.

[133] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[134] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

[135] Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *Advances in neural information processing systems*, 34:2215–2227, 2021.

[136] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 402–410, 2018.

[137] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8052–8072, 2023.

[138] Ziqi Wang, Marco Loog, and Jan van Gemert. Respecting domain relations: Hypothesis invariance for domain generalization. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9756–9763. IEEE, 2021.

[139] Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International conference on machine learning*, pages 6872–6881. PMLR, 2019.

[140] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6502–6509, Apr. 2020.

[141] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14383–14392, 2021.

[142] Samir S Yadav and Shivajirao M Jadhav. Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big data*, 6(1):1–18, 2019.

[143] Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. Towards a theoretical framework of out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:23519–23531, 2021.

[144] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2100–2110, 2019.

[145] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[146] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019.

[147] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31:8559–8570, 2018.

[148] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. *Advances in Neural Information Processing Systems*, 33:14435–14447, 2020.

[149] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33:16096–16107, 2020.

[150] Fan Zhou, Zhuqing Jiang, Changjian Shui, Boyu Wang, and Brahim Chaib-draa. Domain generalization via optimal transport with metric similarity learning. *Neurocomputing*, 456:469–480, 2021.

[151] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022.

[152] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022.

[153] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13025–13032, Apr. 2020.

[154] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 561–578. Springer, 2020.

[155] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *IEEE Transactions on Image Processing*, 30:8008–8018, 2021.

[156] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021.