

Partial Feature Decorrelation for Non-I.I.D Image classification

Zhengxu Yu
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, China
yuzxfred@gmail.com

Pengfei Wang
DAMO Academy
Alibaba Group
Hangzhou, China
wpf2106@gmail.com

Junkai Xu
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, China
21960439@zju.edu.cn

Liang Xie
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, China
lilydedbb@gmail.com

Zhongming Jin
DAMO Academy
Alibaba Group
Hangzhou, China
zhongming.jinzm@alibaba-inc.com

Jianqiang Huang
DAMO Academy
Alibaba Group
Hangzhou, China
jianqiang.jqh@alibaba-inc.com

Xiaofei He
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, China
xiaofeihe@cad.zju.edu.cn

Deng Cai
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, China
dengcai@cad.zju.edu.cn

Xian-Sheng Hua
DAMO Academy
Alibaba Group
Hangzhou, China
huaxiansheng@gmail.com

Abstract

Most deep-learning-based image classification methods follow a consistency hypothesis that all samples are generated under an independent and identically distributed (I.I.D) setting. Such property can hardly be guaranteed in many real-world applications, resulting in an agnostic context distribution shift between training and testing environments. It can misguide the model overfit some statistical correlations between features in the training set. To address this problem, we present a novel Partial Feature Decorrelation Learning (PFDL) Algorithm, which jointly optimizes a feature decomposition network and the target image classification model. The feature decomposition network decomposes feature embedding into independent and correlated

parts such that the correlations between features will be highlighted. Then, the correlated features are used to help learn a stable feature representation by decorrelating the highlighted correlations while optimizing the image classification model. Extensive experiments demonstrate that our proposal can improve the backbone model's accuracy in non-i.i.d image classification datasets.

1. Introduction

Classifying unknown images based on a model trained on a training dataset is a common machine learning problem. With the rise of deep learning methods, image classification accuracy has been improved tremendously in many public datasets. Most deep-learning based image classifica-

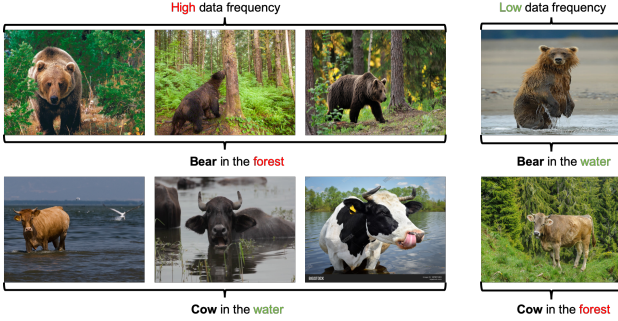


Figure 1: Illustration of context distribution shifting between training and testing datasets. Image source: NICO dataset [4].

tion methods follow a consistency assumption that all samples are generated under an independent and identically distributed (I.I.D) setting. Under this assumption, an approximation of the true distribution can be learned by yielding a lower loss on the training set. However, in real-world applications, we cannot guarantee the unknown test data will have the same distribution as the training data due to the data generation bias [4]. Consequently, in a non-I.I.D training environment, these I.I.D-based methods will inevitably overfit the statistical correlations between features, and omitted the real causality between the target object and the label [17].

A toy example of this agnostic distribution shift problem caused by data generation bias has shown in Figure 1, in which "bear in the forest" has a higher data frequency than "bear in the water". The label for the first row is "bear" and "cow" for the second row. A typical I.I.D-based deep-learning model trained in such a dataset will inevitably absorb the correlation between object "bear" and context "forest" to yield a lower training loss. Consequently, causality between the feature of the target object "bear" and the label "bear" is blurred by the correlation between the feature of context "forest" and the object "bear". We can found a similar correlation between the feature of the context "water" and the object "cow" in Figure 1. This dilemma can lead to an unstable performance in a testing environment where "bear in the water" has high data frequency. Because the misspecified model partly relies on the correlation between the context "water" and the object "cow".

Many works have been proposed recently to solve this agnostic distribution shift problem [7, 11, 15, 18], including domain adaption based methods and some causality-based sample re-weighting methods. Most domain adaption based methods [2, 11, 18] are based on a straightforward thought of taking advantage of the prior knowledge of the testing environment. However, in many real-world applications, it is impossible to obtain prior knowledge of

the test data, which hinders the application of these methods. As for causality-based sample re-weighting methods, most of them [1, 7, 14, 15] are based on increasing the importance of samples with lower data frequency to mitigate the impact of the agnostic distribution shift problem. Most of these methods require a large re-weighting matrix whose parameter number is proportional to the number of training samples, which is both computation and memory intensive.

To address these problems, we propose to learn a stable feature representation in a non-I.I.D environment by decomposing and removing the effect of correlations on the feature embedding. Specifically, we propose a novel Partial Feature Decorrelation Learning (PFDL) Algorithm for non-I.I.D image classification tasks. In this algorithm, we jointly optimize a feature decomposition network and the target image classification model while fixing another network's parameters. The start point of the feature decomposition network is to learn and decompose the correlated features from the learned feature embedding for the followed decorrelation operations. After that, we decorrelate the correlated part features while optimizing the target image classification model. By doing so, we can learn a stable feature representation by decorrelating as many correlations as possible without disturbing the stable connections between features. Compared with previous sample re-weighting-based methods, our proposal's parameter number is not proportional to the training sample number, which provides better scalability than previous works in applications with massive training data. The experimental results demonstrate that the model's accuracy and stability trained by our method can outperform all baselines and state-of-the-art stable learning methods in several non-I.I.D datasets.

We summarize the contributions of this work as follows:

(1) In this work, we study how to learn a stable feature representation under non-I.I.D problem from the perspective of decomposing and decorrelating the correlated features, so as to improve the model's accuracy. To achieve that, we proposed a novel Partial Feature Decorrelation Learning (PFDL) Algorithm for non-I.I.D image classification tasks.

(2) The first part of the PFDL Algorithm is a feature decomposition network, which aims to separate the independent features from the correlated. The second part of our proposal is a Partial Feature Decorrelation objective function. It aims to learn a stable feature representation by decorrelating correlations while optimizing the target image classification model.

(3) We conduct extensive experiments in non-I.I.D datasets to validate our proposal. The experimental results demonstrate that our proposal can achieve state-of-the-art performance on both synthetic and real-world datasets.

2. Partial Feature Decorrelation Learning

2.1. Problem Definition

In this paper, our goal is to learn a predictive model for image classification with agnostic distribution shift. As we discuss above, the agnostic distribution shift can result in correlations between features. We first introduce a formal definition of Non-I.I.D image classification problem proposed by He *et al.* [4] as follow:

Non-I.I.D Image Classification. Given training dataset $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^{n_{train}}$ and testing dataset $\mathcal{D}_{test} = \{(x_i, y_i)\}_{i=1}^{n_{test}}$, where $x_i \in \mathbb{R}^{c \times h \times w}$ represents an image sample and $y_i \in \mathbb{R}^1$ represents its label. The task is to learn a feature extractor $f : \mathbb{R}^{c \times h \times w} \rightarrow \mathbb{R}^p$ with dimension hyper-parameter p and a classifier $z : \mathbb{R}^p \rightarrow \mathbb{R}^1$, so that $z \circ f$ can predict the labels of testing data precisely when data distribution $\psi(\mathcal{D}_{train}) \neq \psi(\mathcal{D}_{test})$ and $\psi(\mathcal{D}_{test})$ is unknown.

Notations. In this work, n refers to the sample size, and p is the dimensions of variables. For any matrix $U \in \mathbb{R}^{n \times p}$, we let U_i , and $U_{-,j}$ represent the i -th row and the j -th column in U , respectively. $U_{-,j} \in \mathbb{R}^{n \times (p-1)}$ denotes the rest of U by removing its j -th column.

2.2. Feature Decomposition Network

In this subsection, we introduce our feature decomposition network to decompose feature embedding into independent part and correlated part.

Letting multivariate random variable U denotes the feature embedding extracted by f , thus U represents samples of U in a particular dataset. We can treat U as a concatenation of multivariate random variables S and V . Inspired by Wang *et al.* [20], we suppose S can be decomposed into two parts $S = S_{ind} + \hat{g}(V)$. The independent part S_{ind} is independent with V , and the correlated part $\hat{g}(V)$ is a function of V .

We propose to learn the correlation function \hat{g} with a network g . Then, we can divide $\mathbb{E}[\|S - g(V)\|_2^2]$ as follows:

$$\begin{aligned} \mathbb{E}[\|S - g(V)\|_2^2] &= \mathbb{E}[\|S_{ind} + \hat{g}(V) - g(V)\|_2^2] \\ &= \mathbb{E}[\|S_{ind}\|_2^2] + \mathbb{E}[\|\hat{g}(V) - g(V)\|_2^2] \quad (1) \\ &\quad + 2\mathbb{E}[S_{ind}] \mathbb{E}[(\hat{g}(V) - g(V))^T]. \end{aligned}$$

Since S_{ind} and V are independent, if $\mathbb{E}[S_{ind}] = 0$, then $\mathbb{E}[\|S - g(V)\|_2^2]$ reach minimum if and only if $g(V) = \hat{g}(V)$. If $\mathbb{E}[S_{ind}] \neq 0$, we only need to set $\hat{g}(V) \leftarrow (\hat{g}(V) + \mathbb{E}[S_{ind}])$ and $S_{ind} \leftarrow (S_{ind} - \mathbb{E}[S_{ind}])$ to keep the result.

In conclusion, the solution of optimization problem

$$\min_g \mathbb{E}[\|S - g(V)\|_2^2] \quad (2)$$

is \hat{g} .

So we can learn the function \hat{g} by optimizing:

$$\min_g \frac{1}{n} \sum_{i=1}^n \|s_i - g(v_i)\|_2^2, \quad (3)$$

where s_i and v_i are samples of S and V in the training set respectively.

We can use multi-layer perceptrons (MLPs) to model correlations between S and V , thanks to the universal approximation theorem [6].

In practice, given a training set $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^n$, and feature embedding $U_i = f(x_i)$, $U \in \mathbb{R}^{n \times p}$ where p denotes the dimension of the feature embedding. We successively consider each dimension $U_{i,j}$ of the feature embedding as n samples of S and the rest part $U_{-,j}$ as n samples of V . Then we have the loss function of the feature decomposition network g as follows:

$$\mathcal{L}_g = \frac{1}{n} \frac{1}{p} \sum_{j=1}^p \sum_{i=1}^n \|U_{i,j} - \tilde{U}_{i,j}\|_2^2, \quad (4)$$

where $\tilde{U}_{i,j} = g(U_{i,-j})$ and $U_{i,-j}$ denotes the sample x_i 's feature embedding except the j -th dimension.

However, if \mathcal{D}_{train} and \mathcal{D}_{test} are generated under non-I.I.D settings, the training set \mathcal{D}_{train} is biased from the true distribution, the model will inevitably overfit the correlations which only exists in the training set and omitted some causality and connections between features. It can make the residual \tilde{S}_{ind} of the misspecified model deviate from the true S_{ind} .

Hence, if we can adjust the feature extractor to reduce the distance between \tilde{S}_{ind} and S_{ind} , then we can learn a stable image classification model. Although the true S_{ind} is unknown, but we can still lower loss in Eq. 4 by iteratively optimizing g and the target image classification model. In real-world applications, there is no prior knowledge about which part of the feature embedding is the effect of correlation features. Hence, as shown in Eq. 4 and Figure. 2 "Step 1", we successively modeling each dimension $U_{i,j}$ of the feature embedding using a parameter-shared feature decomposition network g .

2.3. Partial Feature Decorrelation Algorithm

In this subsection, we introduce the PFDL algorithm, which uses Eq. 4 and g to help learn a stable feature representation.

Our goal is to decorrelate correlated features. Given a feature extractor f , we use the well-trained fixed feature decomposition network g to decompose the correlated part features when training feature extractor f and classifier z .

In general cases, if two multivariate random variable A and B is independent, from Kuang *et al.* [7], we have $\mathbb{E}[A^T B] = \mathbb{E}[A^T] \mathbb{E}[B]$. Moreover, if $\mathbb{E}[B] = 0$,

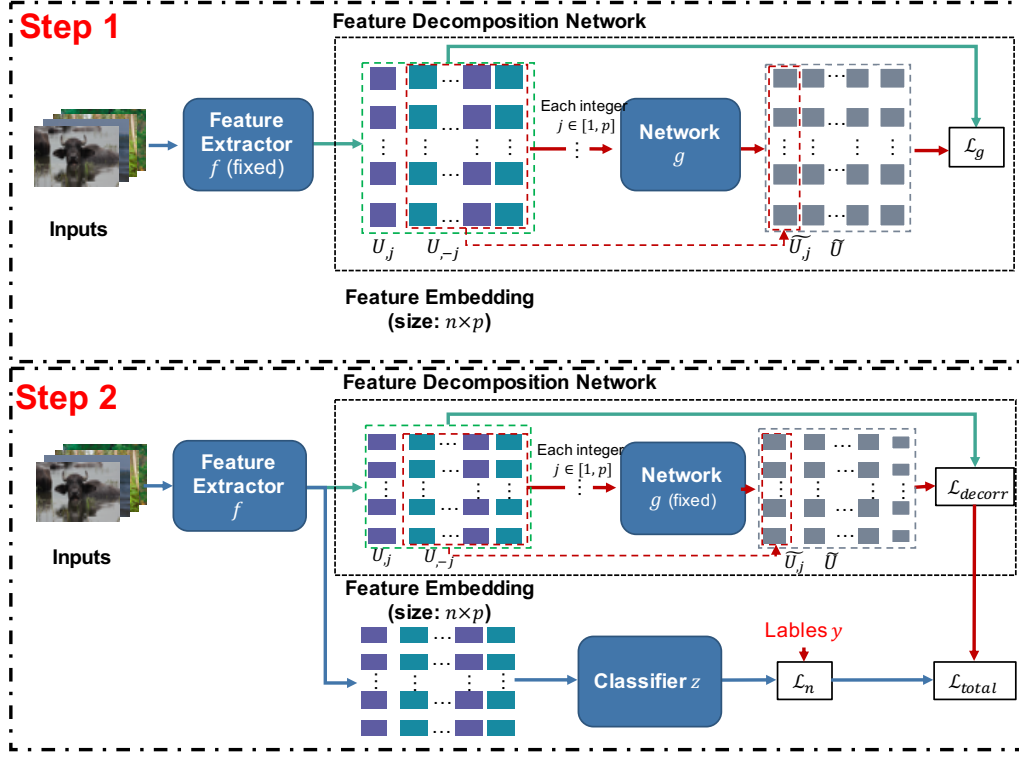


Figure 2: The framework of the proposed PFDL Algorithm. Step 1 and 2 are repeated for each mini-batch during training. n denotes sample size of each mini-batch. During inference, the feature decomposition network is dropped.

$\mathbb{E}[A^T B] = 0$. Hence, we propose the feature decorrelation constraint $\mathbb{E}[g(V)^T V] = 0$ to partially decorrelate features when training feature extractor f and classifier z . Formlly:

$$\mathcal{L}_{decorr} = \frac{1}{p} \sum_{j=1}^p \left\| \frac{1}{n} \sum_{i=1}^n \tilde{U}_{i,j}^T U_{i,-j} \right\|_2^2, \quad (5)$$

where the parameters of g is fixed, $U_i = f(\mathbf{x}_i)$ and $\tilde{U}_{i,j} = g(U_{i,-j})$.

The overall loss function for training f and z is:

$$\mathcal{L}_{total} = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_n(z(U_i)) + \frac{1}{p} \sum_{j=1}^p \left\| \frac{1}{n} \sum_{i=1}^n \tilde{U}_{i,j}^T U_{i,-j} \right\|_2^2, \quad (6)$$

where \mathcal{L}_n is the original image classification loss function, e.g. a Cross-Entropy loss function.

Based on Eq. 4 and Eq. 6, we come out with the PFDL algorithm for non-I.I.D image classification, which has shown in Algorithm 1. The framework of the proposed PFDL algorithm has shown in Figure. 2.

As shown in Figure. 2 and Algorithm 1, for each mini-batch, we first train the feature decomposition network in "Step 1" with the fixed feature extractor using Eq. 4. We then train the feature extractor and classifier in "Step 2" with the fixed feature decomposition network using Eq. 6.

Algorithm 1 Partial Feature Decorrelation Learning Algorithm

Input: Training set $D_{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, maximum epoch number E , initialized feature extractor $f^{(0)}$, classifier $z^{(0)}$ and feature decomposition network $g^{(0)}$

Output: Optimized f, z

- 1: $t = 0$
- 2: **for** e in range(0, E) **do**
- 3: **repeat**
- 4: Sample mini-batch $\{\mathbf{x}, y\}_t$ from D_{train}
- 5: Calculate \mathcal{L}_g using Eq. 4 with $(g^{(t)}, f^{(t)}, \{\mathbf{x}, y\}_t)$
- 6: Update $g^{(t+1)}$ with a stochastic gradient descent optimizer by fixing $f^{(t)}$
- 7: Calculate \mathcal{L}_{total} using Eq. 6 with $(g^{(t+1)}, f^{(t)}, z^{(t)}, \{\mathbf{x}, y\}_t)$
- 8: Update $f^{(t+1)}, z^{(t+1)}$ with a stochastic gradient descent optimizer by fixing $g^{(t+1)}$
- 9: $t = t + 1$
- 10: **until** D_{train} has been traversed
- 11: **end for**
- 12: **return** $f^{(t)}$ and $z^{(t)}$

Connections to Causal Inference Causal effect decomposition methods [19, 20] shed some lights on our proposal.

The causal effect decomposition method is intended to partition observed associations among variables into direct and indirect effects when independent relationships exist [19]. By doing so, one can analyze the direct effects of treatment features on the outcome. In fact, the direct and indirect effects are connected to the independent and correlated parts in our feature decomposition network.

Connections to Information Theory From Paninski and Liang [12], we know that two multivariate random variables X and Y with values over the space $\mathcal{X} \times \mathcal{Y}$ are independent if and only if the mutual information between X and Y is equal to 0. In fact, our proposal’s motivation can also be interpreted as learning a discriminative feature representation by reducing the mutual information between the independent part features and other features.

3. Related Works

3.1. Causality based Methods

Most recently proposed causality-based methods [1, 7, 13, 14, 15] are based on sample re-weighting, which not directly change the biased sample features, but shifts the training dataset’s distribution by varying the importance of the samples. Shen *et al.* [15] proposed a sample re-weighting method to address the collinearity among input variables caused by the agnostic distribution shift. Kuang *et al.* [7] proposed a feature decorrelation based sample re-weighting method to decorrelate all features without considering the possible causality between features. However, these sample re-weighting methods require a large re-weighting matrix whose parameter number is proportional to the training samples. Hence, these works are both computation and memory intensive. This disadvantage limits their scalability in machine learning tasks with a large number of training data. Unlike these methods, our proposal learns the associations between features by using a neural network which parameter number is not proportional to the sample number, which provides feasibility for tasks with an extensive training dataset.

3.2. Non-causality based Methods

In addition to causality based methods, a variety of domain adaptation [2, 9, 21], domain generalization [10] and transfer learning methods [11, 18] were proposed to address the non-i.i.d problem. Most of these methods handle the distribution shift between training and testing datasets by aligning the training dataset to the target dataset or vice versa. To achieve that, these methods require prior knowledge of the distribution of the target domain. However, it is impossible to acquire prior knowledge in many real-world applications. Unlike them, our proposal is intended to partition the observed associations among features to improve

the model’s performance, which requires no prior knowledge of the testing environments.

4. Experiments

We evaluated the proposed PFDL Algorithm in terms of the model performance quantitatively and qualitatively. We first introduce the datasets, evaluation metric. We verify the correlation modeling ability of the proposed Feature Decomposition Network on a synthetic non-I.I.D prediction dataset **MMADS** [7] comparing with several baseline methods and a state-of-the-art stable learning method [7]. We then demonstrate the effectiveness of PFDL on different non-I.I.D image classification datasets comparing with several baseline and state-of-the-art methods.

4.1. Datasets

MMADS. This synthetic dataset is to mimic the agnost distribution shift problem in which part of the causality is omitted due to the data generation bias, resulting in a statistical correlations between features. The task in this dataset is to predict the ground-truths using the input features. Given a input features $X = (S, V)$ and the ground-truth Y . The stable feature S have real causality with the ground-truth Y , and correlation feature V does not. Following the setting used by the state-of-the-art stable learning method DWR [7], we mimic three kinds of relationship between S and V , including independent ($S \perp V$), V dependent on S ($S \rightarrow V$) and S dependent on V ($S \leftarrow V$). The relationship among S , V and Y has form: $Y_{poly} = [S, V] \cdot [\beta_S, \beta_V]^T + S_{.,1} S_{.,2} S_{.,3} + \epsilon$, where Y_{poly} denotes the ground-truth is generated from a polynomial nonlinear function, $\beta_S = \{\frac{1}{3}, -\frac{2}{3}, 1, -\frac{1}{3}, \frac{2}{3}, -1, \dots\}$, $\beta_V = \vec{0}$ and $\epsilon = \mathcal{N}(0, 0.3)$.

To mimic the non-I.I.D settings, we generate a set of environments, each with a distinct joint distribution $P^e(X, Y)$ while preserving $P(Y|S)$ as the same. To achieve that, we generate environments by varying $P^e(V_b|S)$ on a subset of $V_b \in V$. Following the setting used by DWR [7], we vary $P^e(V_b|S)$ via biased sample selection with a bias rate $r \in [-3, -1) \cup (1, 3]$. For each sample, the probability of being selected is defined as $P_r^e = \prod_{V_i \in V_b} |r|^{-5 \cdot D_i}$, where $D_i = |f(S) - \text{sign}(r) * V_i|$. If $r > 0$, $\text{sign}(r) = 1$, otherwise $\text{sign}(r) = -1$.

NICO. NICO [4] dataset is a non-I.I.D image classification dataset, which contains 25,000 images obtained from the Internet. NICO contains two superclasses: Animal and Vehicle, with 10 classes for Animal and 9 classes for vehicle. Specifically, there are 9 or 10 contexts in each class, and each image has been manually annotated with its context. The average size of classes is about 1300 images.

We evaluate our proposal in the NICO dataset using the Animal superclass. We used the proportional bias setting

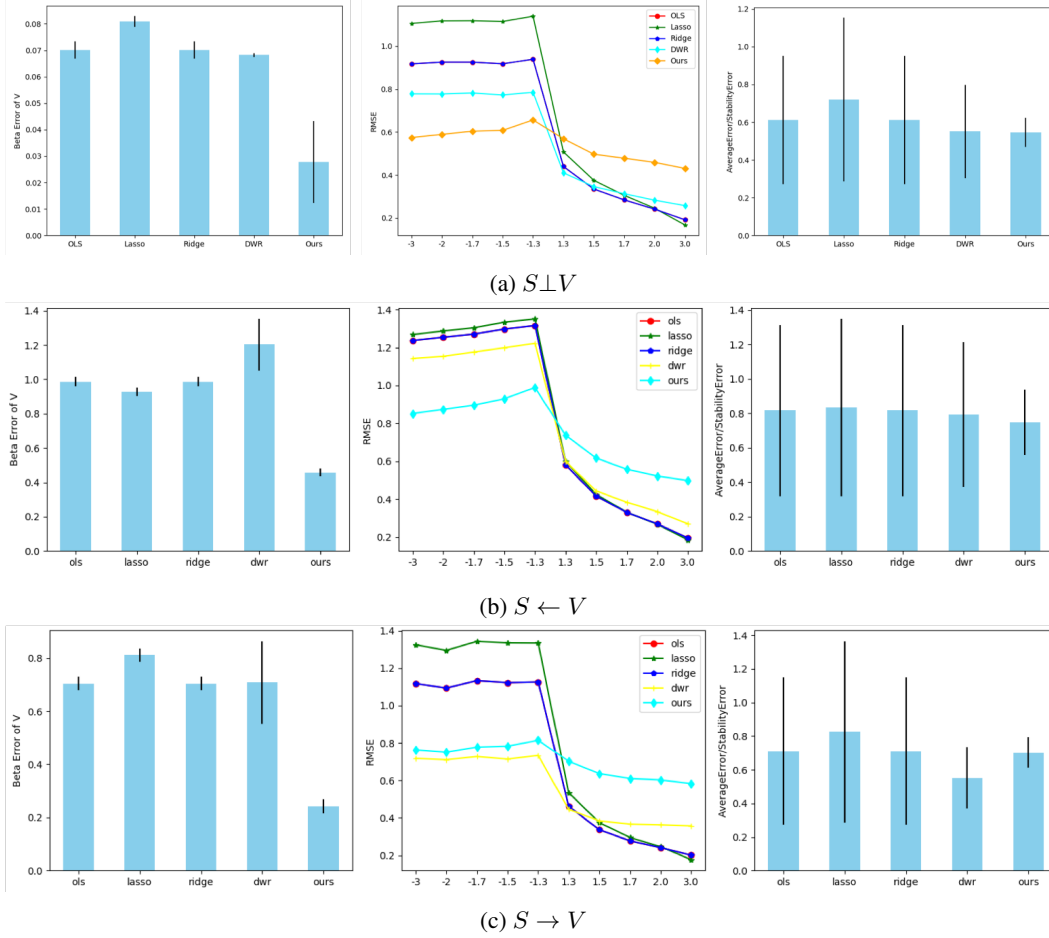


Figure 3: Experimental results with different causality settings and nonlinear function $Y = Y_{poly}$. All models are trained with $n = 2000, p = 20, r_{train} = 1.7$.

used by previous work [4], in all contexts are used in both training and testing sets, but the percentage of each context is different in training and testing sets. Particularly, one context of each class will be randomly select as the dominant context for both training and testing, and the rest considered as minor contexts. The dominant ratio [4] is defined as $Dominant\ ratio = N_{dominant}/N_{minor}$ where $N_{dominant}$ refers to the sample size of the dominant context and N_{minor} refers to the average size of other contexts where we uniformly sample other contexts.

Colored-MNIST. This dataset is a synthetic dataset formed by coloring each image, either red or green in a way that correlates strongly (but spuriously) with the class label. We followed the experiment setting of state-of-the-art work IRM [1].

4.2. Evaluation Metrics

NICO and Colored-MNIST. Following [1, 4], we use accuracy to evaluate the classification ability in both NICO and Colored-MNIST datasets.

MMADS. We use coefficient estimation error β_error [7] to evaluate correlation modeling ability of the feature decomposition network. The β_error between a learned coefficient $\hat{\beta}$ and the true coefficient β is defined as $\beta_error = \frac{1}{p} \|\beta - \hat{\beta}\|_1$, where p is feature dimension of β . We report both mean and variance of β_error of five independent experiments. Moreover, we use RMSE, Average Error (AE) [7] and Stability Error (SE) [7] to evaluate stability of our proposal.

4.3. Experiments on MMADS Dataset

We first evaluate our proposal in the synthetic dataset MMADS. Because the causality and correlation are manually defined in MMADS dataset, we can evaluate the cor-

Table 1: Experimental results under setting $S \perp V$ with $Y = Y_{poly}$ when varying sample size n , feature dimension p and training bias rate r . The smaller value in this table, the better. We use bold font to highlight the results of our proposal.

Scenario 1: varying sample size n															
n,p,r	n=1000,p=10,r=1.7					n=2000,p=10,r=1.7					n=4000,p=10,r=1.7				
Methods	OLS	Lasso	Ridge	DWR	Our	OLS	Lasso	Ridge	DWR	Our	OLS	Lasso	Ridge	DWR	Our
β_{v_error}	0.099	0.102	0.099	0.066	0.027	0.097	0.101	0.097	0.060	0.025	0.097	0.101	0.097	0.057	0.016
AE	0.604	0.639	0.603	0.519	0.629	0.583	0.617	0.583	0.509	0.613	0.587	0.621	0.587	0.505	0.569
SE	0.254	0.285	0.254	0.103	0.086	0.236	0.267	0.236	0.110	0.071	0.236	0.267	0.236	0.114	0.089
Scenario 2: varying feature dimension p															
n,p,r	n=2000,p=10,r=1.7					n=2000,p=20,r=1.7					n=2000,p=40,r=1.7				
Methods	OLS	Lasso	Ridge	DWR	Our	OLS	Lasso	Ridge	DWR	Our	OLS	Lasso	Ridge	DWR	Our
β_{v_error}	0.097	0.101	0.097	0.060	0.025	0.070	0.080	0.070	0.066	0.027	0.044	0.047	0.044	0.038	0.013
AE	0.583	0.617	0.583	0.509	0.613	0.612	0.720	0.612	0.550	0.546	0.538	0.618	0.538	0.519	0.471
SE	0.236	0.267	0.236	0.110	0.071	0.319	0.408	0.319	0.232	0.071	0.312	0.370	0.312	0.297	0.082
Scenario 3: varying bias rate r on training data															
n,p,r	n=2000,p=20,r=1.5					n=2000,p=20,r=1.7					n=2000,p=20,r=2.0				
Methods	OLS	Lasso	Ridge	DWR	Our	OLS	Lasso	Ridge	DWR	Our	OLS	Lasso	Ridge	DWR	Our
β_{v_error}	0.059	0.067	0.059	0.060	0.010	0.070	0.080	0.070	0.066	0.027	0.079	0.091	0.079	0.077	0.023
AE	0.519	0.590	0.519	0.548	0.497	0.612	0.720	0.612	0.550	0.546	0.660	0.781	0.660	0.613	0.618
SE	0.220	0.297	0.220	0.197	0.031	0.319	0.408	0.319	0.232	0.071	0.364	0.447	0.364	0.303	0.119

relation modeling ability of our proposal quantitatively. As we discussed above, the ground-truth coefficients of correlation features $\beta_V = \vec{0}$ in a pure I.I.D environment. To mimic the non-I.I.D scenario, we use different probability $P(V_b|S)$ on a subset of $V_b \in V$ for training and testing environment, to create correlations between V and the ground-truth Y while simulating the agnostic distribution shift between training and testing. By doing so, we can use the coefficient estimation error on correlation features V as a quantitative index to evaluate how our proposal models correlations. We also use Average Error (AE) and Stability Error (SE) to evaluate the stability improvement provided by our proposal.

Compared Methods We use four methods as baselines in MMADS dataset, including three I.I.D-based machine learning methods OLS, Lasso [16], Ridge Regression [5], and the state-of-the-art stable learning method DWR [7]. We used the official implementation of DWR provided by the authors.

Implementation All models are trained on the same training dataset generated using a specific bias rate r_{train} . We repeat this training process five times with different r_{train} , and report the mean and variance of β_{error} on V since the V is correlation features correlated with Y due to data generation bias. To evaluate the prediction stability, we test all models on several test environments with various bias rate $r_{test} \in [-3, -1) \cup (1, 3]$. For each test bias rate, five different test datasets are generated. To verify our proposal's correlation modeling ability, we directly using the output of the feature decomposition network as auxiliary features feed into the coefficient estimator (same as DWR)

to fit the biased training data and then evaluate the coefficient estimator error.

As for the hyper-parameters of DWR, we used the hyper-parameters provided in their official implementation. The learning rate of DWR is 0.005, and the maximum iteration is 5000. As for the hyper-parameters of PFDL, the learning rate of PFDL is 0.001, and the maximum iteration is the same as DWR.

Results The experimental results have shown in Figure 3 and Table 1, we visualized the results of three different correlation settings $S \perp V, S \leftarrow V, S \rightarrow V$. We can notice that our algorithm can achieve the lowest coefficient estimation error on β_v compare with all baselines. It shows that our proposal can model the correlations between features accurately in all three different correlations settings. Similar evidence can be found in Table 1 where our method achieved the lowest β_{v_error} and SE in all experiments comparing with all baselines including state-of-the-art method DWR. Besides, we can notice that our model can significantly mitigate the model misspecification caused by correlations and improve the model's stability across different test datasets using the output of the feature decomposition network.

We can notice that our proposal's performance is stable with the sample size changing, but the state-of-the-art method DWR is affected by the sample size, as shown in Table 1. It shows that our method is more robust when training with a small number of samples.

These observations lead to the conclusion that our proposal can achieve better stability across different test datasets than state-of-the-art methods.

Table 2: Results in the NICO dataset. The evaluation metric is standard classification accuracy in percentage. Models are trained in the same training set with *Dominant Ratio* = 5 : 1. The symbol ”-” denotes the previous work has not implemented this setting, we added these settings to show a fine-grained evaluation.

<i>Dominant Ratio</i>	1 : 5	1 : 3	1 : 1	2 : 1	3 : 1	4 : 1	5 : 1
CN [4]	37.17	-	37.80	41.46	42.50	43.23	-
CN+BN [4]	38.70	-	39.60	41.64	42.00	43.85	-
CNBB [4]	39.06	-	39.60	42.12	43.33	44.15	-
ResNet-50 [3]	39.79	39.98	40.54	42.41	43.21	46.67	44.73
ResNet-50+Ours	43.19	43.40	44.21	44.29	46.21	48.72	49.20

Algorithm	Acc. train envs	Acc. test envs
ERM [1]	87.40 \pm 0.2	17.1 \pm 0.6
IRM [1]	70.8 \pm 0.9	66.9 \pm 2.5
IRM+Ours	70.6\pm0.8	67.5 \pm1.8

Table 3: Results in Colored-MNIST.

4.4. Experiments on NICO Dataset

We then demonstrate the effectiveness of our proposal in the non-I.I.D image classification dataset NICO. As we discussed above, we use the proportional bias setting to show the stability improvement bring by our proposal compared with several baseline models.

Compared Methods. We compared our proposal with several baseline methods, including CN, CN+BN, CNBB [4] and ResNet-50 [3]. The results of CN, CN+BN and CNBB models are reported by He *et al.* [4], the backbone model CN is a typical ConvNet-based neural network. The CN+BN denotes a typical CNN-based model with batch normalization. And the CNBB is proposed by He *et al.* [4].

Implementation We first randomly sampled 60% images of each context of each class to form training set, and the rest as testing set. In each class, we randomly sampled a dominant context, and the rest become minor contexts. For example, in a dataset with *Dominant Ratio* = 1 : 5, we will sample five minor context images of each minor context if we sampled one dominant context image, until either dominant or minor context image running out.

In Table. 2, all models are trained in the same training set with *Dominant Ratio* = 5 : 1, and testing in seven different testing set with different *Dominant Ratio*.

As for hyper-parameters, the init learning rate is set to 0.01 for all models, and multiple 0.1 for every 80 epochs. All models are trained for 160 epochs.

4.4.1 Results

Table. 2 lists the non-I.I.D image classification performance in NICO dataset. We can observe that PFDL deployed on a ResNet-50 backbone achieves the best accuracy in all seven settings. We can notice that the accuracy of each model varies proportionally to the non-I.I.D settings. Both base-lines and our proposal achieved the best performance in *Dominant Ratio* = 5 : 1, because all models are trained in a training set with *Dominant Ratio* = 5 : 1. Besides, we can notice that our proposal can significantly improve the classification accuracy in the hardest setting 1 : 5, it demonstrates the effectiveness of our proposal.

4.5. Experiments on Colored-MNIST.

We then evaluate our proposal in the Colored-MNIST. The Colored-MNIST is a synthetic image classification dataset formed by coloring each image of MNIST [8] dataset, either red or green in a way that correlates strongly with the class label.

Compared Methods. We compared our proposal with a MLP-based model trained by using the standard Empirical Risk Minimization (ERM) scheme. We also compared with the state-of-the-art stable learning method IRM [1].

Implementation. We used the same implementation settings as in IRM [1]. The IRM model is also a MLP-based neural network. We generate two training sets and one testing set with different correlation ratios (same as IRM [1]). The init learning rate is 0.001 for all methods.

Results. Table 3 list the image classification accuracy of all models. As we can notice, our proposal deployed in the IRM learning scheme can further improve accuracy in test environments. It also demonstrates the effectiveness of our proposal.

5. Conclusion

In this work, we study the non-I.I.D image classification problem. We presented a novel Partial Feature Decorrela-

tion learning (PFDL) algorithm to improve the backbone model stability by partially decorrelating features during training. Experiments on synthetic and real-world datasets demonstrate that our proposal can help enhance the backbone models' stability and outperform the state-of-the-art stable learning methods. We first verified the correlation modeling ability of our proposal in a synthetic dataset. We then demonstrate the effectiveness of our proposal in two non-I.I.D image classification datasets. To upgrade PFDL, we will seek other decorrelation strategies in future works.

References

- [1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *stat*, 1050:27, 2020. [2](#), [5](#), [6](#), [8](#)
- [2] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. *arXiv preprint arXiv:2001.01046*, 2020. [2](#), [5](#)
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. [8](#)
- [4] Yue He, Zheyang Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, page 107383, 2020. [2](#), [3](#), [5](#), [6](#), [8](#)
- [5] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. [7](#)
- [6] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. [3](#)
- [7] Kun Kuang, Ruoxuan Xiong, Peng Cui, Susan Athey, and Bo Li. Stable prediction with model misspecification and agnostic distribution shift. In *AAAI*, pages 4485–4492, 2020. [2](#), [3](#), [5](#), [6](#), [7](#)
- [8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. [8](#)
- [9] Anqi Liu and Brian Ziebart. Robust classification under sample selection bias. In *Advances in neural information processing systems*, pages 37–45, 2014. [5](#)
- [10] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, 2013. [5](#)
- [11] Boyuan Pan, Yazheng Yang, Hao Li, Zhou Zhao, Yueting Zhuang, Deng Cai, and Xiaofei He. Macnet: Transferring knowledge from machine comprehension to sequence-to-sequence models. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6092–6102. Curran Associates, Inc., 2018. [2](#), [5](#)
- [12] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003. [5](#)
- [13] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016. [5](#)
- [14] Zheyang Shen, Peng Cui, Kun Kuang, Bo Li, and Peixuan Chen. Causally regularized learning with agnostic data selection bias. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 411–419, 2018. [2](#), [5](#)
- [15] Zheyang Shen, Peng Cui, Tong Zhang, and Kun Kuang. Stable learning via sample reweighting. *arXiv preprint arXiv:1911.12580*, 2019. [2](#), [5](#)
- [16] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. [7](#)
- [17] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. [2](#)
- [18] Bo Wang, Minghui Qiu, Xisen Wang, Yaliang Li, Yu Gong, Xiaoyi Zeng, Jun Huang, Bo Zheng, Deng Cai, and Jingren Zhou. A minimax game for instance based selective transfer learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '19*, page 34–43, New York, NY, USA, 2019. Association for Computing Machinery. [2](#), [5](#)
- [19] Peizhong Peter Wang. Casual effect decomposition and its implications in epidemiological studies. *JP JOURNAL OF BIOSTATISTICS*, 2(3):169–184, 2008. [4](#), [5](#)
- [20] P Peter Wang, Elizabeth M Badley, and Monique Gignac. Exploring the role of contextual factors in disability models. *Disability and rehabilitation*, 28(2):135–140, 2006. [3](#), [4](#)
- [21] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114, 2004. [5](#)