

---

# The Appendix for “Generalizing to Unseen Domains for Regression”

---

Anonymous Author(s)

Affiliation

Address

email

## 1 A Introduction of Baselines

2 The simple introductions of baselines are described as follows:

3 **ERM** [1]. The Empirical Risk Minimization method is the most simple baseline that minimizes the  
4 regression loss on source domains and reports regression loss on unseen target domains.

5 **IRM** [2]. Invariant Risk Minimization estimates invariant correlations across multiple training  
6 domains. For implementation, it can apply the gradient correlations from two batches as a penalty.

7 **MMD** [3]. The core of MMD is to align the distribution among different domains by the Maxi-  
8 mum Mean Discrepancy measure. [3] incorporate MMD into an adversarial auto-encoder to learn  
9 generalized feature representations.

10 **MTL** [4]. Marginal Transfer Learning views DG as a kind of supervised learning problem by  
11 augmenting the original feature space with the marginal distribution of feature vectors.

12 **MLDG** [5]. Meta-Learning for Domain Generalization (MLDG) is a pioneering work that applies  
13 MAML to domain generalization. MLDG optimizes meta-train and meta-test simultaneously in the  
14 outer loop. Original MAML only optimizes meta-test objective in the outer loop. The reason to  
15 optimize the meta-train objective is that we want the learned model to be capable of directly predicting  
16 the target domain. Note that there are other meta-learning methods for DG, such as MetaNorm  
17 [6] and MASF [7]. But this baseline did not release codes, e.g., MetaNorm, or are specialized for  
18 classification tasks, e.g., MASF.

19 **DANN** [8]. Domain-Adversarial Neural Networks is originally proposed to address domain adaptation  
20 problems. Besides the introduced domain adversarial framework that aligns the domain distribution,  
21 DANN also proposes an elegant implementation with a gradient reversal layer.

22 **SD** [9]. Spectral Decoupling controls the learning dynamic of models and tries to reduce the learning  
23 speed for unrelated features for out-of-distribution generalization. In the training process, the model  
24 has two options to reduce the loss toward an example, i.e., to get more confident in a learned feature  
25 or to learn a new feature. SD tends to increase feature diversity by encouraging learning new features.

26 **RSD** [10]. Representation Subspace Distance (RSD) tries to deal with general cross-domain regres-  
27 sion via subspace alignment, which reduces domain gap by minimizing RSD via the principal angles  
28 of representation matrices.

29 **SelfReg** [11]. SelfReg proposes a domain perturbation layer to make data augmentation methods like  
30 Mixup [12] more useful in self-supervised contrastive regularization.

31 **Transfer** [13]. The method successfully finds more transferable features via representation learning  
32 using adversarial training.

33 **MODE** [14]. MODE is a distribution robust optimization method that performs moderate distribu-  
34 tional exploration via style transfer. For fast implementation, we adopt its Fourier mixing version.

35 **DDG** [15]. Disentanglement-constrained Domain Generalization (DDG) tries to disentangle the  
 36 domain-agnostic semantic features and the domain-specific variation features to achieve out of  
 37 distribution prediction. The data generation and augmentation technics are also utilized to disentangle  
 38 the semantic and variation features.

39 **CAD** [16]. CAD also uses self-supervised learning like SelfReg but learns discriminative representa-  
 40 tions and aligns representation’s marginal support among different domains.

41 **CORAL** [17]. CORAL aligns the second-order statistics of the source and target distributions with a  
 42 linear transformation. In our experiments, we use the deep learning version that the distributions are  
 43 derived from the learned latent features.

44 **CausIRL** [18]. CausIRL tries to capture the invariant representations by minimizing the distance of  
 45 intervened distributions. We use MMD as the distance function in our experiments.

## 46 B Hyper-parameter Setting

47 To help the readers reproduce the reported results, we provide more hyper-parameters in Tab. 1. The  
 48 outer loop learning rate, the inner loop learning rate and the inner loop iteration steps are used by  
 49 our MAMR model, and the left hyper-parameters are shared by all methods. Note that the settings  
 50 are only suitable for age estimation datasets. To find the proper hyper-parameters for each algorithm  
 51 under limited computation resources, 5 times random hyper-parameter searches are conducted. Then  
 52 we repeat 3 times with different seeds on each group of hyper-parameters.

53 Following the data configuration in the DomainBed<sup>1</sup> benchmark, we randomly split each domain into  
 54 90% and 10% subsets. The former is used in model training and the latter for model selection. We  
 55 use two popular model selection methods in DG, i.e., test-domain validation and training domain  
 56 validation. The former is also named the oracle method that the model is selected based on the 10%  
 57 data of the test domain. The latter uses the 10% data of the training domain to select the best model.

Table 1: The hyper-parameter settings of our MAMR model and baselines.

Hyper-Parameters Setting	Values
Inner loop learning rate $\beta$	0.05
Outer loop learning rate $\alpha$	$0.1 * \beta$
Inner loop iteration steps	1
Batch size of each support or query task	64
Holdout fraction for each domain	0.1
Trial seeds:	3057, 3058, 3059
Optimizer:	SGD
Optimizer weight decay:	$5e - 4$
Data augmentation	RandomResizedCrop, RandomHorizontalFlip
Data normalization (mean)	mean=[0.485, 0.456, 0.406]
Data normalization (std)	std=[0.229, 0.224, 0.225]

<sup>1</sup><https://github.com/facebookresearch/DomainBed>

## 58 C Causal Mechanism in Toy Experiments

59 We provide the used causal mechanism in toy experiments. Fig. 1 demonstrate the mechanism to  
 60 generate  $Y$  given two inputs  $X_1$  and  $X_2$ . In this example, the dominant variable  $X_1$  controls 5  
 61 generation factors, the auxiliary variable  $X_2$  controls 3 generation factors. All the generation factors  
 62 form a sum and the sum is normalized to the interval  $[0, 1]$  for  $Y$ .

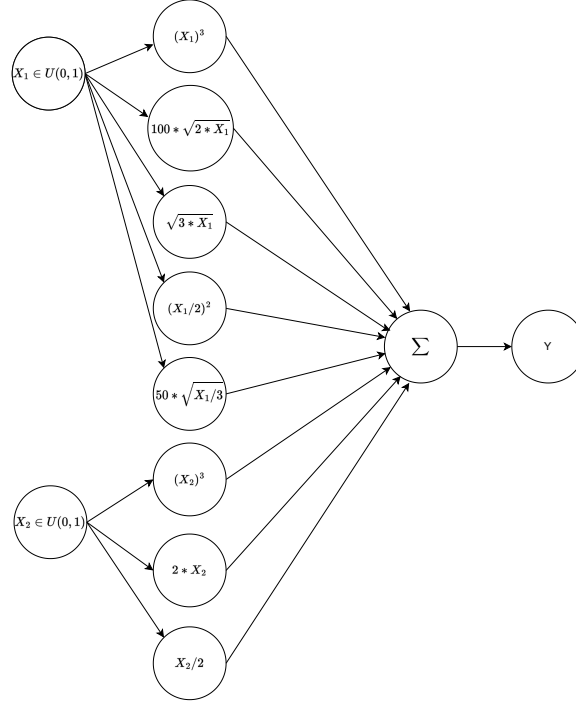


Figure 1: The example of the generation mechanism for toy experiments. Note that  $\sum$  denotes the sum of all the coming elements, and the responding value  $Y$  is normalized to  $[0, 1]$  after  $\sum$ .

## 63 D More Results on Rental Dataset

64 We provide the regression performances of our MAMR and baselines on Rental dataset <sup>2</sup>. This  
 65 dataset is released by an online competition in 2019 to predict housing rental in Shang Hai, China.  
 66 The data categories include rental housing, regions, second-hand housing, supporting facilities, new  
 67 houses, land, population, customers, real rent, etc. We split 15 regions into 4 groups as 4 different  
 68 domains (i.e., Region1, Region2, Region3 and Region4). Every domain have different rentals due  
 69 to their populations and economic conditions. The rental prices vary from 100RMB/month to  
 70 450000RMB/month. We normalize all the attributes (including target values) to  $[0, 1]$  and calculate  
 71 the MSE loss at test stage. Fig. 2 provides the statistics of the four domains. Different from age  
 72 estimation dataset, the responding values in Rental are closer to a continuous distribution. Hence  
 73 some methods like DDG is not suitable for this dataset (on age estimation dataset, each age can be  
 74 seen as a class for DDG). If you need the origin datasets, please contact us by e-mail.

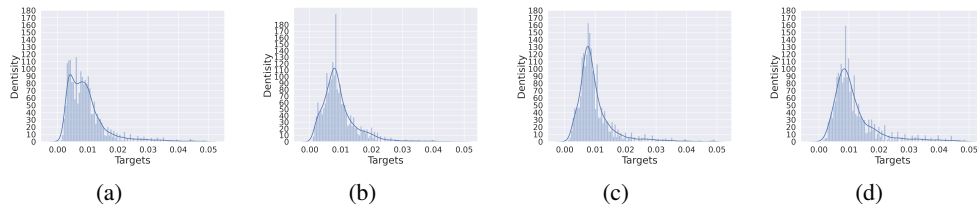


Figure 2: The histograms of four domains with kernel function density estimation. The responding values meet long-tail distribution, so we only visualize the responding values whose normalized values are less than 0.05.

75 Different from the settings on age estimation dataset, we use a 5-layer MLPs as the encoder for all  
 76 methods. Moreover, we use Adam optimizer and MAE training loss on all methods. The dataset and  
 77 codes can be found in our supplementary materials.

78 The regression results can be seen from Tab. 2. Our method also gets strong performance in average  
 79 evaluation. Besides that, we find CausIRL also shows strong performance via its causal mechanism.  
 80 However, CausIRL is normal on age estimation datasets. The above comparisons show the good  
 81 scalability of MAMR on cross-domain regression tasks.

## 82 E More Details on Age Estimation Datasets

83 Perfect age estimation is based on the assumption that all age data are available, while many real-  
 84 world datasets are not perfect and have partial ages due to privacy concerns. Hence age estimation  
 85 has been introduced in cross-domain works [19, 20].

86 CACD<sup>3</sup>. Cross-Age Celebrity Dataset (CACD) contains 163,446 images from 2,000 celebrities  
 87 collected from the Internet. The age of celebrities ranges from 16-62 and can be classified into 5  
 88 disjoint age intervals (domains), i.e.,  $[15 - 20)$ ,  $[20 - 30)$ ,  $[30 - 40)$ ,  $[40 - 50)$ ,  $[50 - 60]$ . The images  
 89 of each celebrity are sampled by different devices across multiple years. Therefore each domain  
 90 has different facial characteristics. To consider the overlapped intervals, we further create CACD-O  
 91 dataset, where each interval has 3 ages of neighbors, e.g.,  $[15 - 20)$  includes 8 different ages from  
 92 15 to 22 and  $[20 - 30)$  has 15 ages from 18 to 32. Tab. 3 and Tab. 4 provide the performances on  
 93 datasets CACD and CACD-O.

94 AFAD<sup>4</sup>. The Asian Face Age Dataset (AFAD) originally is an age estimation dataset containing  
 95 more than 160K face images and aging labels. We split the dataset into 5 age intervals (domains),  
 96 i.e.,  $[15 - 20)$ ,  $[20 - 25)$ ,  $[25 - 30)$ ,  $[30 - 35)$ ,  $[35 - 40]$ . Like CACD, each age interval has its  
 97 own face characteristics and can be viewed as 5 related domains for regression. Tab. 5 provides the  
 98 performances on this dataset.

<sup>2</sup>[https://ai.futurelab.tv/contest\\_detail/3#contest\\_des](https://ai.futurelab.tv/contest_detail/3#contest_des)

<sup>3</sup><http://bcsiriuschen.github.io/CARC/>

<sup>4</sup><https://afad-dataset.github.io/>

Table 2: Regression results on Rental dataset with training-domain validation. Each region column denotes the target domain with the others as source domains. Many original MSE results are less than  $1e - 3$ . For elegant demonstration, we have multiplied  $1e + 3$  for each mean result as well as its standard variance. Note that in the main paper, we multiplied  $1e + 2$  for each mean result to keep the same magnitude as other datasets.

Algorithm	Region1	Region2	Region3	Region4	Avg
ERM	0.53358 $\pm$ 0.0603	0.58015 $\pm$ 0.0002	0.37834 $\pm$ 0.0000	0.41955 $\pm$ 0.0000	0.47791
IRM	0.60709 $\pm$ 0.0090	0.58003 $\pm$ 0.0001	0.37835 $\pm$ 0.0000	0.41952 $\pm$ 0.0000	0.49625
MLDG	0.46439 $\pm$ 0.0046	0.57993 $\pm$ 0.0001	0.37835 $\pm$ 0.0000	0.42002 $\pm$ 0.0002	0.46067
CORAL	1.00023 $\pm$ 0.4004	0.63521 $\pm$ 0.0451	0.37834 $\pm$ 0.0000	0.41955 $\pm$ 0.0001	0.60833
MMD	0.46926 $\pm$ 0.0010	0.58035 $\pm$ 0.0002	0.37834 $\pm$ 0.0000	0.41952 $\pm$ 0.0000	0.46187
DANN	0.52026 $\pm$ 0.0312	0.58011 $\pm$ 0.0001	<b>0.37833</b> $\pm$ 0.0000	0.41965 $\pm$ 0.0000	0.47459
MTL	0.49153 $\pm$ 0.0266	0.58023 $\pm$ 0.0003	0.37834 $\pm$ 0.0000	0.41971 $\pm$ 0.0001	0.46745
SD	0.59537 $\pm$ 0.0151	0.58003 $\pm$ 0.0001	0.37835 $\pm$ 0.0000	0.41954 $\pm$ 0.0000	0.49332
SelfReg	0.71201 $\pm$ 0.2118	0.58015 $\pm$ 0.0001	0.37836 $\pm$ 0.0000	0.41969 $\pm$ 0.0000	0.52255
CAD	0.82516 $\pm$ 0.3045	0.58014 $\pm$ 0.0001	<b>0.37833</b> $\pm$ 0.0000	0.41961 $\pm$ 0.0002	0.55081
Transfer	0.50443 $\pm$ 0.0283	0.58002 $\pm$ 0.0001	0.37834 $\pm$ 0.0000	<b>0.41946</b> $\pm$ 0.0000	0.47056
RSD	0.53013 $\pm$ 0.0238	0.58022 $\pm$ 0.0002	0.37834 $\pm$ 0.0000	<b>0.41946</b> $\pm$ 0.0000	0.47704
CausIRL	0.45862 $\pm$ 0.0026	0.58044 $\pm$ 0.0001	0.37834 $\pm$ 0.0000	0.41951 $\pm$ 0.0000	0.45923
MODE	0.48086 $\pm$ 0.0000	0.58001 $\pm$ 0.0001	0.37835 $\pm$ 0.0000	0.41985 $\pm$ 0.0000	0.46477
MAMR	<b>0.45689</b> $\pm$ 0.0037	<b>0.58002</b> $\pm$ 0.0001	0.37834 $\pm$ 0.0000	0.42012 $\pm$ 0.0001	<b>0.45884</b>

Table 3: Regression results on CACD dataset with training-domain validation. Each regression interval (domain) in all tables denotes the target interval with the others as source intervals. The minimum Mean Squared Errors are bolded. Note that we set the standard variances to 0 if they are less than 0.01.

Algorithm	[15-20)	[20-30)	[30-40)	[40-50)	[50-60]	Avg
ERM	0.0434 $\pm$ 0.00	0.0159 $\pm$ 0.00	0.0024 $\pm$ 0.00	0.0127 $\pm$ 0.00	0.0547 $\pm$ 0.00	0.0258
IRM	0.0903 $\pm$ 0.04	0.0119 $\pm$ 0.00	0.0016 $\pm$ 0.00	0.0174 $\pm$ 0.00	0.0626 $\pm$ 0.00	0.0368
MLDG	0.0454 $\pm$ 0.00	0.0140 $\pm$ 0.00	0.0028 $\pm$ 0.00	0.0137 $\pm$ 0.00	0.0540 $\pm$ 0.00	0.0260
MMD	0.0486 $\pm$ 0.00	0.0178 $\pm$ 0.00	<b>0.0010</b> $\pm$ 0.00	0.0152 $\pm$ 0.00	0.0603 $\pm$ 0.00	0.0286
CORAL	0.0446 $\pm$ 0.00	0.0135 $\pm$ 0.00	0.0030 $\pm$ 0.00	0.0130 $\pm$ 0.00	0.0535 $\pm$ 0.00	0.0255
DANN	0.0474 $\pm$ 0.00	0.0151 $\pm$ 0.00	0.0013 $\pm$ 0.00	0.0142 $\pm$ 0.00	0.0566 $\pm$ 0.00	0.0269
SD	0.0382 $\pm$ 0.00	<b>0.0109</b> $\pm$ 0.00	0.0026 $\pm$ 0.00	0.0131 $\pm$ 0.00	0.0593 $\pm$ 0.00	0.0248
MTL	<b>0.0330</b> $\pm$ 0.00	0.0641 $\pm$ 0.00	0.1199 $\pm$ 0.00	0.2022 $\pm$ 0.00	0.3040 $\pm$ 0.00	0.1447
SelfReg	0.0433 $\pm$ 0.00	0.0133 $\pm$ 0.00	0.0023 $\pm$ 0.00	0.0130 $\pm$ 0.00	0.0542 $\pm$ 0.00	0.0252
Transfer	<b>0.0330</b> $\pm$ 0.00	0.0641 $\pm$ 0.00	0.1199 $\pm$ 0.00	0.2022 $\pm$ 0.00	0.3040 $\pm$ 0.00	0.1446
RSD	0.0464 $\pm$ 0.00	0.0190 $\pm$ 0.00	0.0045 $\pm$ 0.00	0.0217 $\pm$ 0.00	0.0650 $\pm$ 0.01	0.0313
CAD	<b>0.0330</b> $\pm$ 0.00	0.0641 $\pm$ 0.00	0.1199 $\pm$ 0.00	0.2022 $\pm$ 0.00	0.3040 $\pm$ 0.00	0.1447
CausIRL	0.0464 $\pm$ 0.00	0.0167 $\pm$ 0.00	0.0012 $\pm$ 0.00	0.0147 $\pm$ 0.00	0.0604 $\pm$ 0.00	0.0278
DDG	0.0490 $\pm$ 0.00	0.0176 $\pm$ 0.00	0.0016 $\pm$ 0.00	0.0153 $\pm$ 0.00	0.0598 $\pm$ 0.00	0.0287
MODE	0.0481 $\pm$ 0.00	0.0176 $\pm$ 0.00	0.0010 $\pm$ 0.00	0.0146 $\pm$ 0.00	0.0602 $\pm$ 0.00	0.0283
MAMR	0.0331 $\pm$ 0.01	0.0143 $\pm$ 0.00	0.0021 $\pm$ 0.00	<b>0.0078</b> $\pm$ 0.00	<b>0.0371</b> $\pm$ 0.01	<b>0.0189</b>

99 For age estimation datasets, we normalize the labels from 0 to 1 and leave out one domain at the  
100 training stage then make predictions on this domain at the test stage. To ensure a similar capacity  
101 among different age intervals, we make compensation for the small capacity interval by slightly  
102 relaxing the interval.

Table 4: Regression results on CACD-O dataset with training-domain validation.

Algorithm	[15-20)	[20-25)	[25-30)	[30-35)	[35-40]	Avg
ERM	0.0370 $\pm$ 0.00	0.0146 $\pm$ 0.00	0.0032 $\pm$ 0.00	0.0126 $\pm$ 0.00	0.0506 $\pm$ 0.00	0.0236
IRM	0.0382 $\pm$ 0.00	0.0140 $\pm$ 0.00	0.0029 $\pm$ 0.00	0.0166 $\pm$ 0.00	0.0562 $\pm$ 0.00	0.0256
MLDG	0.0371 $\pm$ 0.00	0.0141 $\pm$ 0.00	0.0035 $\pm$ 0.00	0.0130 $\pm$ 0.00	0.0496 $\pm$ 0.00	0.0235
MMD	0.0422 $\pm$ 0.00	0.0168 $\pm$ 0.00	0.0021 $\pm$ 0.00	0.0144 $\pm$ 0.00	0.0561 $\pm$ 0.00	0.0263
CORAL	0.0357 $\pm$ 0.00	0.0145 $\pm$ 0.00	0.0031 $\pm$ 0.00	0.0120 $\pm$ 0.00	0.0503 $\pm$ 0.00	0.0231
DANN	0.0399 $\pm$ 0.00	0.0185 $\pm$ 0.00	0.0022 $\pm$ 0.00	0.0141 $\pm$ 0.00	0.0546 $\pm$ 0.00	0.0259
MTL	0.0393 $\pm$ 0.00	0.0673 $\pm$ 0.00	0.1207 $\pm$ 0.00	0.2026 $\pm$ 0.00	0.2981 $\pm$ 0.00	0.1456
SD	<b>0.0307</b> $\pm$ 0.00	<b>0.0105</b> $\pm$ 0.00	0.0028 $\pm$ 0.00	0.0135 $\pm$ 0.00	0.0558 $\pm$ 0.00	0.0227
SelfReg	0.0369 $\pm$ 0.00	0.0130 $\pm$ 0.00	0.0033 $\pm$ 0.00	0.0122 $\pm$ 0.00	0.0507 $\pm$ 0.00	0.0232
Transfer	0.0393 $\pm$ 0.00	0.0673 $\pm$ 0.00	0.0823 $\pm$ 0.03	0.2026 $\pm$ 0.00	0.2981 $\pm$ 0.00	0.1379
RSD	0.0423 $\pm$ 0.00	0.0181 $\pm$ 0.00	0.0024 $\pm$ 0.00	0.0145 $\pm$ 0.00	0.0549 $\pm$ 0.00	0.0264
CAD	0.0393 $\pm$ 0.00	0.2294 $\pm$ 0.13	0.1207 $\pm$ 0.00	0.2373 $\pm$ 0.03	0.2981 $\pm$ 0.00	0.1849
CausIRL	0.0415 $\pm$ 0.00	0.0172 $\pm$ 0.00	<b>0.0020</b> $\pm$ 0.00	0.0142 $\pm$ 0.00	0.0536 $\pm$ 0.00	0.0257
DDG	0.0424 $\pm$ 0.00	0.0179 $\pm$ 0.00	0.0025 $\pm$ 0.00	0.0151 $\pm$ 0.00	0.0563 $\pm$ 0.00	0.0268
MODE	0.0416 $\pm$ 0.00	0.0176 $\pm$ 0.00	0.0021 $\pm$ 0.00	0.0147 $\pm$ 0.00	0.0557 $\pm$ 0.00	0.0263
MAMR	0.0449 $\pm$ 0.01	0.0205 $\pm$ 0.01	0.0026 $\pm$ 0.00	<b>0.0069</b> $\pm$ 0.00	<b>0.0375</b> $\pm$ 0.01	<b>0.0225</b>

Table 5: Regression results on AFAD dataset with training-domain validation.

Algorithm	[15-20)	[20-25)	[25-30)	[30-35)	[35-40]	Avg
ERM	0.0483 $\pm$ 0.00	0.0151 $\pm$ 0.00	0.0032 $\pm$ 0.00	0.0139 $\pm$ 0.00	<b>0.0540</b> $\pm$ 0.00	0.0269
IRM	0.0467 $\pm$ 0.00	0.0143 $\pm$ 0.00	0.0049 $\pm$ 0.00	0.0165 $\pm$ 0.00	0.0599 $\pm$ 0.00	0.0285
MLDG	0.0474 $\pm$ 0.00	0.0160 $\pm$ 0.00	0.0031 $\pm$ 0.00	<b>0.0131</b> $\pm$ 0.00	0.0543 $\pm$ 0.00	0.0268
MMD	0.0552 $\pm$ 0.00	0.0170 $\pm$ 0.00	<b>0.0009</b> $\pm$ 0.00	0.0160 $\pm$ 0.00	0.0615 $\pm$ 0.00	0.0301
CORAL	0.0481 $\pm$ 0.00	0.0157 $\pm$ 0.00	0.0031 $\pm$ 0.00	0.0138 $\pm$ 0.00	0.0555 $\pm$ 0.00	0.0272
DANN	0.0537 $\pm$ 0.00	0.0163 $\pm$ 0.00	0.0011 $\pm$ 0.00	0.0153 $\pm$ 0.00	0.0587 $\pm$ 0.00	0.0290
SD	0.0342 $\pm$ 0.00	0.0124 $\pm$ 0.00	0.0026 $\pm$ 0.00	0.0194 $\pm$ 0.00	0.0667 $\pm$ 0.00	0.0270
MTL	0.3914 $\pm$ 0.00	0.2936 $\pm$ 0.00	0.1990 $\pm$ 0.00	0.1168 $\pm$ 0.00	0.0601 $\pm$ 0.00	0.2122
SelfReg	0.0499 $\pm$ 0.00	0.0167 $\pm$ 0.00	0.0028 $\pm$ 0.00	0.0132 $\pm$ 0.00	0.0579 $\pm$ 0.00	0.0281
Transfer	0.3914 $\pm$ 0.00	0.2936 $\pm$ 0.00	0.1990 $\pm$ 0.00	0.1168 $\pm$ 0.00	0.0601 $\pm$ 0.00	0.2122
RSD	0.0506 $\pm$ 0.00	0.0194 $\pm$ 0.00	0.0042 $\pm$ 0.00	0.0171 $\pm$ 0.00	0.0576 $\pm$ 0.00	0.0298
CAD	0.3915 $\pm$ 0.00	0.2936 $\pm$ 0.00	0.1990 $\pm$ 0.00	0.1168 $\pm$ 0.00	0.0601 $\pm$ 0.00	0.2122
CausIRL	0.0505 $\pm$ 0.00	0.0178 $\pm$ 0.00	0.0010 $\pm$ 0.00	0.0157 $\pm$ 0.00	0.0632 $\pm$ 0.00	0.0296
DDG	0.0556 $\pm$ 0.00	0.0166 $\pm$ 0.00	0.0012 $\pm$ 0.00	0.0164 $\pm$ 0.00	0.0610 $\pm$ 0.00	0.0302
MODE	0.0546 $\pm$ 0.00	0.0166 $\pm$ 0.00	0.0008 $\pm$ 0.00	0.0161 $\pm$ 0.00	0.0614 $\pm$ 0.01	0.0299
MAMR	<b>0.0281</b> $\pm$ 0.00	<b>0.0068</b> $\pm$ 0.00	0.0012 $\pm$ 0.00	0.0190 $\pm$ 0.00	0.0641 $\pm$ 0.00	<b>0.0238</b>

## References

- [1] V. Vapnik., *The nature of statistical learning theory*. Springer science business media, 1999. **1**
- [2] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, “Invariant risk minimization,” 2019. **1**
- [3] H. Li, S. J. Pan, S. Wang, and A. C. Kot, “Domain generalization with adversarial feature learning,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018. **1**
- [4] G. Blanchard, A. A. Deshmukh, U. Dogan, G. Lee, and C. Scott, “Domain generalization by marginal transfer learning,” *J. Mach. Learn. Res.*, vol. 22, jan 2021. **1**
- [5] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, “Learning to generalize: Meta-learning for domain generalization,” in *AAAI Conference on Artificial Intelligence*, 2018. **1**
- [6] Y. Du, X. Zhen, L. Shao, and C. G. M. Snoek, “Metanorm: Learning to normalize few-shot batches across domains,” in *International Conference on Learning Representations*, 2021. **1**
- [7] Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker, “Domain generalization via model-agnostic learning of semantic features,” in *Advances in Neural Information Processing Systems* (H. Wallach,

- 116 H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates,  
117 Inc., 2019. 1
- 118 [8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky,  
119 “Domain-adversarial training of neural networks,” *Journal of Machine Learning Research*, vol. 17, no. 59,  
120 pp. 1–35, 2016. 1
- 121 [9] M. Pezeshki, S.-O. Kaba, Y. Bengio, A. Courville, D. Precup, and G. Lajoie, “Gradient starvation: A learn-  
122 ing proclivity in neural networks,” in *Advances in Neural Information Processing Systems* (A. Beygelzimer,  
123 Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), 2021. 1
- 124 [10] X. Chen, S. Wang, J. Wang, and M. Long, “Representation subspace distance for domain adaptation  
125 regression,” in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and  
126 T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 1749–1759, PMLR, 18–24 Jul  
127 2021. 1
- 128 [11] D. Kim, Y. Yoo, S. Park, J. Kim, and J. Lee, “Selfreg: Self-supervised contrastive regularization for  
129 domain generalization,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*,  
130 pp. 9619–9628, 2021. 1
- 131 [12] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in  
132 *International Conference on Learning Representations*, 2018. 1
- 133 [13] G. Zhang, H. Zhao, Y. Yu, and P. Poupart, “Quantifying and improving transferability in domain general-  
134 ization,” *Advances in Neural Information Processing Systems*, 2021. 1
- 135 [14] R. Dai, Y. Zhang, Z. Fang, B. Han, and X. Tian, “Moderately distributional exploration for domain  
136 generalization,” in *Proceedings of the 40th International Conference on Machine Learning*, 2023. 1
- 137 [15] H. Zhang, Y.-F. Zhang, W. Liu, A. Weller, B. Schölkopf, and E. P. Xing, “Towards principled disentangle-  
138 ment for domain generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and  
139 Pattern Recognition*, pp. 8024–8034, 2022. 2
- 140 [16] Y. Ruan, Y. Dubois, and C. J. Maddison, “Optimal representations for covariate shift,” in *International  
141 Conference on Learning Representations*, 2022. 2
- 142 [17] B. Sun and K. Saenko, “Deep coral: Correlation alignment for deep domain adaptation,” in *ECCV 2016  
143 Workshops*, 2016. 2
- 144 [18] M. Chevalley, C. Bunne, A. Krause, and S. Bauer, “Invariant causal mechanisms through distribution  
145 matching,” 2022. 2
- 146 [19] X. Liu, S. Li, Y. Ge, P. Ye, J. You, and J. Lu, “Recursively conditional gaussian for ordinal unsupervised  
147 domain adaptation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*,  
148 pp. 764–773, October 2021. 4
- 149 [20] X. Liu, S. Li, Y. Ge, P. Ye, J. You, and J. Lu, “Ordinal unsupervised domain adaptation with recursively  
150 conditional gaussian imposed variational disentanglement,” *IEEE Transactions on Pattern Analysis and  
151 Machine Intelligence*, pp. 1–14, 2022. 4