分布外问题及其应对策略

——大作业展示

Contents

OoD概述

Out-of-Distribution

i.i.d.

独立同分布

(Independent and Identically Distributed)

数据分布

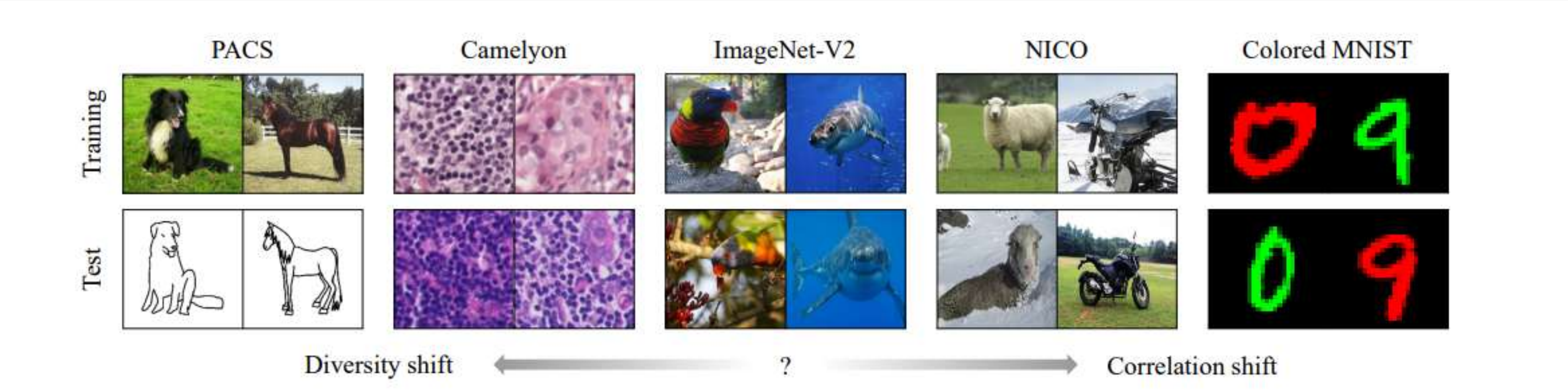
OoD

分布外

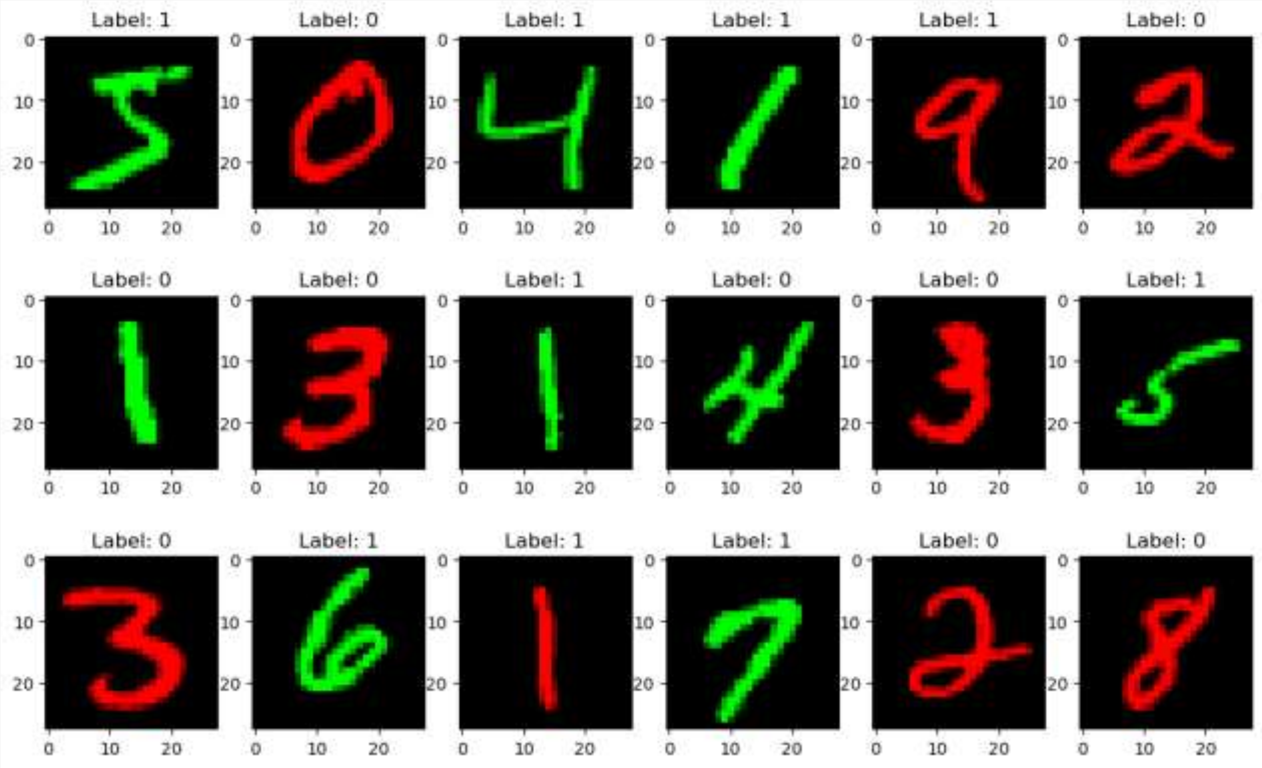
(Out-of-Distribution)

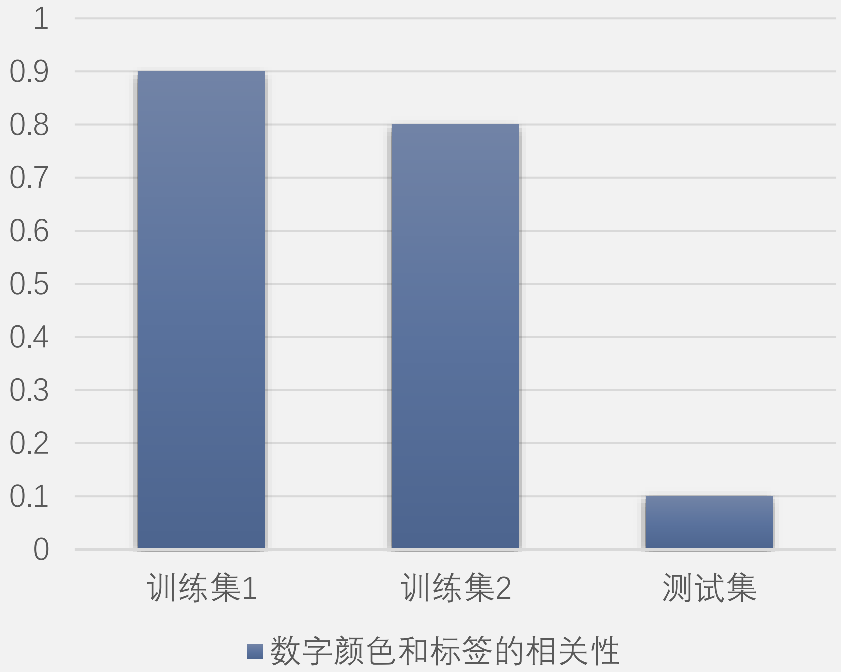
Out-of-Distribution

Diversity shift ⟸ Out-of-Distribution ⟹ Correlation shift



Colored MNIST

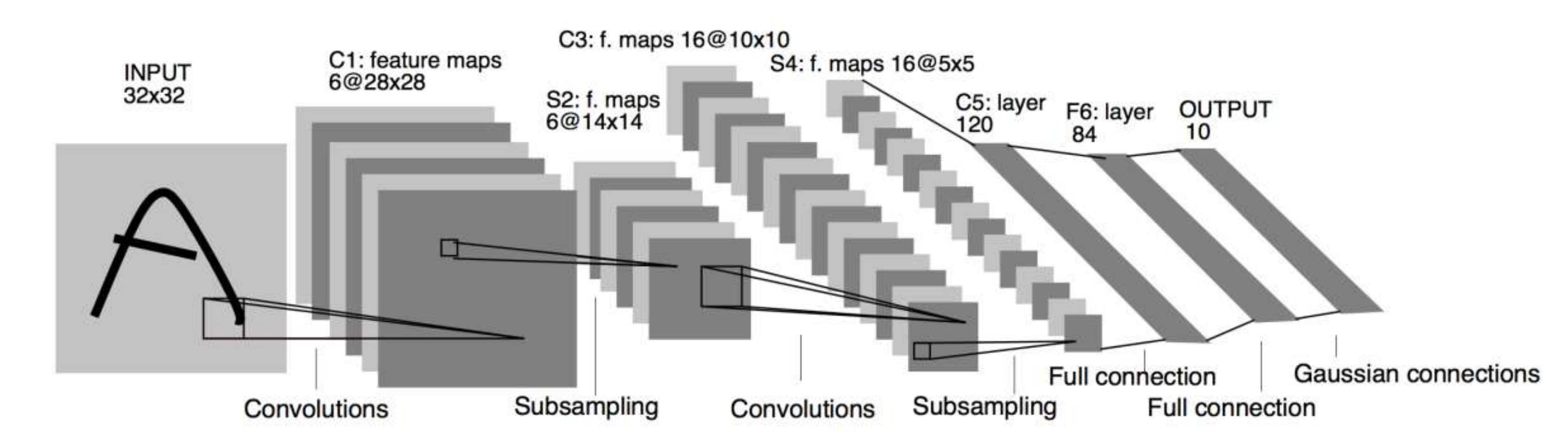




预处理

LeNet

- 输入层：INPUT
- 卷积层：C1、C3、C5
- 池化层：S2、S4
- 全连接层：F6
- 输出层：OUTPUT



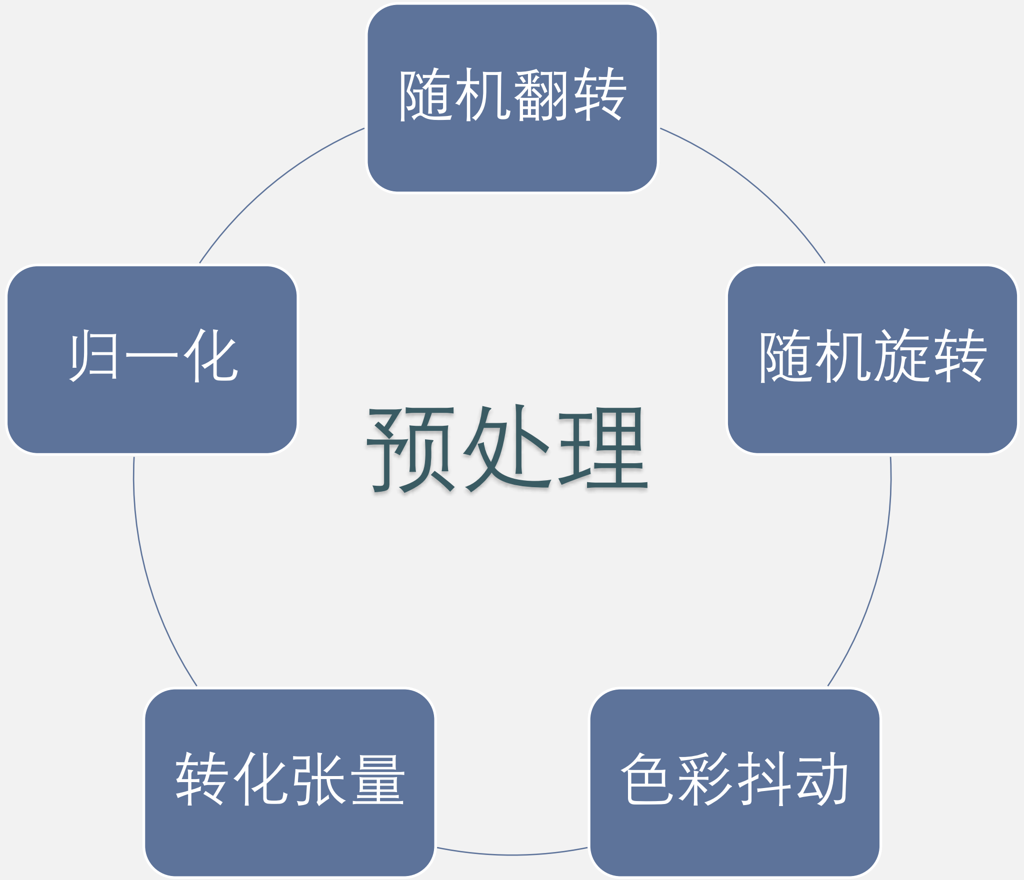
LeNet网络结构

在Colored MNIST上训练和测试LeNet

- 优化器(Optimizer)：SGD
- 损失函数(Loss function)：Cross-entropy loss function
- 批尺寸(Bach size)：128
- 训练次数(Epoch)：50

```
Colored MNIST dataset already exists
Colored MNIST dataset already exists
Epoch 10/50, Train Loss: 0.3521, Train Accuracy: 89.89%, Test Loss: 1.8687, Test Accuracy: 10.20%
Epoch 20/50, Train Loss: 0.3404, Train Accuracy: 89.89%, Test Loss: 2.0372, Test Accuracy: 10.20%
Epoch 30/50, Train Loss: 0.3380, Train Accuracy: 89.89%, Test Loss: 2.0222, Test Accuracy: 10.20%
Epoch 40/50, Train Loss: 0.3362, Train Accuracy: 89.89%, Test Loss: 2.0147, Test Accuracy: 10.20%
Epoch 50/50, Train Loss: 0.3345, Train Accuracy: 89.89%, Test Loss: 2.0145, Test Accuracy: 10.20%
```

预处理

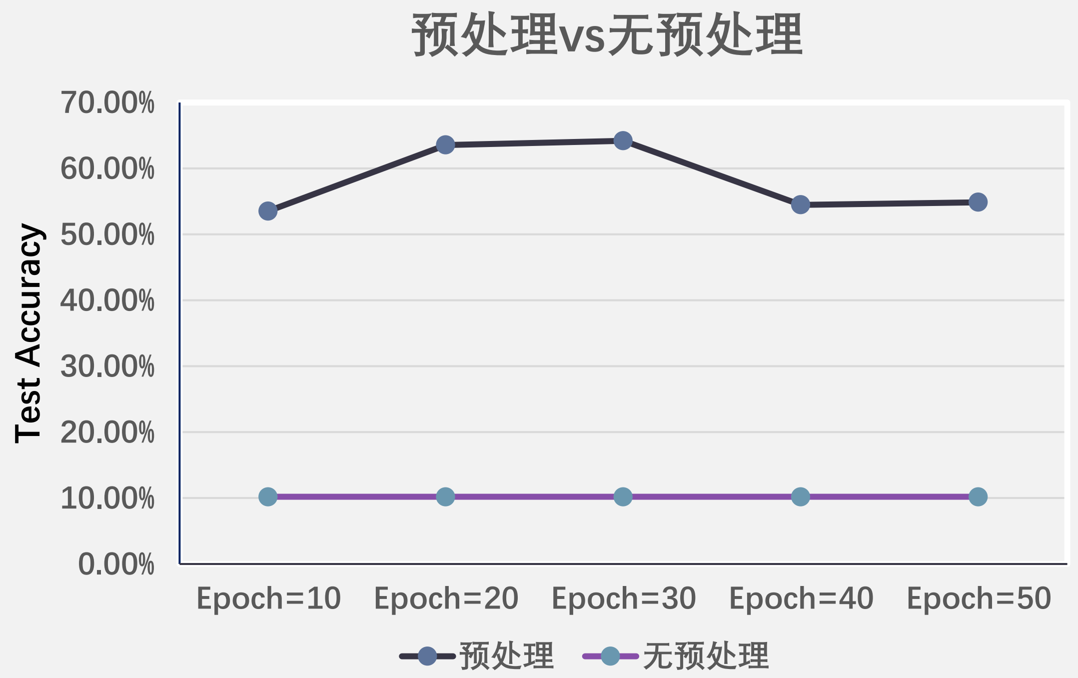


预处理

- 结果:

```
Colored MNIST dataset already exists
Colored MNIST dataset already exists
Epoch 10/50, Train Loss: 0.6901, Train Accuracy: 53.79%, Test Loss: 0.6819, Test Accuracy: 53.54%
Epoch 20/50, Train Loss: 0.6858, Train Accuracy: 55.62%, Test Loss: 0.6774, Test Accuracy: 63.55%
Epoch 30/50, Train Loss: 0.6834, Train Accuracy: 55.80%, Test Loss: 0.6745, Test Accuracy: 64.19%
Epoch 40/50, Train Loss: 0.6811, Train Accuracy: 56.41%, Test Loss: 0.6788, Test Accuracy: 54.48%
Epoch 50/50, Train Loss: 0.6779, Train Accuracy: 57.63%, Test Loss: 0.6802, Test Accuracy: 54.86%
```

- 对比图:



预处理

- 删去色彩抖动:

```
Colored MNIST dataset already exists
Colored MNIST dataset already exists
Epoch 10/50, Train Loss: 0.3438, Train Accuracy: 89.89%, Test Loss: 2.0837, Test Accuracy: 10.20%
Epoch 20/50, Train Loss: 0.3415, Train Accuracy: 89.89%, Test Loss: 2.0616, Test Accuracy: 10.20%
Epoch 30/50, Train Loss: 0.3401, Train Accuracy: 89.89%, Test Loss: 2.0514, Test Accuracy: 10.20%
Epoch 40/50, Train Loss: 0.3387, Train Accuracy: 89.89%, Test Loss: 2.0471, Test Accuracy: 10.20%
Epoch 50/50, Train Loss: 0.3374, Train Accuracy: 89.88%, Test Loss: 2.0461, Test Accuracy: 10.20%
```

- 保留色彩抖动:

```
Colored MNIST dataset already exists
Colored MNIST dataset already exists
Epoch 10/50, Train Loss: 0.6901, Train Accuracy: 53.79%, Test Loss: 0.6819, Test Accuracy: 53.54%
Epoch 20/50, Train Loss: 0.6858, Train Accuracy: 55.62%, Test Loss: 0.6774, Test Accuracy: 63.55%
Epoch 30/50, Train Loss: 0.6834, Train Accuracy: 55.80%, Test Loss: 0.6745, Test Accuracy: 64.19%
Epoch 40/50, Train Loss: 0.6811, Train Accuracy: 56.41%, Test Loss: 0.6788, Test Accuracy: 54.48%
Epoch 50/50, Train Loss: 0.6779, Train Accuracy: 57.63%, Test Loss: 0.6802, Test Accuracy: 54.86%
```

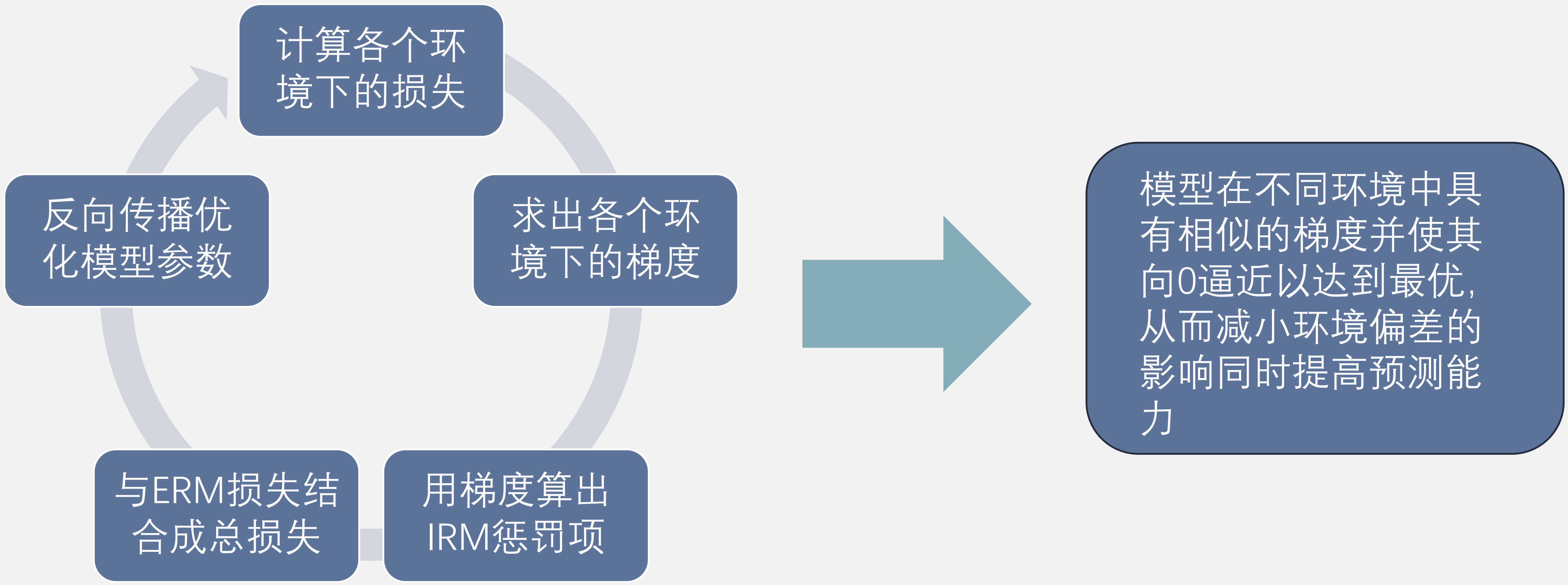
可能的处理方法

- 随机调整数字颜色
- 数字颜色的统一化
- 数字背景分离
- 多任务学习
- 模型集成
- 预训练

IRM

IRM

- 目标：让模型学习到causal feature并利用它来决定结果，减小spurious correlation的干扰
- 方式：引入IRM惩罚项



IRM

- 结果:

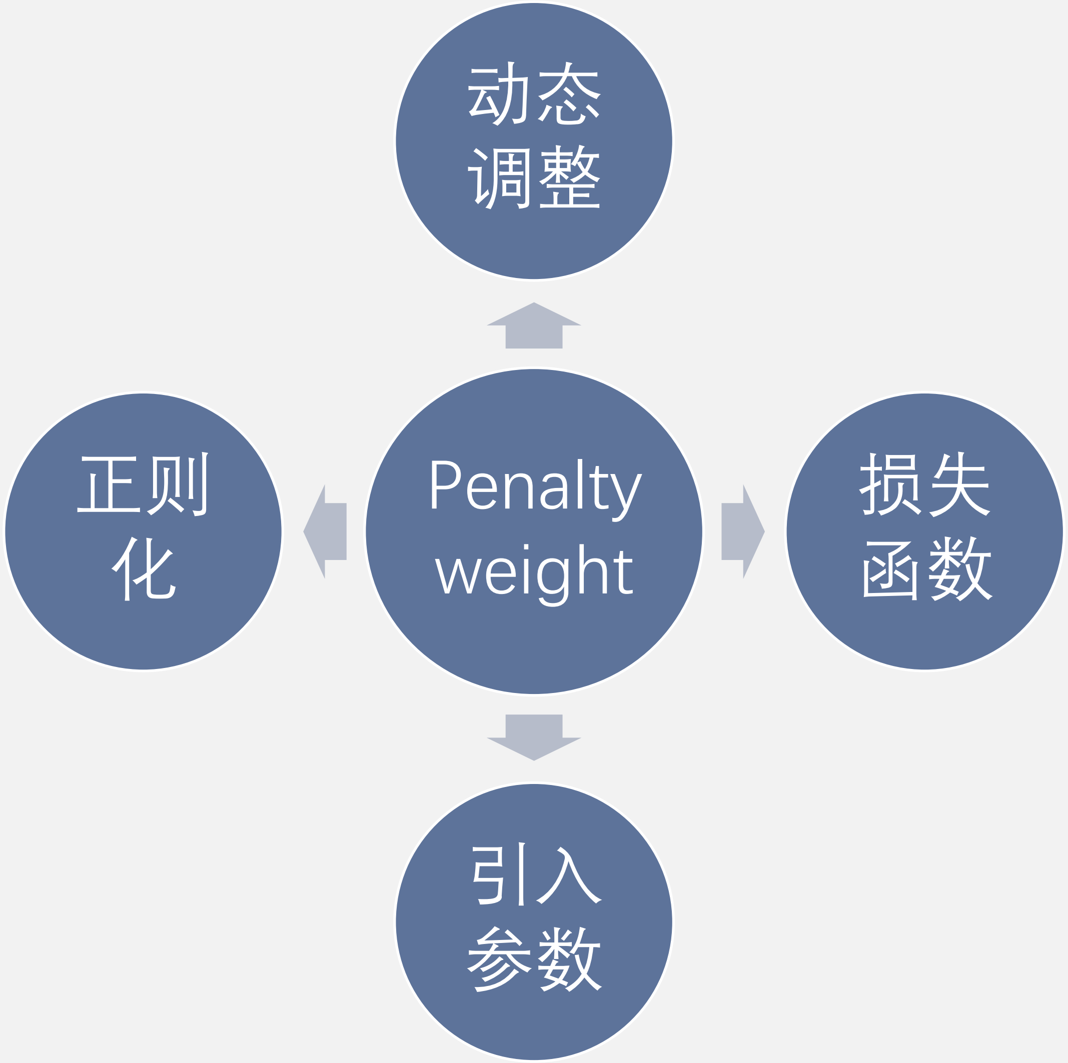
```
Performance on train1 set: Average loss: 0.5201, Accuracy: 14845/20000 (74.22%)

Performance on train2 set: Average loss: 0.5071, Accuracy: 14863/20000 (74.31%)

Performance on test set: Average loss: 0.8663, Accuracy: 12311/20000 (61.55%)
```

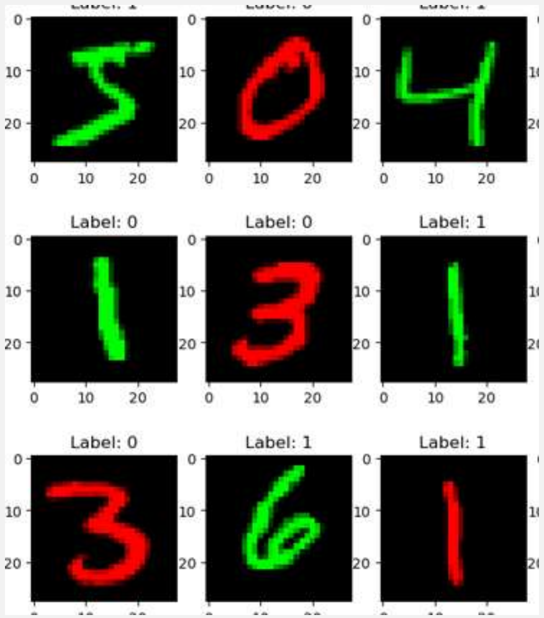
Epoch = 35

IRM算法的改进



VREx

VREx



Colored MNIST

Algorithm	Colored MNIST	CelebA	NICO	Average	Prev score	Ranking score
VREx [37]	$56.3 \pm 1.9^{\uparrow}$	87.3 ± 0.2	71.0 ± 1.3	71.5	-1	+1
GroupDRO [62]	$32.5 \pm 0.2^{\uparrow}$	87.5 ± 1.1	71.8 ± 0.8	63.9	-1	+1
ERM [68]	29.9 ± 0.9	87.2 ± 0.6	71.4 ± 1.3	62.8	0	0
MTL [15]	29.3 ± 0.1	87.0 ± 0.7	70.2 ± 0.6	62.2	-2	0
ERDG [79]	$31.6 \pm 1.3^{\uparrow}$	$84.5 \pm 0.2^{\downarrow}$	70.6 ± 1.3	62.2	-2	0
ARM [78]	$34.6 \pm 1.8^{\uparrow}$	86.6 ± 0.7	$63.9 \pm 1.8^{\downarrow}$	61.7	-3	0
MMD [41]	$50.7 \pm 0.1^{\uparrow}$	$86.0 \pm 0.5^{\downarrow}$	$68.3 \pm 1.0^{\downarrow}$	68.3	+2	-1
IGA [36]	29.7 ± 0.5	$86.2 \pm 0.7^{\downarrow}$	70.5 ± 1.2	62.1	0	-1
IRM [7]	$60.2 \pm 2.4^{\uparrow}$	$85.4 \pm 1.2^{\downarrow}$	$67.6 \pm 1.4^{\downarrow}$	71.1	-1	-1
MLDG [40]	$32.7 \pm 1.1^{\uparrow}$	$85.4 \pm 1.3^{\downarrow}$	$51.6 \pm 6.1^{\downarrow}$	56.6	-4	-1
SagNet [49]	30.5 ± 0.7	$85.8 \pm 1.4^{\downarrow}$	$69.3 \pm 1.0^{\downarrow}$	61.9	+1	-2
CORAL [64]	30.0 ± 0.5	$86.3 \pm 0.5^{\downarrow}$	$68.3 \pm 1.4^{\downarrow}$	61.5	-1	-2
ANDMask [51]	$27.2 \pm 1.4^{\downarrow}$	$86.2 \pm 0.2^{\downarrow}$	72.2 ± 1.2	61.9	-2	-2
Mixup [76]	$27.6 \pm 1.8^{\downarrow}$	87.5 ± 0.5	$66.6 \pm 0.9^{\downarrow}$	60.6	-2	-2
RSC [34]	$28.6 \pm 1.5^{\downarrow}$	$85.9 \pm 0.2^{\downarrow}$	$69.7 \pm 0.3^{\downarrow}$	61.4	+2	-3
DANN [24]	$24.5 \pm 0.8^{\downarrow}$	$86.0 \pm 0.4^{\downarrow}$	$68.6 \pm 1.1^{\downarrow}$	59.7	-2	-3
Average	34.5	68.4	86.4	63.1	–	–

Performance of ERM and OoD generalization algorithms on datasets dominated by **correlation shift**

VREx

- 目标：解决分布偏移问题，提高模型分布外泛化能力
- 方式：引入VREx惩罚项——训练风险的方差，通过它减小各训练领域的风险差异
- 公式：

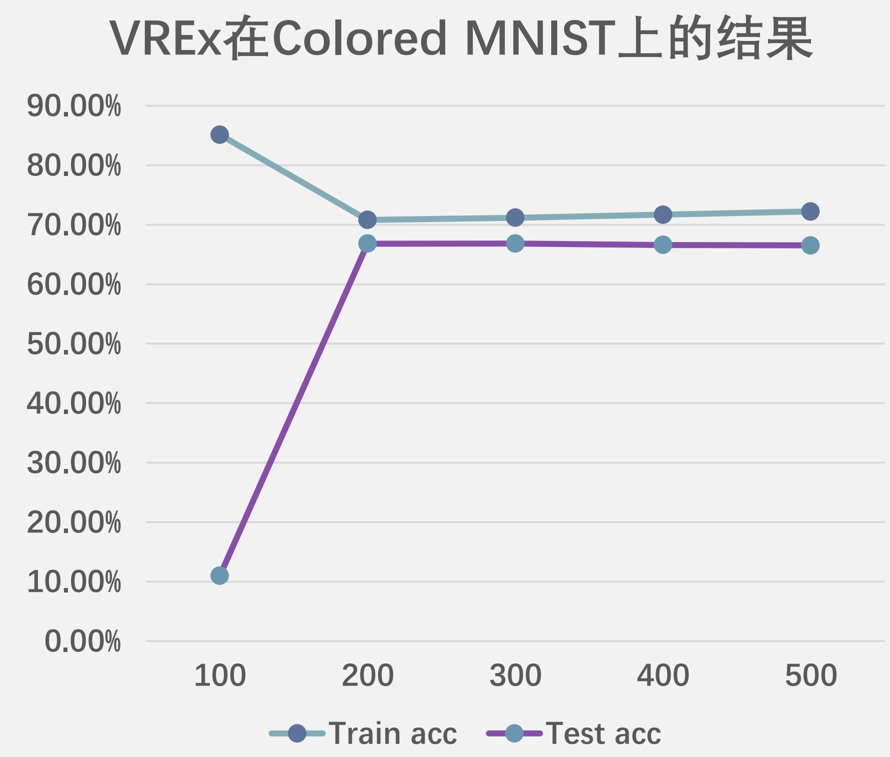
$$\mathcal{R}_{\text{V-REx}}(\theta) \doteq \beta \,\mathrm{Var}(\{\mathcal{R}_1(\theta), \ldots, \mathcal{R}_m(\theta)\}) + \sum_{e=1}^{m} \mathcal{R}_e(\theta)$$

$\beta \in [0,\infty)$，β的取值决定了是向降低平均风险还是向令各领域的风险一致优化。当β=0时，就变成了普通的ERM，而当β -> ∞时V-REx就迫使各个领域的风险严格一致，使方差为0。通过调整合适的值，可以获得最佳性能的模型。

VREx

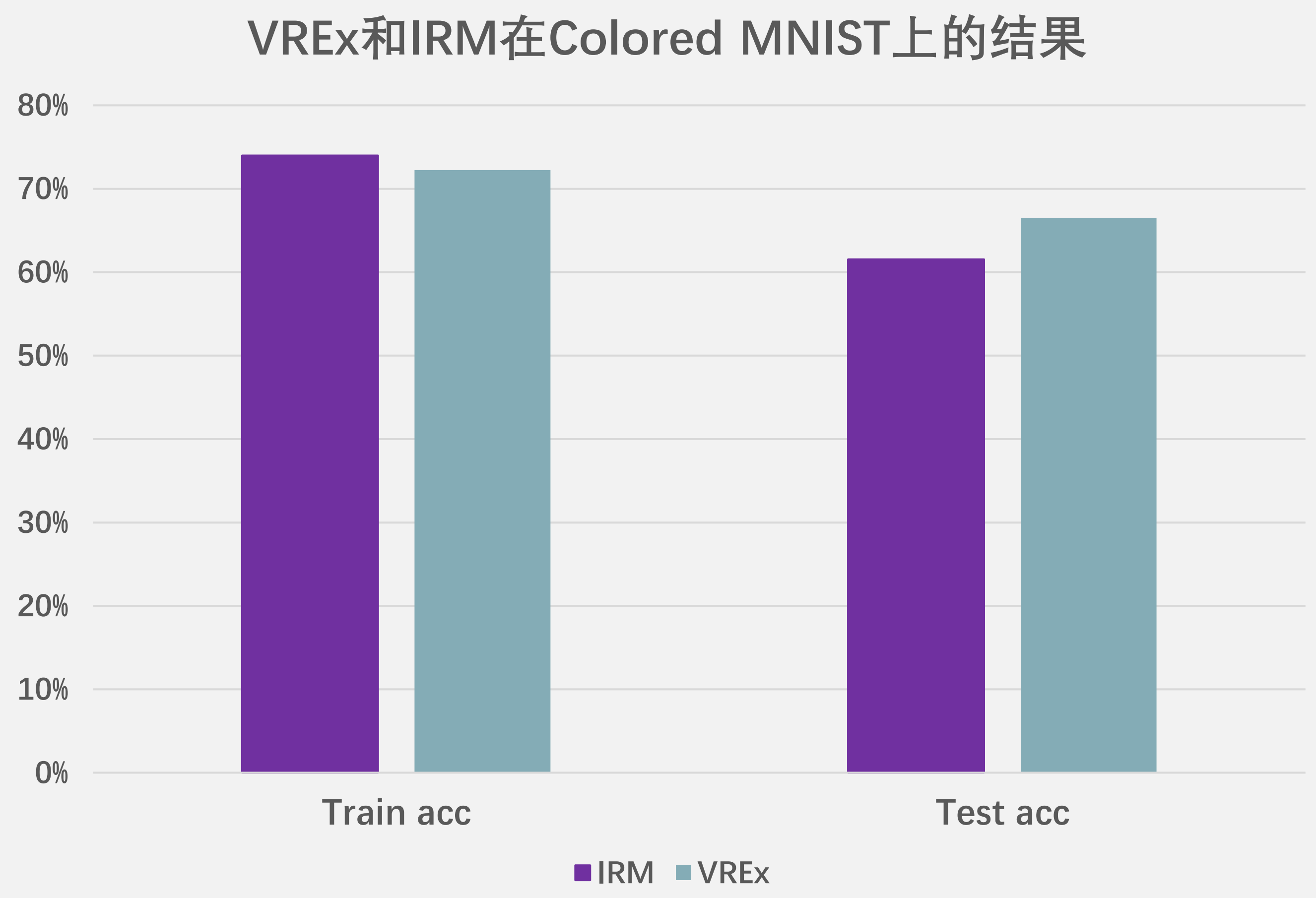
- 结果

```
Restart 0
step            train nll       train acc       rex penalty     irmv1 penalty
test acc
0               0.68907         0.54244         4.57316e-06     8.45346e-06
0.44890
100             0.36482         0.85138         0.01908         0.00399
0.10990
200             0.57021         0.70808         7.42565e-06     2.81167e-07
0.66830
300             0.56621         0.71182         7.70136e-06     1.82156e-07
0.66840
400             0.56065         0.71680         1.00755e-05     1.89763e-07
0.66600
500             0.55410         0.72240         1.35928e-05     2.03200e-07
0.66520
highest test acc this run: 0.6684
Final train acc (mean/std across restarts so far):
0.7224 0.0
Final test acc (mean/std across restarts so far):
0.6652 0.0
Highest test acc (mean/std across restarts so far):
```



VREx vs IRM

- 在Colored MNIST上的结果：

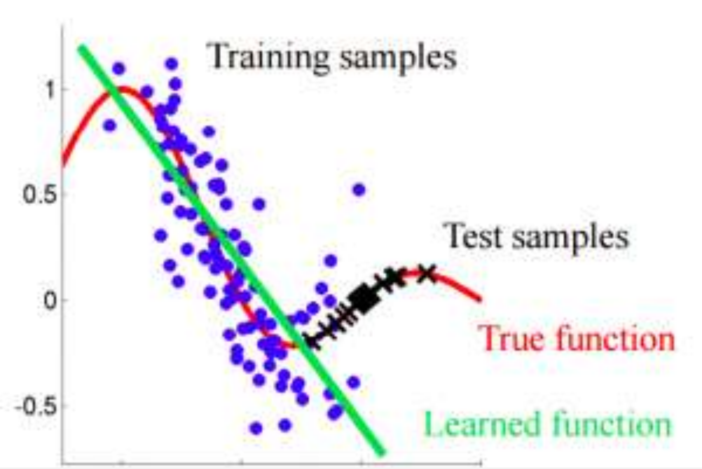


VREx vs IRM

- 协变量转变(Covariate shift)

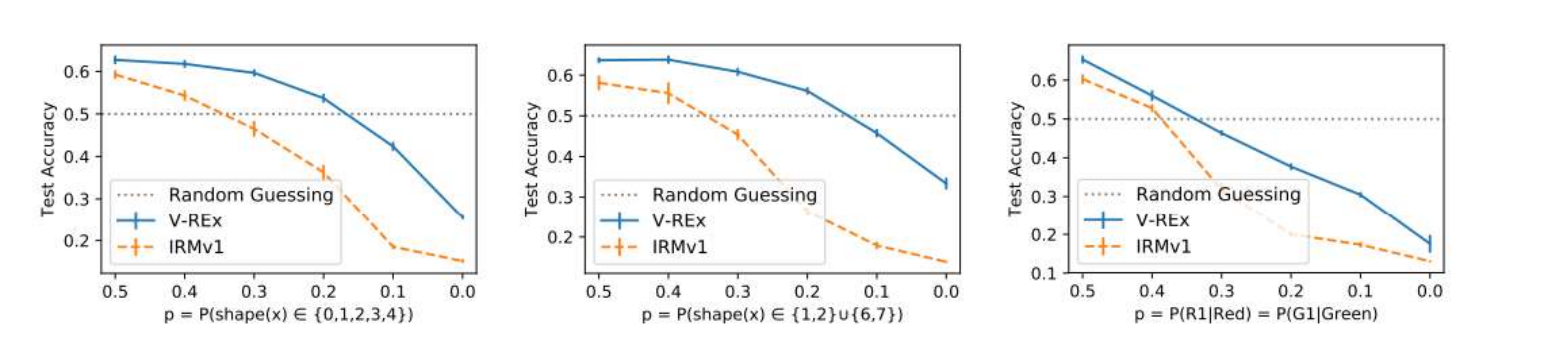
Method	Invariant Prediction	Cov. Shift Robustness	Suitable for Deep Learning
DRO	✗	✓	✓
(C-)ADA	✗	✓	✓
ICP	✓	✗	✗
IRM	✓	✗	✓
REx	✓	✓	✓

- 协变量转变：训练数据和测试数据之间的输入特征分布发生变化



VREx vs IRM

- 在协变量转变下的Colored MNIST上的结果:



三幅图分别带表三种不同的协变量转变情况：

1. Class imbalance: varying p = P(shape(x) ∈ {0, 1, 2, 3, 4}).
2. Digit imbalance: varying p = P(shape(x) ∈ {1, 2} ∪ {6, 7}); digits 0 and 5 are removed.
3. Color imbalance: We use 2 versions of each color, for 4 total channels: R1, R2, G1, G2. We vary p = P(R1|Red) = P(G1|Green).

参考

[1] Ye N, Li K, Bai H, et al. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 7947-7958.
[2] Krueger D, Caballero E, Jacobsen J H, et al. Out-of-distribution generalization via risk extrapolation (rex)[C]//International Conference on Machine Learning. PMLR, 2021: 5815-5826.
[3] https://www.analyticsvidhya.com/blog/2017/07/covariate-shift-the-hidden-problem-of-real-world-data-science/
[4] Arjovsky M, Bottou L, Gulrajani I, et al. Invariant risk minimization[J]. arXiv preprint arXiv:1907.02893, 2019.
[5] https://baike.baidu.com/item/LeNet-5/61427772?fr=ge_ala

Thanks