Improving Out-of-Distribution Robustness via Selective Augmentation

Huaxiu Yao [*1] **Yu Wang** [*2] **Sai Li** [3] **Linjun Zhang** [4] **Weixin Liang** [1]
James Zou [1] **Chelsea Finn** [1]

ICML 2022

Background

Machine learning algorithms typically assume that training and test examples are drawn from the same distribution. However, distribution shift is a common problem in real-world applications and can cause models to perform dramatically worse at test time.

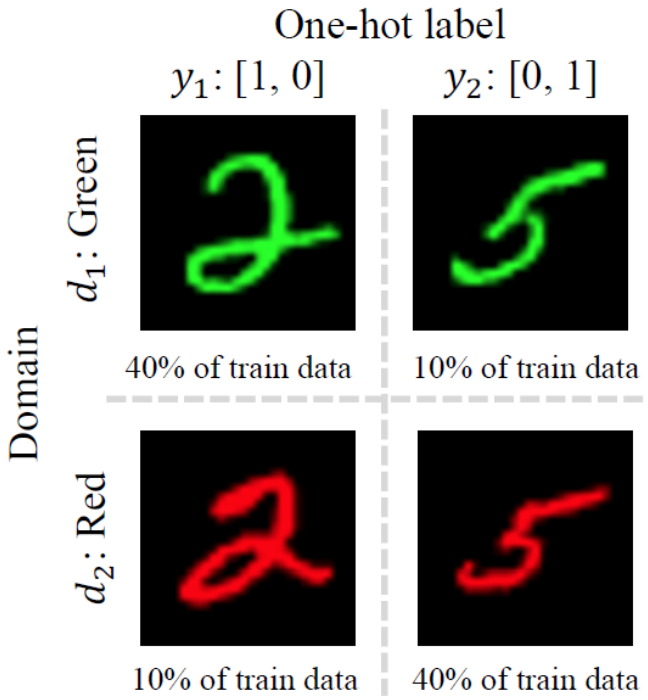
Here, we specifically consider the problems of subpopulation shifts and domain shifts:

1. In subpopulation shifts, the test domains (or subpopulations) are seen but underrepresented in the training data, When subpopulation shift occurs, models may perform poorly when they falsely rely on spurious correlations between the particular subpopulation and the label;

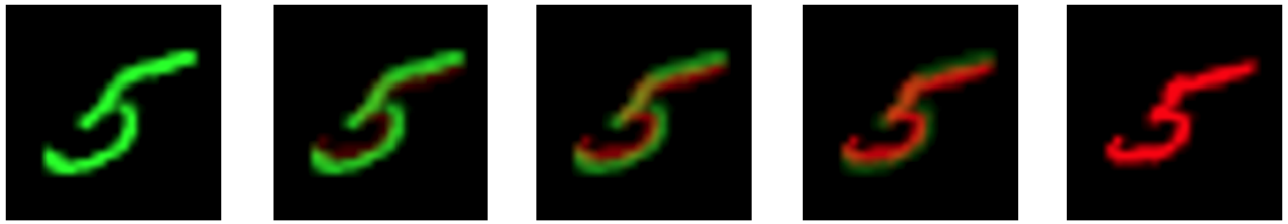
2. In domain shifts, the test data is from new domains, which requires the trained model to generalize well to test domains without seeing the data from those domains at training time.

Introduction

(a) Colored MNIST dataset



(b) Intra-label LISA: interpolates samples with the same label but different domains



$\lambda = 0.0$ $\lambda = 0.25$ $\lambda = 0.5$ $\lambda = 0.75$ $\lambda = 1.0$

All $y = [0, 1]$

(c) Intra-domain LISA: interpolates samples with the same domain but different labels

$\lambda = 0.0$ $\lambda = 0.25$ $\lambda = 0.5$ $\lambda = 0.75$ $\lambda = 1.0$

$y = [1, 0]$ $y = [0.25, 0.75]$ $y = [0.5, 0.5]$ $y = [0.75, 0.25]$ $y = [0, 1]$

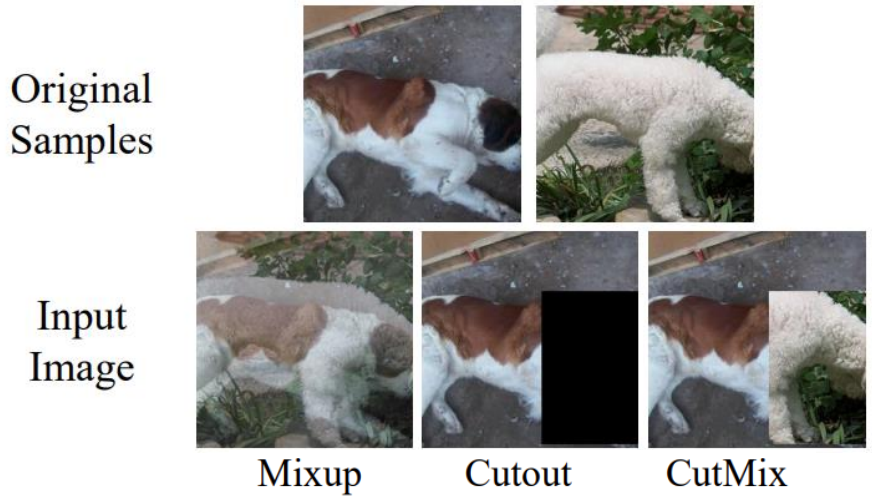
Domain information is not the reason for the label change

Method

Following (WILDS: A Benchmark of in-the-Wild Distribution Shifts [ICML2021])

we regard the overall data distribution containing $\mathcal{D} = \{1, \ldots, D\}$ domains and each domain $d \in \mathcal{D}$ is associated with a data distribution P_d over a set $(X, Y, d) = \{(x_i, y_i, d)\}_{i=1}^{N^d}$

Mixup:



$$x_{mix} = \lambda x_i + (1 - \lambda) x_j, \; y_{mix} = \lambda y_i + (1 - \lambda) y_j$$

where the interpolation ratio $\lambda \in [0, 1]$ is sampled from a Beta distribution $\text{Beta}(\alpha, \beta)$

Method

the optimization process is reformulated as

$$\theta^* := \arg\min_{\theta \in \Theta} \mathbb{E}_{\{(x_i, y_i, d_i), (x_j, y_j, d_j)\} \sim \hat{P}} [\ell(f_\theta(x_{mix}), y_{mix})].$$

Without additional selective augmentation strategies, vanilla mixup will regularize the model and reduce, allowing it to attain good indistribution generalization. However, vanilla mixup may not be able to cancel out spurious correlations, causing the model to still fail at attaining good OOD generalization.

Method

In LISA, we instead adopt a new strategy where mixup is only applied across specific domains or groups, which leans towards learning invariant predictors and thus better OOD performance. Specifically, the two kinds of selective augmentation strategies are presented as:

1.Intra-label LISA (LISA-L): Interpolating samples with the same label

this produces datapoints that have both domains partially present, effectively eliminating spurious correlations between domain and label in cases where the pair of domains correlate differently with the label.

2. Intra-domain LISA (LISA-D): Interpolating samples with the same domain

This causes the model to make predictions that are less dependent on the domain, again improving OOD robustness.

Algorithm

Require: Training data $\mathcal{D}$, step size η, learning rate γ, shape parameters α, β of Beta distribution

while not converge **do**
  Sample $\lambda \sim \text{Beta}(\alpha, \beta)$
  Sample minibatch $B_1 \sim \mathcal{D}$
  Initialize $B_2 \leftarrow \{\}$
  Select strategy $s \sim Bernoulli(p_{sel})$
  if s is True **then**
    for $(x_i, y_i, d_i) \in B_1$ **do**
      Randomly sample $(x_j, y_j, d_j) \sim \{(x, y, d) \in \mathcal{D}\}$ which satisfies $(y_i = y_j)$ and $(d_i \neq d_j)$.
      Put (x_j, y_j, d_j) into B_2.
  else
    for $(x_i, y_i, d_i) \in B_1$ **do**
      Randomly sample $(x_j, y_j, d_j) \sim \{(x, y, d) \in \mathcal{D}\}$ which satisfies $(y_i \neq y_j)$ and $(d_i = d_j)$.
      Put (x_j, y_j, d_j) into B_2.
  Update θ with data $\lambda B_1 + (1 - \lambda) B_2$ with learning rate γ.

Experiment

Evaluating Robustness to Subpopulation Shifts

Table 1. Dataset Statistics for Subpopulation Shifts. All datasets are binary classification tasks and we use the worst group accuracy as the evaluation metric.

Datasets	Domains	Model Architecture	Class Information
CMNIST	2 digit colors	ResNet-50	digit (0,1,2,3,4) v.s. (5,6,7,8,9)
Waterbirds	2 backgrounds	ResNet-50	waterbirds v.s. landbirds
CelebA	2 hair colors	ResNet-50	man v.s. women
CivilComments	8 demographic identities	DistilBERT-uncased	toxic v.s. non-toxic

In these datasets, the domain information is highly spurious correlated with the label information.

Experiment

Evaluating Robustness to Subpopulation Shifts

	CMNIST		Waterbirds		CelebA		CivilComments	
	Avg.	Worst	Avg.	Worst	Avg.	Worst	Avg.	Worst
ERM	27.8%	0.0%	97.0%	63.7%	94.9%	47.8%	92.2%	56.0%
UW	72.2%	66.0%	95.1%	88.0%	92.9%	83.3%	89.8%	69.2%
IRM	72.1%	70.3%	87.5%	75.6%	94.0%	77.8%	88.8%	66.3%
IB-IRM	72.2%	70.7%	88.5%	76.5%	93.6%	85.0%	89.1%	65.3%
V-REx	71.7%	70.2%	88.0%	73.6%	92.2%	86.7%	90.2%	64.9%
CORAL	71.8%	69.5%	90.3%	79.8%	93.8%	76.9%	88.7%	65.6%
GroupDRO	72.3%	68.6%	91.8%	**90.6%**	92.1%	87.2%	89.9%	70.0%
DomainMix	51.4%	48.0%	76.4%	53.0%	93.4%	65.6%	90.9%	63.6%
Fish	46.9%	35.6%	85.6%	64.0%	93.1%	61.2%	89.8%	71.1%
LISA (ours)	74.0%	**73.3%**	91.8%	89.2%	92.4%	**89.3%**	89.2%	**72.6%**

Table 2. Results of subpopulation shifts. Here, we show the average and worst group accuracy. We repeat the experiments three times and put full results with standard deviation in Table 10.

Experiment

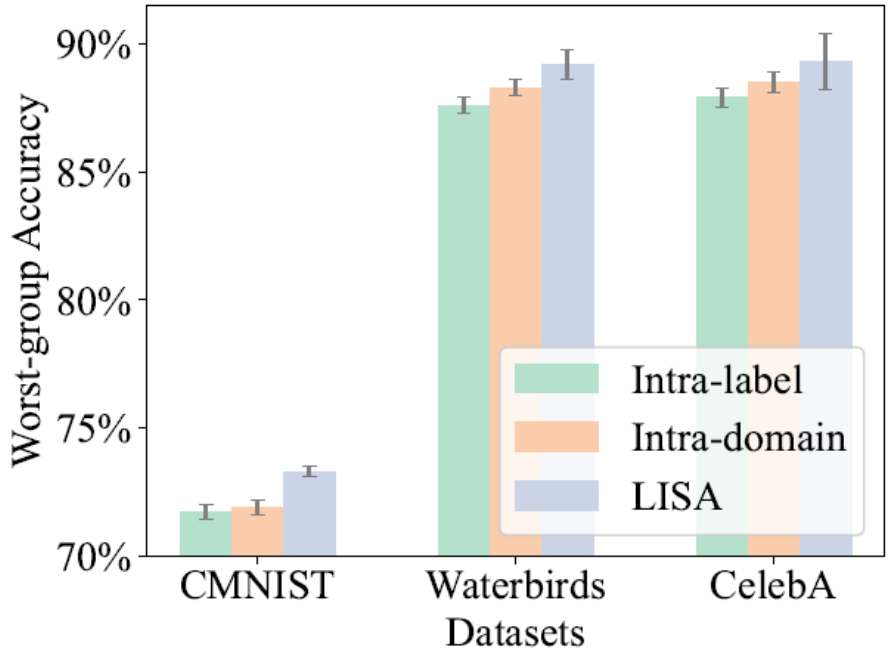


Figure 2. Effects of intra-label and intra-domain LISA in CMNIST, Waterbirds and CelebA. The experiments are repeated three times with different seeds.

Experiment

Evaluating Robustness to Domain Shifts

Table 4. Dataset Statistics for Domain Shifts.

Datasets	Domains	Metric	Base Model	Num. of classes
Camelyon17	5 hospitals	Avg. Acc.	DenseNet-121	2
FMoW	16 years x 5 regions	Worst-group Acc.	DenseNet-121	62
RxRx1	51 experimental batches	Avg. Acc.	ResNet-50	1,139
Amazon	7,676 reviewers	10th Percentile Acc.	DistilBERT-uncased	5
MetaShift	4 backgrounds	Worst-group Acc.	ResNet-50	2

Experiment

Evaluating Robustness to Domain Shifts

Table 3. Main domain shifts results. LISA outperforms prior methods on all five datasets. Following the instructions of Koh et al. (2021), we report the performance of Camelyon17 over 10 different seeds and the results of other datasets are obtained over 3 different seeds.

	Camelyon17	FMoW	RxRx1	Amazon	MetaShift
	Avg. Acc.	Worst Acc.	Avg. Acc.	10-th Per. Acc.	Worst Acc.
ERM	70.3 ± 6.4%	32.3 ± 1.25%	29.9 ± 0.4%	53.8 ± 0.8%	52.1 ± 0.4%
IRM	64.2 ± 8.1%	30.0 ± 1.37%	8.2 ± 1.1%	52.4 ± 0.8%	51.8 ± 0.8%
IB-IRM	68.9 ± 6.1%	28.4 ± 0.90%	6.4 ± 0.6%	53.8 ± 0.7%	52.3 ± 1.0%
V-REx	71.5 ± 8.3%	27.2 ± 0.78%	7.5 ± 0.8%	53.3 ± 0.0%	51.6 ± 1.8%
CORAL	59.5 ± 7.7%	31.7 ± 1.24%	28.4 ± 0.3%	52.9 ± 0.8%	47.6 ± 1.9%
GroupDRO	68.4 ± 7.3%	30.8 ± 0.81%	23.0 ± 0.3%	53.3 ± 0.0%	51.9 ± 0.7%
DomainMix	69.7 ± 5.5%	34.2 ± 0.76%	30.8 ± 0.4%	53.3 ± 0.0%	51.3 ± 0.5%
Fish	74.7 ± 7.1%	34.6 ± 0.18%	10.1 ± 1.5%	53.3 ± 0.0%	49.2 ± 2.1%
LISA (ours)	**77.1 ± 6.5%**	**35.5 ± 0.65%**	**31.9 ± 0.8%**	**54.7 ± 0.0%**	**54.2 ± 0.7%**

Experiment

	Camelyon17	FMoW	RxRx1	Amazon	MetaShift
	Avg. Acc.	Worst Acc.	Avg. Acc.	10-th Per. Acc.	Worst Acc.
ERM	$70.3 \pm 6.4\%$	$32.8 \pm 0.45\%$	$29.9 \pm 0.4\%$	$53.8 \pm 0.8\%$	$52.1 \pm 0.4\%$
Vanilla mixup	$71.2 \pm 5.3\%$	$34.2 \pm 0.45\%$	$26.5 \pm 0.5\%$	$53.3 \pm 0.0\%$	$51.3 \pm 0.7\%$
In-group mixup	$75.5 \pm 6.7\%$	$32.2 \pm 1.18\%$	$24.4 \pm 0.2\%$	$53.8 \pm 0.6\%$	$52.7 \pm 0.5\%$
LISA (ours)	$\mathbf{77.1 \pm 6.5\%}$	$\mathbf{35.5 \pm 0.65\%}$	$\mathbf{31.9 \pm 0.8\%}$	$\mathbf{54.7 \pm 0.0\%}$	$\mathbf{54.2 \pm 0.7\%}$

- Vanilla mixup: in Vanilla mixup, we do not add any constraints on the sample selection, i.e., the mixup is performed on any pairs of samples.
- In-group mixup: this strategy applies data interpolation on samples with the same labels and from the same domains.

Thanks