Improving Out-of-Distribution Robustness via Selective Augmentation

Huaxiu Yao¹, Yu Wang², Sai Li³, Linjun Zhang⁴
Weixin Liang¹, James Zou¹, Chelsea Finn¹

¹Stanford University, ²University of California San Diego ³Renmin University of China, ⁴Rutgers University
Machine Learning Systems are Fragile

Models often fail when **domain shift** happens

Deploy model to new environment

Trained on 3 hospitals

Deploy to a new hospital
Why ML Models Fail – Spurious Correlation

\[ y_1: \text{digit} < 5 \]

\[ y_2: \text{digit} \geq 5 \]

Spurious Correlation: color

Prediction: digit < 5

True: digit ≥ 5
Why ML Models Fail – Spurious Correlation

\( y_1: \) digit < 5

\( y_2: \) digit ≥ 5

40% of train data

10% of train data

10% of train data

40% of train data

Domain-invariant Correlation:
digit information

Prediction: digit > 5

True: digit ≥ 5
Why ML Models Fail – Spurious Correlation

Building robust machine learning models that can capture domain-invariant information

\[ y_1 : \text{digit} < 5 \]
\[ y_2 : \text{digit} \geq 5 \]

Prediction: digit > 5
True: digit ≥ 5
Prior Works Focus on Explicit Regularization

Standard empirical risk minimization (ERM)

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \hat{p}} [\ell(f_\theta(x), y)]$$

loss

average over training examples

Prior approaches to learn invariant representations/predictors

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \hat{p}} [\ell(f_\theta(x), y)] + \lambda \mathcal{L}_{reg}$$

explicit regularizers to learn domain-invariant representations/predictors
Discussion of Prior Works

Best prior domain invariance method

Camelyon17

Standard ERM

70.3% → 74.7%

RxRx1

29.9% → 28.4%

[PW Koh et al. ICML 2021]
LISA: Learning Invariant Predictors with Selective Augmentation

Colored MNIST

Domain

$d_1$: Green

$\gamma_1$: digit < 5

40% of train data

$\gamma_2$: digit ≥ 5

10% of train data

$d_2$: Red

10% of train data

40% of train data

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

$\lambda \sim \text{Beta}(\alpha, \beta)$

Intra-label LISA – Interpolates samples with the same label but different domains ($d_i \neq d_j, y_i = y_j$)

Different background, same label

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

All $y = [0, 1]$
LISA: Learning Invariant Predictors with Selective Augmentation

**Mixup:**

\[ x_{mix} = \lambda x_i + (1 - \lambda) x_j, \quad y_{mix} = \lambda y_i + (1 - \lambda) y_j \]

\( \lambda \sim \text{Beta}(\alpha, \beta) \)

**Intra-domain LISA** – Interpolates samples with the same domain but different labels \((d_i = d_j, y_i \neq y_j)\)

Colored MNIST

Domain information is **not** the reason for the label change

Use \( p_{sel} \) to determine intra-label LISA or intra-domain LISA
## Performance – Subpopulation Shift

| Dataset      | Worst-group accuracy | Best prior domain invariance method | LISA  |
|--------------|----------------------|------------------------------------|-------|
| CMNIST       | 0.0%                 | 70.7%                              | 73.3% |
| Waterbirds   | 63.7%                | 79.8%                              | 89.2% |
| CelebA       | 47.8%                | 86.7%                              | 89.3% |
| CivilComments| 56.0%                | 71.1%                              | 72.6% |
## Performance – Domain Shift

| Dataset    | ERM  | Best prior domain invariance method | LISA  |
|------------|------|-------------------------------------|-------|
| Camelyon17 | 70.3%| 74.7%                               | 77.1% |
| FMoW       | 32.3%| 34.6%                               | 35.5% |
| RxRx1      | 29.9%| 28.4%                               | 31.9% |
| Amazon     | 53.8%| 53.8%                               | 54.7% |
| MetaShift  | 52.1%| 52.3%                               | 54.2% |
Analysis

Analysis I: Are the performance gains of LISA from data augmentation?

|                   | Vanilla mixup | LISA   |
|-------------------|---------------|--------|
| Averaged performance over all datasets | 60.9%         | 64.2%  |

Analysis II: Does LISA lead to more invariant predictors?

|                   | Vanilla mixup | LISA   |
|-------------------|---------------|--------|
| Best invariant learning Accuracy of domain prediction | 68.1%         | 64.9%  |
Takeaways

• LISA eliminates spurious correlations between domain & label via **selective augmentation**

• Essentially, LISA improves out-of-distribution robustness by learning more domain-invariant predictors

Code: [https://github.com/huaxiuyao/LISA](https://github.com/huaxiuyao/LISA)