Improving Out-of-Distribution Robustness via Selective Augmentation

Huaxiu Yao\textsuperscript{1}, Yu Wang\textsuperscript{2}, Sai Li\textsuperscript{3}, Linjun Zhang\textsuperscript{4}
Weixin Liang\textsuperscript{1}, James Zou\textsuperscript{1}, Chelsea Finn\textsuperscript{1}

\textsuperscript{1}Stanford University, \textsuperscript{2}University of California San Diego \textsuperscript{3}Renmin University of China, \textsuperscript{4}Rutgers 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

$y_1$: digit < 5

$y_2$: digit $\geq$ 5

40% of train data

10% of train data

10% of train data

40% of train data

Spurious Correlation: color

Prediction: digit < 5

True: digit $\geq$ 5
Why ML Models Fail – Spurious Correlation

$y_1$: digit < 5

40% of train data

$y_2$: digit ≥ 5

10% 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

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

Standard empirical risk minimization (ERM)

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

loss

average over training examples

Prior approaches to learn invariant representations/predictors

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

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

Camelyon17

Standard ERM

70.3% → 74.7%

Best prior domain invariance method

RxRx1

29.9% → 28.4%

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

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 ($d_i \neq d_j, y_i = y_j$)

Colored MNIST

Different background, same label
LISA: Learning Invariant Predictors with Selective Augmentation

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-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

<table>
<thead>
<tr>
<th></th>
<th>ERM</th>
<th>Best prior domain invariance method</th>
<th>LISA</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMNIST</td>
<td>0.0%</td>
<td>70.7%</td>
<td>73.3%</td>
</tr>
<tr>
<td>Waterbirds</td>
<td>63.7%</td>
<td>79.8%</td>
<td>89.2%</td>
</tr>
<tr>
<td>CelebA</td>
<td>47.8%</td>
<td>86.7%</td>
<td>89.3%</td>
</tr>
<tr>
<td>CivilComments</td>
<td>56.0%</td>
<td>71.1%</td>
<td>72.6%</td>
</tr>
</tbody>
</table>
## Performance – Domain Shift

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ERM</th>
<th>Best prior domain invariance method</th>
<th>LISA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camelyon17</td>
<td>70.3%</td>
<td>74.7%</td>
<td>77.1%</td>
</tr>
<tr>
<td>FMoW</td>
<td>32.3%</td>
<td>34.6%</td>
<td>35.5%</td>
</tr>
<tr>
<td>RxRx1</td>
<td>29.9%</td>
<td>28.4%</td>
<td>31.9%</td>
</tr>
<tr>
<td>Amazon</td>
<td>53.8%</td>
<td>53.8%</td>
<td>54.7%</td>
</tr>
<tr>
<td>MetaShift</td>
<td>52.1%</td>
<td>52.3%</td>
<td>54.2%</td>
</tr>
</tbody>
</table>
Analysis

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Averaged performance over all datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vanilla mixup</td>
<td>60.9%</td>
</tr>
<tr>
<td>LISA</td>
<td>64.2% (\uparrow)</td>
</tr>
</tbody>
</table>

Analysis II: Does LISA lead to more invariant predictors?

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy of domain prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best invariant learning</td>
<td>68.1%</td>
</tr>
<tr>
<td>LISA</td>
<td>64.9% (\downarrow)</td>
</tr>
</tbody>
</table>
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)

Thanks

Q & A