Improving Out-of-Distribution Robustness via Selective Augmentation

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

Abstract

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. In this paper, we specifically consider the problems of subpopulation shifts (e.g., imbalanced data) and domain shifts. While prior works often seek to explicitly regularize internal representations or predictors of the model to be domain invariant, we instead aim to learn invariant predictors without restricting the model’s internal representations or predictors. This leads to a simple mixup-based technique which learns invariant predictors via selective augmentation called LISA. LISA selectively interpolates samples either with the same labels but different domains or with the same domain but different labels. Empirically, we study the effectiveness of LISA on nine benchmarks ranging from subpopulation shifts to domain shifts, and we find that LISA consistently outperforms other state-of-the-art methods and leads to more invariant predictors. We further analyze a linear setting and theoretically show how LISA leads to a smaller worst-group error. Code is released in https://github.com/huaxiu1o/LISA

1. Introduction

To deploy machine learning algorithms in real-world applications, we must pay attention to distribution shift, i.e. when the test distribution is different from the training distribution, which substantially degrades model performance. In this paper, we refer this problem as out-of-distribution (OOD) generalization and specifically consider performance gaps caused by two kinds of distribution shifts: subpopulation shifts and domain shifts. 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. For example, in health risk prediction, a machine learning model trained on the entire population may associate the labels with demographic features (e.g., gender and age), making the model fail on the test set when such an association does not hold in reality. 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. In the health risk example, we may want to train a model on patients from a few sampled hospitals and then deploy the model to a broader set of hospitals (Koh et al., 2021).

To improve model robustness under these two kinds of distribution shifts, prior works have proposed various regularizers to learn representations or predictors that are invariant to different domains while still containing sufficient information to fulfill the task (Li et al., 2018; Sun & Saenko, 2016; Arjovsky et al., 2019; Krueger et al., 2021; Rosenfeld et al., 2021). However, designing regularizers that are widely suitable to datasets from diverse domains is challenging, and unsuitable regularizers may adversely limit the model’s expressive power or yield a difficult optimization problem, leading to inconsistent performance among various real-world datasets. For example, on the WILDS datasets, invariant risk minimization (IRM) (Arjovsky et al., 2019) with reweighting – a representative method for learning invariant predictor – outperforms empirical risk minimization (ERM) on CivilComments, but fails to improve robustness on a variety of other datasets like Camelyon17 and RxRx1 (Koh et al., 2021).

Instead of explicitly imposing regularization, we propose to learn invariant predictors through data interpolation, leading to a simple algorithm called LISA (Learning Invariant Predictors with Selective Augmentation). Concretely, inspired by mixup (Zhang et al., 2018), LISA linearly interpolates the features for a pair of samples and applies the same...
interpolation strategy on the corresponding labels. Critically, the pairs are selectively chosen according to two selective augmentation strategies – intra-label LISA (LISA-L) and intra-domain LISA (LISA-D), which are described below and illustrated on Colored MNIST dataset in Figure 1. Intra-label LISA (Figure 1(b)) interpolates samples with the same label but from different domains, aiming to eliminate domain-related spurious correlations. Intra-domain LISA (Figure 1(c)) interpolates samples with the same domain but different labels, such that the model should learn to ignore the domain information and generate different predicted values as the interpolation ratio changes. In this way, LISA encourages the model to learn domain-invariant predictors without any explicit constraints or regularizers.

The primary contributions of this paper are as follows: (1) We propose a simple yet widely-applicable method for learning domain invariant predictors that is shown to be robust to subpopulation shifts and domain shifts. (2) We conduct broad experiments to evaluate LISA on nine benchmark datasets from diverse domains. In these experiments, we make the following key observations. First, we observe that LISA consistently outperforms seven prior methods to address subpopulation and domain shifts. Second, we find that LISA produces predictors that are consistently more domain invariant than prior approaches. Third, we identify that the performance gains of LISA are from canceling out domain-specific information or spurious correlations and learning invariant predictors, rather than simply involving more data via interpolation. Finally, when the degree of distribution shift increases, LISA achieves more significant performance gains. (3) We provide a theoretical analysis of the phenomena distilled from the empirical studies, where we provably demonstrate that LISA can mitigate spurious correlations and therefore lead to smaller worst-domain error compared with ERM and vanilla mixup. We also note that to the best of our knowledge, our work provides the first theoretical framework of studying how mixup (with or without the selective augmentation strategies) affects misclassification error.

2. Preliminaries

In this paper, we consider the setting where one predicts the label $y \in \mathcal{Y}$ based on the input feature $x \in \mathcal{X}$. Given a parameter space $\Theta$ and a loss function $\ell$, we need to train a model $f_\theta$ under the training distribution $P_{tr}$, where $\theta \in \Theta$. In empirical risk minimization (ERM), the empirical distribution over the training data is $P_{tr}$; ERM optimizes the following objective:

$$\theta^* := \arg \min_{\theta \in \Theta} \mathbb{E}_{(x,y) \sim P_{tr}} [\ell(f_\theta(x), y)].$$

In a traditional machine learning setting, a test set, sampled from a test distribution $P_{ts}$, is used to evaluate the generalization of the trained model $\theta^*$, where the test distribution is assumed to be the same as the training distribution, i.e., $P_{tr} = P_{ts}$. In this paper, we are interested in the setting when distribution shift occurs, i.e., $P_{tr} \neq P_{ts}$.

Specifically, following Muandet et al. (2013); Albuquerque et al. (2019); Koh et al. (2021), 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 \( (X, Y, d) = \{(x_i, y_i, d_i)\}_{i=1}^{N_d} \), where \( N_d \) is the number of samples in domain \( d \). Then, we formulate the training distribution as the mixture of \( D \) domains, i.e.,
\[
P_{\mathcal{D}} = \sum_{d \in \mathcal{D}} r_d^p P_d,
\]
where \( \{r_d^p\} \) denotes the mixture probabilities in training set. Here, the training domains are defined as \( \mathcal{D}^{tr} = \{d \in \mathcal{D} | r_d^p > 0\} \). Similarly, the test distribution could be represented as \( P_{\mathcal{D}} = \sum_{d \in \mathcal{D}} r_d^t P_d \), where \( \{r_d^t\} \) is the mixture probabilities in test set. The test domains are defined as \( \mathcal{D}^{ts} = \{d \in \mathcal{D} | r_d^t > 0\} \).

In subpopulation shifts, the test set has domains that have been seen in the training set, but with a different proportion of subpopulations, i.e., \( \mathcal{D}^{ts} \subseteq \mathcal{D}^{tr} \) but \( \{r_d^{ts}\} \neq \{r_d^{tr}\} \).

Under this setting, following Sagawa et al. (2020a), we consider group-based spurious correlations, where each group \( g \in \mathcal{G} \) is defined to be associated with a domain \( d \) and a label \( y \), i.e., \( g = (d, y) \). We assume that the domain is spuriously correlated with the label. For example, we illustrate the CMNIST dataset in Figure 1, where the digit color \( d \) (green or red) is spuriously correlated with the label \( y \) \((1, 0)\) or \((0, 1)\).

Based on the group definition, we evaluate the model via the worst group error, i.e.,
\[
\max_g E_{(x,y) \sim g} [(\ell_{0-1}(f_\theta(x), y))] = \ell_{0-1}
\]
where \( \ell_{0-1} \) represents the 0-1 loss.

In domain shifts, we investigate the problem where the test domains are disjoint from the training domains, i.e., \( \mathcal{D}^{tr} \cap \mathcal{D}^{ts} = \emptyset \). In general, we assume the test domains share some common properties with the training domains. For example, in Camelyon17 (Koh et al., 2021), we train the model on some hospitals and test it in a new hospital. We evaluate the worst-domain and/or average performance of the classifier across all test domains.

3. Learning Invariant Predictors with Selective Augmentation

This section presents LISA, a simple way to improve robustness to subpopulation shifts and domain shifts. The key idea behind LISA is to encourage the model to learn invariant predictors by selective data interpolation, which could also alleviates the effects of domain-related spurious correlations. In the following, we first provide a general formulation for data interpolation and detail how to select interpolated samples by introducing two variants of LISA – intra-label and intra-domain LISA. Then, we discuss the applicable situations for each variant and further introduce domain-free intra-label LISA.

3.1. General Formulation of LISA

In LISA, we perform linear interpolation between training samples. Specifically, given samples \( (x_i, y_i, d_i) \) and \( (x_j, y_j, d_j) \) drawn from domains \( d_i \) and \( d_j \), we apply mixup (Zhang et al., 2018), a simple data interpolation strategy, separately on the input features and corresponding labels as:
\[
x_{\text{mix}} = \lambda x_i + (1 - \lambda) x_j, \quad y_{\text{mix}} = \lambda y_i + (1 - \lambda) y_j, \quad (2)
\]
where the interpolation ratio \( \lambda \in [0, 1] \) is sampled from a Beta distribution \( \text{Beta}(\alpha, \beta) \) and \( y_i \) and \( y_j \) are one-hot vectors for classification problem. Notice that the mixup approach in (2) can be replaced by CutMix (Yun et al., 2019), which shows stronger empirical performance in vision-based applications. In text-based applications, we can use Manifold Mixup (Verma et al., 2019), interpolating the representations of a pre-trained model, e.g., the output of BERT (Devlin et al., 2019).

After obtaining the interpolated features and labels, we replace the original features and labels in ERM with the interpolated ones. Then, the optimization process in (1) is reformulated as:
\[
\theta^* := \arg \min_{\theta} \mathbb{E}_{(x,y,d) \sim \mathcal{D}^{tr}} \left[ (\ell_{0-1}(f_\theta(x_{\text{mix}}), y_{\text{mix}})) \right].
\]

Without additional selective augmentation strategies, vanilla mixup will regularize the model and reduce overfitting (Zhang et al., 2021b), allowing it to attain good in-distribution generalization. However, vanilla mixup may not be able to cancel out spurious correlations, causing the model to still fail at attaining good OOD generalization (see empirical comparisons in Section 4.3 and theoretical discussion in Section 5). 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:

Intra-label LISA (LISA-L): Interpolating samples with the same label. Intra-label LISA interpolates samples with the same label but different domains (i.e., \( d_i \neq d_j, y_i = y_j \)). As shown in Figure 1(a), 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. As a result, intra-label LISA should learn domain-invariant predictors for each class and thus achieve better OOD robustness.

Intra-domain LISA (LISA-D): Interpolating samples with the same domain. Supposing domain information is highly spuriously correlated with the label information, intra-domain LISA (Figure 1(b)) applies the interpolation strategy on samples with the same domain but different labels, i.e., \( d_i = d_j, y_i \neq y_j \). Intuitively, even within the same domain, the model is supposed to generate different predicted labels since the interpolation ratio \( \lambda \) is randomly sampled, corresponding to different labels \( y_{\text{mix}} \). This causes the model to make predictions that are less dependent on the domain, again improving OOD robustness.
3.2. Discussion

How to Select Intra-label or Intra-domain LISA? For each batch of data, we randomly perform intra-label or intra-domain LISA during the training process with probability $p_{sel}$ and $1 - p_{sel}$, where $p_{sel}$ is treated as a hyperparameter in our experiments. The choice of $p_{sel}$ depends on the number of domains and the strength of the spurious correlations. Empirically, intra-domain LISA benefits performance when domain information is highly spuriously correlated with the label. Intra-label LISA brings more benefits when there are more domains or when the spurious correlations are not very strong. The pseudocode of LISA is shown in Algorithm 1.

Domain-free Intra-label LISA. Additionally, we empirically find that only interpolating samples with the same labels without considering the domain information (i.e., $y_i = y_j$) leads to the best performance in some situations with natural domain shifts. We name this variant of intra-label LISA as domain-free intra-label LISA. One potential reason is that the spurious correlations between labels and domains are not very strong in these situations under the existing domain partitions. We offer detailed discussion in Sec. 4.2 and Appendix A.3. Under this situation, enriching the interpolation scope by directly interpolating samples within the same class regardless of existing domain information may bring more benefits.

Algorithm 1 Training Procedure of LISA

Require: Training data $D$, step size $\eta$, learning rate $\gamma$, shape parameters $\alpha, \beta$ of Beta distribution

1: while not converge do
2:    Sample $\lambda \sim \text{Beta}(\alpha, \beta)$
3:    Sample minibatch $B_1 \sim D$
4:    Initialize $B_2 \leftarrow \emptyset$
5:    Select strategy $s \sim \text{Bernoulli}(p_{sel})$
6:    if $s$ is True then
7:        for $(x_i, y_i, d_i) \in B_1$ do
8:            Randomly sample $(x_j, y_j, d_j) \sim \{(x, y, d) \in D\}$ which satisfies $(y_i = y_j)$ and $(d_i \neq d_j)$.
9:            Put $(x_j, y_j, d_j)$ into $B_2$.
10:       end for
11:    else
12:        for $(x_i, y_i, d_i) \in B_1$ do
13:            Randomly sample $(x_j, y_j, d_j) \sim \{(x, y, d) \in D\}$ which satisfies $(y_i \neq y_j)$ and $(d_i = d_j)$.
14:            Put $(x_j, y_j, d_j)$ into $B_2$.
15:       end for
16:    end if
17:    Update $\theta$ with data $\lambda B_1 + (1 - \lambda) B_2$ with learning rate $\gamma$.
18: end while

4. Experiments

In this section, we conduct comprehensive experiments to evaluate the effectiveness of LISA. Specifically, we aim to answer the following questions: Q1: Compared to prior methods, can LISA improve robustness to subpopulation shifts and domain shifts (Section 4.1 and Section 4.2)? Q2: Which aspects of LISA are the most important for improving robustness (Section 4.3)? Q3: Does LISA successfully produce more invariant predictors (Section 4.4)? Q4: How does LISA perform with varying degrees of distribution shifts (Section 4.5)?

To answer Q1, we compare to ERM, IRM (Arjovsky et al., 2019), IB-IRM (Ahuja et al., 2021), V-REx (Krueger et al., 2021), CORAL (Li et al., 2018), DRNN (Ganin & Lempiäisky, 2015), GroupDRO (Sagawa et al., 2020a), Domain-Mix (Xu et al., 2020), and Fish (Shi et al., 2021). Upweighting (UW) is particularly suitable for subpopulation shifts, so we also use it for comparison. We adopt the same model architectures for all approaches. The strategy selection probability $p_{sel}$ is selected via cross-validation.

4.1. Evaluating Robustness to Subpopulation Shifts

Evaluation Protocol. In subpopulation shifts, we evaluate the performance on four binary classification datasets, including Colored MNIST (CMNIST), Waterbirds (Sagawa et al., 2020a), CelebA (Liu et al., 2015), and Civilcomments (Borkan et al., 2019). We detail the data descriptions of subpopulation shifts in Appendix A.1.1 and report the detailed data statistics in Table 1, covering domain information, model architecture, and class information. Following Sagawa et al. (2020a), in subpopulation shifts, we use the worst-group accuracy to evaluate the performance of all approaches. In these datasets, the domain information is highly spurious correlated with the label information. For example, as suggested in Figure 1, 80% images in the CMNIST dataset have the same color in each specific class, i.e., green color for label [1, 0] and red color for label [0, 1].

In CMNIST, Waterbirds, and CelebA, we find that $p_{sel} = 0.5$ works well for choosing selective augmentation strategies, while in CivilComments, we set $p_{sel}$ as 1.0. This is not surprising because it might be more beneficial to use intra-label LISA more often to eliminate domain effects when there are more domains, i.e., eight domains in CivilComments vs. two domains in others. The rest hyperparameter settings and training details are listed in Appendix A.1.2.

Results. In Table 2, we report the overall performance of LISA and other methods. According to Table 2, we observe that the performance of approaches that learn invariant predictors with explicit regularizers (e.g., IRM, IB-IRM, V-REx) is not consistent across datasets. For example, IRM and V-REx outperform UW on CMNIST, but they fail to achieve better performance than UW on Waterbirds. The results corroborate our hypothesis that designing widely effective regularizers is challenging, and that inappropriate regularizers may even hurt the performance. LISA instead consistently outperforms other invariant learning methods
Table 1. Dataset Statistics for Subpopulation Shifts. All datasets are binary classification tasks and we use the worst group accuracy as the evaluation metric.

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

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.

<table>
<thead>
<tr>
<th></th>
<th>CMNIST</th>
<th>Waterbirds</th>
<th>CelebA</th>
<th>CivilComments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>Worst</td>
<td>Avg.</td>
<td>Worst</td>
</tr>
<tr>
<td>ERM</td>
<td>27.8%</td>
<td>0.0%</td>
<td>97.0%</td>
<td>63.7%</td>
</tr>
<tr>
<td>UW</td>
<td>72.2%</td>
<td>66.0%</td>
<td>95.1%</td>
<td>88.0%</td>
</tr>
<tr>
<td>IB-IRM</td>
<td>72.1%</td>
<td>70.3%</td>
<td>87.5%</td>
<td>75.6%</td>
</tr>
<tr>
<td>V-REx</td>
<td>71.7%</td>
<td>70.2%</td>
<td>88.0%</td>
<td>73.6%</td>
</tr>
<tr>
<td>CORAL</td>
<td>71.8%</td>
<td>69.5%</td>
<td>90.3%</td>
<td>79.8%</td>
</tr>
<tr>
<td>GroupDRO</td>
<td>72.3%</td>
<td>68.6%</td>
<td>91.8%</td>
<td><strong>90.6%</strong></td>
</tr>
<tr>
<td>DomainMix</td>
<td>51.4%</td>
<td>48.0%</td>
<td>76.4%</td>
<td>53.0%</td>
</tr>
<tr>
<td>Fish</td>
<td>46.9%</td>
<td>35.6%</td>
<td>85.6%</td>
<td>64.0%</td>
</tr>
<tr>
<td>LISA (ours)</td>
<td><strong>74.0%</strong></td>
<td><strong>73.3%</strong></td>
<td><strong>91.8%</strong></td>
<td><strong>89.2%</strong></td>
</tr>
</tbody>
</table>

Effects of Intra-label and Intra-domain LISA. For CMNIST, Waterbirds and CelebA, both intra-label and intra-domain LISA are used (i.e., \( p_{sel} = 0.5 \)), we illustrate the separate results in Figure 2 and observe that both variants contribute to the final performance.

4.2. Evaluating Robustness to Domain Shifts

Experimental Setup. In domain shifts, we study five datasets. Four of them (Camelyon17, FMoW, RxRx1, and Amazon) are selected from WILDS (Koh et al., 2021), covering natural distribution shifts across diverse domains (e.g., health, language, and vision). Besides the WILDS data, we also apply LISA on the MetaShift datasets (Liang & Zou, 2021), constructed using the real-world images and natural heterogeneity of Visual Genome (Krishna et al., 2016). We summarize these datasets in Table 4, including domain information, evaluation metric, model architecture, and the number of classes. Detailed dataset descriptions and other training details are discussed in Appendix A.2.1 and A.2.2, respectively.

Discussion of Domain-free Intra-Label LISA. The strategy selection probability \( p_{sel} \) is set as 1.0 for these domain shifts datasets, i.e., only intra-label LISA is used. Additionally, as discussed in Sec. 3.2, we adopt domain-free intra-label LISA in Camelyon17, FMoW, and RxRx1, which empirically leads to the best performance. This is consistent with our hypothesis as the spurious correlations between labels and domains are not strong under the existing domain partitions. To further verify our hypothesis and evaluate the strength of spurious correlation, we adopt Cramér’s \( V \) (Cramér, 2016) defined in Appendix A.3 to measure the association between the domain set \( D \) and the label set \( Y \), where the results are reported in Table 13 of Appendix A.3. The Cramér’s \( V \) values in Camelyon17, FMoW, and RxRx1 are significantly smaller than other datasets, indicating relatively weak spurious correlations.

Results. We report the results of domain shifts in Table 3, where full results that include validation performance and other metrics are listed in Appendix A.6. Aligning with the observation in subpopulation shifts, the performance of prior regularization-based invariant predictor learning...
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.

<table>
<thead>
<tr>
<th></th>
<th>Camelyon17</th>
<th>FMoW</th>
<th>RxRx1</th>
<th>Amazon</th>
<th>MetaShift</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Acc.</td>
<td>70.3 ± 6.4%</td>
<td>72.2 ± 3.7%</td>
<td>70.3 ± 5.1%</td>
<td>62.1 ± 5.4%</td>
<td>68.1 ± 4.9%</td>
</tr>
<tr>
<td>Worst Acc.</td>
<td>32.3 ± 1.25%</td>
<td>29.0 ± 1.37%</td>
<td>32.0 ± 1.1%</td>
<td>28.4 ± 0.9%</td>
<td>32.4 ± 0.7%</td>
</tr>
<tr>
<td>Avg. Acc.</td>
<td>29.9 ± 0.4%</td>
<td>9.2 ± 1.1%</td>
<td>35.3 ± 0.8%</td>
<td>4.3 ± 0.8%</td>
<td>32.1 ± 0.5%</td>
</tr>
<tr>
<td>10-th Per. Acc.</td>
<td>53.8 ± 0.8%</td>
<td>32.4 ± 0.8%</td>
<td>32.4 ± 0.8%</td>
<td>32.4 ± 0.8%</td>
<td>32.4 ± 0.8%</td>
</tr>
<tr>
<td>Worst Acc.</td>
<td>52.1 ± 0.4%</td>
<td>32.4 ± 0.8%</td>
<td>32.4 ± 0.8%</td>
<td>32.4 ± 0.8%</td>
<td>32.4 ± 0.8%</td>
</tr>
</tbody>
</table>

Table 4. Dataset Statistics for Domain Shifts.

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

We further analyze the model invariance learned by LISA. From the results, we make the following three key observations. First, compared with Vanilla mixup, the performance of LISA verifies that selective data interpolation indeed improve the out-of-distribution robustness by canceling out the spurious correlations and encouraging learning invariant predictors rather than simple data augmentation. These findings are further strengthened by the results in Table 14 of Appendix A.4, where Vanilla mixup outperforms LISA and ERM without spurious correlations but LISA achieves the best performance with spurious correlations. Second, the superiority of LISA over In-group mixup verifies that only interpolating samples within each group is incapable of eliminating the spurious information, where In-group mixup still performs the role of data augmentation. Third, though incorporating UW significantly improves the performance of Vanilla mixup and In-group mixup in subpopulation shifts, LISA still achieves larger benefits than these enhanced substitute strategies, demonstrating its stronger power in improving OOD robustness.

4.4. Does LISA Lead to More Invariant Predictors?

We further analyze the model invariance learned by LISA. Specifically, for each sample \((x_i, y_i, d)\) in domain \(d\), we denote the unscaled output (i.e., logits) of the model as \(g_{i,d}\). We use two metrics to measure the invariance (see Appendix A.5.1 for additional metrics and the corresponding results):

- **Accuracy of domain prediction (IP_{adp})**. In the first metric, we use the unscaled output to predict the domain. Concretely, the entire dataset is re-split into training, validation, and test sets, where logits are used as features and
We investigate the performance of LISA with respect to Appendix A.2.1). The performance with varied distances is variant predictors, we observe that LISA also leads to more increases of distance. A potential reason is that the effects that LISA achieves more substantial improvements with the methods under all scenarios. Another interesting finding is that LISA consistently outperforms other ERM, IRM, IB-IRM, GroupDRO) are reported for comparison. We observe that LISA consistently outperforms other methods under all scenarios. Another interesting finding is that LISA achieves more substantial improvements with the increases of distance. A potential reason is that the effects of eliminating domain information is more obvious when there is a larger distance between training and test domains.

5. Theoretical Analysis

In this section, we provide some theoretical understandings that explain several of the empirical phenomena from the previous experiments and theoretically compare the worst-group errors of three methods: the proposed LISA, ERM, and vanilla mixup. Specifically, we consider a Gaussian mixture model with subpopulation and domain shifts, which has been widely adopted in theory to shed light upon complex machine learning phenomenon such as in (Montanari et al., 2019; Zhang et al., 2021c; Liu et al., 2021b). We note here that despite the popularity of mixup in practice, the theoretical analysis of how mixup (w/ or w/o the selective augmentation strategies) affects the misclassification error is still largely unexplored in the literature even in the simple models. As discussed in Section 2, here, we define $y \in \{0, 1\}$ as the label, and $d \in \{R, G\}$ as the domain information. For $y \in \{0, 1\}$ and $d \in \{R, G\}$, we consider the following model:

\[ x_i | y_i = y, d_i = d \sim N(\mu^{(y, d)}, \Sigma^{(d)}), i = 1, \ldots, n^{(y, d)}, \]

where $\mu^{(y, d)} \in \mathbb{R}^p$ is the conditional mean vector and $\Sigma^{(d)} \in \mathbb{R}^{p \times p}$ is the covariance matrix. Let $n = \sum_{y \in \{0, 1\}, d \in \{R, G\}} n^{(y, d)}$. Let $\mu^{(y, d)} = \mathbb{E}(y_i = y, d_i = d)$, $\pi^{(y)} = \mathbb{P}(y_i = y)$, and $\pi^{(d)} = \mathbb{P}(d_i = d)$.

To account for the spurious correlation brought by domains, we consider $\mu^{(y, R)} \neq \mu^{(y, G)}$ in general for $y \in \{0, 1\}$ and the imbalanced case where $\pi^{(0, R)}, \pi^{(1, G)} < 1/4$. Moreover, we assume there exists some invariance across different labels represent the corresponding domain ID. A logistic regression model is trained to predict the domain.

• **Pairwise divergence of prediction** ($IP_{kl}$). We calculate the KL divergence of the distribution of logits among all domains, where kernel density estimation is used to estimate the probability density function $P(g_D^y)$ of logits from domain $d$ with label $y$. The pairwise divergence of the predictions is defined as $\frac{1}{|Y||D|} \sum_{y \in Y} \sum_{d, d' \in D} KL(P(g_D^y | D = d) || P(g_{D'}^y | D = d'))$.

Small values of $IP_{adj}$ and $IP_{kl}$ represent strong function-level invariance. In Table 7, we report the results of LISA and other approaches on CMNIST, Waterbirds, Camelyon17 and MetaShift. The results verify that LISA learns predictors with greater domain invariance. Besides having more invariant predictors, we observe that LISA also leads to more invariant representations, as detailed in Appendix A.5.2.

### 4.5. Effect of the Degree of Distribution Shifts

We investigate the performance of LISA with respect to the degree of distribution shifts. Here, we use MetaShift to evaluate performance, where the distance between training and test domains is measured as the node similarity on a meta-graph (Liang & Zou, 2021). To vary the distance between training and test domains, we change the backbones of training objects (see full experimental details in Appendix A.2.1). The performance with varied distances is illustrated in Table 8, where the top four best methods (i.e., ERM, IRM, IB-IRM, GroupDRO) are reported for comparison. We observe that LISA consistently outperforms other methods under all scenarios. Another interesting finding is that LISA achieves more substantial improvements with the increases of distance. A potential reason is that the effects of eliminating domain information is more obvious when there is a larger distance between training and test domains.

### Table 5. Compared LISA with substitute mixup strategies in domain shifts.

<table>
<thead>
<tr>
<th></th>
<th>Camelyon17</th>
<th>FMoW</th>
<th>RxRx1</th>
<th>Amazon</th>
<th>MetaShift</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>70.3 ± 6.4%</td>
<td>32.8 ± 0.45%</td>
<td>29.9 ± 0.4%</td>
<td>53.8 ± 0.8%</td>
<td>52.1 ± 0.4%</td>
</tr>
<tr>
<td>Vanilla mixup</td>
<td>71.2 ± 5.3%</td>
<td>34.2 ± 0.45%</td>
<td>26.5 ± 0.5%</td>
<td>53.3 ± 0.0%</td>
<td>51.3 ± 0.7%</td>
</tr>
<tr>
<td>In-group mixup</td>
<td>75.5 ± 6.7%</td>
<td>32.2 ± 1.18%</td>
<td>24.4 ± 0.2%</td>
<td>53.8 ± 0.6%</td>
<td>52.7 ± 0.5%</td>
</tr>
<tr>
<td><strong>LISA (ours)</strong></td>
<td><strong>77.1 ± 6.5%</strong></td>
<td><strong>35.5 ± 0.65%</strong></td>
<td><strong>31.9 ± 0.8%</strong></td>
<td><strong>54.7 ± 0.0%</strong></td>
<td><strong>54.2 ± 0.7%</strong></td>
</tr>
</tbody>
</table>

### Table 6. Compared LISA with substitute mixup strategies in subpopulation shifts. UW represents upweighting. Full results with standard deviation is listed in Table 11.

<table>
<thead>
<tr>
<th></th>
<th>CMNIST</th>
<th>Waterbirds</th>
<th>CelebA</th>
<th>CivilComments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>Worst</td>
<td>Avg.</td>
<td>Worst</td>
</tr>
<tr>
<td>ERM</td>
<td>27.8%</td>
<td>0.0%</td>
<td>97.0%</td>
<td>63.7%</td>
</tr>
<tr>
<td>Vanilla mixup</td>
<td>32.6%</td>
<td>3.1%</td>
<td>81.0%</td>
<td>56.2%</td>
</tr>
<tr>
<td>Vanilla mixup + UW</td>
<td>72.2%</td>
<td>71.8%</td>
<td>92.1%</td>
<td>85.6%</td>
</tr>
<tr>
<td>In-group mixup</td>
<td>33.6%</td>
<td>24.0%</td>
<td>88.7%</td>
<td>68.0%</td>
</tr>
<tr>
<td>In-group mixup + UW</td>
<td>72.6%</td>
<td>71.6%</td>
<td>91.4%</td>
<td>87.1%</td>
</tr>
<tr>
<td><strong>LISA (ours)</strong></td>
<td><strong>74.0%</strong></td>
<td><strong>73.3%</strong></td>
<td><strong>91.8%</strong></td>
<td><strong>89.2%</strong></td>
</tr>
</tbody>
</table>
Table 7. Results of the analysis of learned invariant predictors. Accuracy of domain prediction (IP_{adp}) and pairwise divergence of prediction among all domains (IP_{kl}) are used to measure the invariance. Smaller values denote stronger invariance.

<table>
<thead>
<tr>
<th></th>
<th>CMNIST</th>
<th>Waterbirds</th>
<th>Camelyon17</th>
<th>MetaShift</th>
<th>CMNIST</th>
<th>Waterbirds</th>
<th>Camelyon17</th>
<th>MetaShift</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>82.85%</td>
<td>94.99%</td>
<td>49.43%</td>
<td>67.98%</td>
<td>6.286</td>
<td>1.888</td>
<td>1.536</td>
<td>1.205</td>
</tr>
<tr>
<td>Vanilla mixup</td>
<td>92.34%</td>
<td>94.49%</td>
<td>52.79%</td>
<td>67.36%</td>
<td>4.737</td>
<td>2.912</td>
<td>0.790</td>
<td>1.171</td>
</tr>
<tr>
<td>IRM</td>
<td>69.42%</td>
<td>95.12%</td>
<td>47.96%</td>
<td>67.59%</td>
<td>7.555</td>
<td>1.122</td>
<td>0.875</td>
<td>1.148</td>
</tr>
<tr>
<td>IB-IRM</td>
<td>74.72%</td>
<td>94.78%</td>
<td>48.37%</td>
<td>67.39%</td>
<td>1.004</td>
<td>3.563</td>
<td>0.756</td>
<td>1.115</td>
</tr>
<tr>
<td>V-REx</td>
<td>63.58%</td>
<td>93.32%</td>
<td>61.38%</td>
<td>68.38%</td>
<td>3.190</td>
<td>3.791</td>
<td>1.281</td>
<td>1.094</td>
</tr>
<tr>
<td>LISA (ours)</td>
<td>58.42%</td>
<td>90.28%</td>
<td>45.15%</td>
<td>66.01%</td>
<td>0.567</td>
<td>0.134</td>
<td>0.723</td>
<td>1.001</td>
</tr>
</tbody>
</table>

Table 8. Effects of the degree of distribution shifts w.r.t. the performance on the MetaShift benchmark. Distance represents the distribution distance between training and test domains. Best B/L represents best baseline.

<table>
<thead>
<tr>
<th>Distance</th>
<th>0.44</th>
<th>0.71</th>
<th>1.12</th>
<th>1.43</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>80.1%</td>
<td>68.4%</td>
<td>52.1%</td>
<td>33.2%</td>
</tr>
<tr>
<td>IRM</td>
<td>79.5%</td>
<td>67.4%</td>
<td>51.8%</td>
<td>32.0%</td>
</tr>
<tr>
<td>IB-IRM</td>
<td>79.7%</td>
<td>66.9%</td>
<td>52.3%</td>
<td>33.6%</td>
</tr>
<tr>
<td>GroupDRO</td>
<td>77.0%</td>
<td>68.9%</td>
<td>51.9%</td>
<td>34.2%</td>
</tr>
<tr>
<td>LISA (ours)</td>
<td>81.3%</td>
<td>69.7%</td>
<td>54.2%</td>
<td>37.5%</td>
</tr>
<tr>
<td>LISA v.s. Best B/L</td>
<td>+1.5%</td>
<td>+1.2%</td>
<td>+3.6%</td>
<td>+9.6%</td>
</tr>
</tbody>
</table>

domains. Specifically, we assume

\[
\mu^{(1,R)} - \mu^{(0,R)} = \mu^{(1,G)} - \mu^{(0,G)} := \Delta \quad \text{and} \quad \Sigma^{(G)} = \Sigma^{(R)} := \Sigma.
\]

According to Fisher’s linear discriminant analysis (Anderson, 1962; Tony Cai & Zhang, 2019; Cai & Zhang, 2021), the optimal classification rule is linear with slope \(\Sigma^{-1}\Delta\). The assumption above implies that \((\Sigma^{-1}\Delta)^\top x\) is the (unknown) invariant prediction rule for model (4).

Suppose we use some method \(A\) and obtain a linear classifier \(x^\top b + b_0 > 0\) from a training data, we will apply it to a test data and compute the worst-group misclassification error, where the mis-classification error for domain \(d\) and class \(y\) is \(E_A^{(u,d)}(b, b_0) := \mathbb{P}(\mathbb{1}(x^\top b + b_0 > \frac{1}{2}) \neq y | d = d, y_i = y)\), and we denote the worst-group error with the method \(A\) as

\[
E_A^{(wst)}(b, b_0) = \max_{d \in \{R, G\}, \pi \in \{0, 1\}} E_A^{(u,d)}(b_d, b_0),
\]

where \(b_d\) and \(b_0,d\) are the slope and intercept based on the method \(A\). Specifically, \(A = \text{ERM}\) denotes the ERM method (by minimizing the sum of squares loss on the training data altogether), \(A = \text{mix}\) denotes the vanilla mixup method (without any selective augmentation strategy), and \(A = \text{LISA}\) denotes the mixup strategy for LISA. We also denote its finite sample version by \(E_A^{(wst)}\).

Let \(\Delta = \mathbb{E}[x_i | y_i = 1] - \mathbb{E}[x_i | y_i = 0]\) denote the marginal difference and \(\xi = \Delta^{\top} \Sigma^{-1} \Delta\) denote the correlation operator between the domain-specific difference \(\Delta\) and the marginal difference \(\Delta\) with respect to \(\Sigma\). We see that smaller \(\xi\) indicates larger discrepancy between the marginal difference and the domain-specific difference and therefore implies stronger spurious correlation between the domains and labels. We present the following theorem showing that our proposed LISA algorithm outperforms the ERM and vanilla mixup in the subpopulation shifts setting.

**Theorem 1 (Error comparison with subpopulation shifts)**
Consider \(n\) independent samples generated from model (4), \(\pi^{(R)} = \pi^{(1)} = 1/2, \pi^{(0,R)} = \pi^{(0,G)} = \alpha < 1/4,\) max_{\pi, d} \|\mu^{(d)}\|_2 \leq C, and \(\Sigma\) is positive definite. Suppose \((\xi, \alpha)\) satisfies that \(\xi < \min\{\|\Delta\|_2, \|\Delta\|_\Sigma\} - C\alpha\) for some large enough constant \(C\) and \(\mathbb{E}[\lambda_2^2]/\max\{\var{\lambda_2}, 1/4\} \geq \|\Delta\|_2^2 + \|\Delta\|_\Sigma \|\Delta\|_\Sigma\}. Then for any \(p_{sel} \in [0, 1]\),

\[
\mathbb{E}_{\text{LISA}}(\text{wst}) \leq \min\{\mathbb{E}_{\text{ERM}}^{(wst)}, \mathbb{E}_{\text{mix}}^{(wst)}\} + O_P\left(\frac{p \log n}{n} + \frac{p}{\alpha n}\right).
\]

In Theorem 1, \(\lambda_i\) is the random mixup coefficient for the \(i\)-th sample. If \(\lambda_\star = \lambda\) are the same for all the samples in a mini-batch, the results still hold. Theorem 1 implies that when \(\xi\) is small (indicating that the domain has strong spurious correlation with the label) and \(p = o(\alpha n)\), the worst-group classification errors of LISA are asymptotically smaller than that of ERM and vanilla mixup. In fact, our analysis shows that LISA yields a classification rule closer to the invariant classification rules by leveraging the domain information.

In the next theorem, we present the mis-classification error comparisons with domain shifts. That is, consider samples from a new unseen domain:

\[
x_i^{(0, \star)} \sim N(\mu^{(0, \star)}, \Sigma), x_i^{(1, \star)} \sim N(\mu^{(1, \star)}, \Sigma).
\]

Let \(\Delta^\star = 2(\mu^{(0, \star)} - \mathbb{E}[x_i])\), where \(\mathbb{E}[x_i]\) is the mean of the training distribution, and assume \(\mu^{(1, \star)} - \mu^{(0, \star)} = \Delta\). Let \(\xi^\star = \Delta^{\top} \Sigma^{-1} \Delta^\star\) and \(\gamma = \Delta^{\top} \Sigma^{-1} \Delta\), respectively, with respect to \(\Sigma^{-1}\). Let \(E_A^{(wst)} = \max_{\pi \in \{0, 1\}} E_A^{(y,d)}(b_d, b_0)\) and its sample version be \(E_A^{(wst)}\).

**Theorem 2 (Error comparison with domain shifts)**
Suppose \(n\) samples are independently generated from model (4), \(\pi^{(R)} = \pi^{(1)} = 1/2, \pi^{(0,R)} = \pi^{(1,G)} = \alpha < 1/4,\)
max_{α, δ} ∥μ(y, δ)∥_2 ≤ C and Σ is positive definite. Suppose that (ξ, ξ∗, γ) satisfy that 0 ≤ ξ ≤ γ ≤ ξ∗ and ξ < min{γ, ||Δ||_Σ^{-1}}, -C_α for some large enough constant C and E[Δ^2] / max{var(λ), 1/4} ≥ ||Δ||_Σ^2 + ||Δ||_Σ ||Δ||_Σ. Then for any p_α ∈ [0, 1],

\[ \hat{E}_{\text{LISA}}^{(w_{α})} < \min\{\hat{E}_{\text{ERM}}^{(w_{α})}, \hat{E}_{\text{max}}^{(w_{α})}\} + O\left(\frac{p \log n}{n} + \frac{p}{\alpha n}\right). \]

Similar to Theorem 1, this result shows that when domain has strong spurious correlation with the label (corresponding to small ξ), such a spurious correlation leads to the downgraded performance of ERM and vanilla mixup, while our proposed LISA method is able to mitigate such an issue by selective data interpolation. Proofs of Theorem 1 and Theorem 2 are provided in Appendix B.

6. Related Work and Discussion

In this paper, we focus on improving the robustness of machine learning models to subpopulation shifts and domain shifts. Here, we discuss related approaches from the following three categories:

Learning Invariant Representations. Motivated by unsupervised domain adaptation (Ben-David et al., 2010; Ganin et al., 2016), the first category of works learns invariant representations by aligning representations across domains. The major research line of this category aims to eliminate the domain dependency by minimizing the divergence of feature distributions with different distance metrics, e.g., maximum mean discrepancy (Tzeng et al., 2014; Long et al., 2015), an adversarial loss (Ganin et al., 2016; Li et al., 2018), Wassertein distance (Zhou et al., 2020a). Follow-up works applied data augmentation to (1) generate more domains and enhance the consistency of representations during training (Vye et al., 2019; Zhou et al., 2020b; Xu et al., 2020; Yan et al., 2020; Shu et al., 2021; Wang et al., 2020; Yao et al., 2021) or (2) generate new domains in an adversarial way to imitate the challenging domains without using training domain information (Zhao et al., 2020; Qiao et al., 2020; Volpi et al., 2018). Unlike these latter methods, LISA instead focuses on learning invariant predictors without restricting the internal representations, leading to stronger empirical performance.

Learning Invariant Predictors. Beyond using domain alignment to learning invariant representations, recent work aims to further enhance the correlations between the invariant representations and the labels (Koyama & Yamaguchi, 2020), leading to invariant predictors. Representatively, motivated by casual inference, invariant risk minimization (IRM) (Arjovsky et al., 2019) and its variants (Guo et al., 2021; Khezeli et al., 2021; Ahuja et al., 2021) aim to find a predictor that performs well across all domains through regularizations. Other follow-up works leverage regularizers to penalize the variance of risks across all domains (Krueger et al., 2021), to align the gradient across domains (Koyama & Yamaguchi, 2020), to smooth the cross-domain interpolation paths (Chuang & Mroueh, 2021), or to involve game-theoretic invariant rationalization criterion (Chang et al., 2020). Instead of using regularizers, LISA instead learns domain-invariant predictors via data interpolation.

Group Robustness. The last category of methods combating spurious correlations and are particularly suitable for subpopulation shifts. These approaches include directly optimizing the worst-group performance with Distributionally Robust Optimization (Sagawa et al., 2020a; Zhang et al., 2021a; Zhou et al., 2021), generating samples around the minority groups (Goel et al., 2021), and balancing the majority and minority groups via reweighting (Sagawa et al., 2020b) or regularizing (Cao et al., 2019; 2020). A few recent approaches in this category target on subpopulation shifts without annotated group labels (Nam et al., 2020; Liu et al., 2021a; Zhang et al., 2021d; Creager et al., 2021; Lee et al., 2022). LISA proposes a more general strategy that is suitable for both domain shifts and subpopulation shifts.

7. Conclusion

To tackle distribution shifts, we propose LISA, a simple and efficient algorithm, to improve the out-of-distribution robustness. LISA aims to eliminate the domain-related spurious correlations among the training set via selective interpolation. We evaluate the effectiveness of LISA on nine datasets under subpopulation shifts and domain shifts settings, demonstrating its promise. Besides, detailed analyses verify that the performance gains caused by LISA result from encouraging learning invariant predictors and representations. Theoretical results further strengthen the superiority of LISA by showing smaller worst-group mis-classification error compared with ERM and vanilla data interpolation.

Acknowledgement

We thank Pang Wei Koh for the many insightful discussions. This research was funded in part by JPMorgan Chase & Co. Any views or opinions expressed herein are solely those of the authors listed, and may differ from the views and opinions expressed by JPMorgan Chase & Co. or its affiliates. This material is not a product of the Research Department of J.P. Morgan Securities LLC. This material should not be construed as an individual recommendation for any particular client and is not intended as a recommendation of particular securities, financial instruments or strategies for a particular client. This material does not constitute a solicitation or offer in any jurisdiction. The research was also supported by Apple and Juniper Networks. The research of Linjun Zhang is partially supported by NSF DMS-2015378.
References


Improving Out-of-Distribution Robustness via Selective Augmentation


Improving Out-of-Distribution Robustness via Selective Augmentation


A. Additional Experiments

A.1. Additional Experiments on Subpopulation Shifts

A.1.1. Dataset Details

**Colored MNIST (CMNIST):** We classify MNIST digits from 2 classes, where classes 0 and 1 indicate original digits (0,1,2,3,4) and (5,6,7,8,9). The color is treated as a spurious attribute. Concretely, in the training set, the proportion between red samples and green samples is 8:2 in class 0, while the proportion is set as 2:8 in class 1. In the validation set, the proportion between green and red samples is 1:1 for all classes. In the test set, the proportion between green and red samples is 1:9 in class 0, while the ratio is 9:1 in class 1. The data sizes of train, validation, and test sets are 30000, 10000, and 20000, respectively.

**Waterbirds (Sagawa et al., 2020a):** The Waterbirds dataset aims to classify birds as “waterbird” or “landbird”, where each bird image is spuriously associated with the background “water” or “land”. Waterbirds is a synthetic dataset where each image is composed by pasting a bird image sampled from CUB dataset (Wah et al., 2011) to a background drawn from the Places dataset (Zhou et al., 2017). The bird categories in CUB are stratified as land birds or water birds. Specifically, the following bird species are selected to construct the waterbird class: albatross, auklet, cormorant, frigatebird, fulmar, gull, jaeger, kittiwake, pelican, puffin, tern, gadwall, grebe, mallard, merganser, guillemot, or Pacific loon. All other bird species are combined as the landbird class. We define (land background, waterbird) and (water background, landbird) are minority groups. There are 4,795 training samples while only 56 samples are “waterbirds on land” and 184 samples are “landbirds on water”. The remaining training data include 3,498 samples from “landbirds on land”, and 1,057 samples from “waterbirds on water”.

**CelebA (Liu et al., 2015; Sagawa et al., 2020a):** For the CelebA data (Liu et al., 2015), we follow the data preprocess procedure from (Sagawa et al., 2020a). CelebA defines a image classification task where the input is a face image of celebrities and the classification label is its corresponding hair color – “blond” or “not blond.” The label is spuriously correlated with gender, i.e., male or female. In CelebA, the minority groups are (blond, male) and (not blond, female). The number of samples for each group are 71,629 “dark hair, female”, 66,874 “dark hair, male”, 22,880 “blond hair, female”, 1,387 “blond hair, male”.

**CivilComments (Borkan et al., 2019; Koh et al., 2021):** We use CivilComments from the WILDS benchmark (Koh et al., 2021). CivilComments is a text classification task, aiming to predict whether an online comment is toxic or non-toxic. The spurious domain identifications are defined as the demographic features, including male, female, LGBTQ, Christian, Muslim, other religion, Black, and White. CivilComments contains 450,000 comments collected from online articles. The number of samples for training, validation, and test are 269,038, 45,180, and 133,782, respectively. The readers may kindly refer to Table 17 in (Koh et al., 2021) for the detailed group information.

A.1.2. Training Details

We adopt pre-trained ResNet-50 (He et al., 2016) and BERT (Sanh et al., 2019) as the model for image data (i.e., CMNIST, Waterbirds, CelebA) and text data (i.e., CivilComments), respectively. In each training iteration, we sample a batch of data per group. For intra-label LISA, we randomly apply mixup on sample batches with the same labels but different domains. For intra-domain LISA, we instead apply mixup on sample batches with the same domain but different labels. The interpolation ratio $\lambda$ is sampled from the distribution Beta(2, 2). All hyperparameters are selected via cross-validation and are listed in Table 9.

A.1.3. Additional Results

In this section, we have added the full results of subpopulation shifts in Table 10 and Table 11.

A.2. Additional Experimental Settings on Domain Shifts

A.2.1. Dataset Details

In this section, we provide detailed descriptions of datasets used in the experiments of domain shifts and report the data statistics in Table 4.
### Table 9. Hyperparameter settings for the subpopulation shifts.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CMNIST</th>
<th>Waterbirds</th>
<th>CelebA</th>
<th>CivilComments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning rate</td>
<td>1e-3</td>
<td>1e-3</td>
<td>1e-4</td>
<td>1e-5</td>
</tr>
<tr>
<td>Weight decay</td>
<td>1e-4</td>
<td>1e-4</td>
<td>1e-4</td>
<td>0</td>
</tr>
<tr>
<td>Scheduler</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Batch size</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Type of mixup</td>
<td>mixup</td>
<td>mixup</td>
<td>CutMix</td>
<td>ManifoldMix</td>
</tr>
<tr>
<td>Architecture</td>
<td>ResNet50</td>
<td>ResNet50</td>
<td>ResNet50</td>
<td>DistilBert</td>
</tr>
<tr>
<td>Optimizer</td>
<td>SGD</td>
<td>SGD</td>
<td>SGD</td>
<td>Adam</td>
</tr>
<tr>
<td>Maximum Epoch</td>
<td>300</td>
<td>300</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>Strategy sel. prob. $p_{sel}$</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### Table 10. Full results of subpopulation shifts with standard deviation. All the results are performed with three random seed.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CMNIST</th>
<th>Waterbirds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>Worst</td>
</tr>
<tr>
<td>ERM</td>
<td>27.8 ± 1.9%</td>
<td>0.0 ± 0.0%</td>
</tr>
<tr>
<td>UW</td>
<td>72.2 ± 1.1%</td>
<td>66.0 ± 0.7%</td>
</tr>
<tr>
<td>IRM</td>
<td>72.1 ± 1.2%</td>
<td>70.3 ± 0.8%</td>
</tr>
<tr>
<td>IB-IRM</td>
<td>72.2 ± 1.3%</td>
<td>70.7 ± 1.2%</td>
</tr>
<tr>
<td>V-REx</td>
<td>71.7 ± 1.2%</td>
<td>70.2 ± 0.9%</td>
</tr>
<tr>
<td>Coral</td>
<td>71.8 ± 1.7%</td>
<td>69.5 ± 0.9%</td>
</tr>
<tr>
<td>GroupDRO</td>
<td>72.3 ± 1.2%</td>
<td>68.6 ± 0.8%</td>
</tr>
<tr>
<td>DomainMix</td>
<td>51.4 ± 1.3%</td>
<td>48.0 ± 1.3%</td>
</tr>
<tr>
<td>Fish</td>
<td>46.9 ± 1.4%</td>
<td>35.6 ± 1.7%</td>
</tr>
</tbody>
</table>

| LISA (ours)      | 92.4 ± 0.4% | 89.3 ± 1.1% |

Cameylon17 We use Cameylon17 from the WILDS benchmark (Koh et al., 2021; Bandi et al., 2018), which provides 450,000 lymph-node scans sampled from 5 hospitals. Cameylon17 is a medical image classification task where the input $x$ is a $96 \times 96$ image and the label $y$ is whether there exists tumor tissue in the image. The domain $d$ denotes the hospital that the patch was taken from. The training dataset is drawn from the first 3 hospitals, while out-of-distribution validation and out-of-distribution test datasets are sampled from the 4-th hospital and 5-th hospital respectively.

FMoW The FMoW dataset is from the WILDS benchmark (Koh et al., 2021; Christie et al., 2018) — a satellite image classification task which includes 62 classes and 80 domains (16 years x 5 regions). Concretely, the input $x$ is a $224 \times 224$ RGB satellite image, the label $y$ is one of the 62 building or land use categories, and the domain $d$ represents the year that the image was taken as well as its corresponding geographical region – Africa, the Americas, Oceania, Asia, or Europe. The train/test/validation splits are based on the time when the images are taken. Specifically, images taken before 2013 are used as the training set. Images taken between 2013 and 2015 are used as the validation set. Images taken after 2015 are used for testing.

RxRx1 RxRx1 (Koh et al., 2021; Taylor et al., 2019) from the WILDS benchmark is a cell image classification task. In the dataset, some cells have been genetically perturbed by siRNA. The goal of RxRx1 is to predict which siRNA that
Table 11. Full table of the comparison between LISA and other substitute mixup strategies in subpopulation shifts. UW represents upweighting.

<table>
<thead>
<tr>
<th></th>
<th>CMNIST</th>
<th></th>
<th>Waterbirds</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>Worst</td>
<td>Avg.</td>
<td>Worst</td>
</tr>
<tr>
<td>ERM</td>
<td>27.8 ± 1.9%</td>
<td>0.0 ± 0.0%</td>
<td>97.0 ± 0.2%</td>
<td>63.7 ± 1.9%</td>
</tr>
<tr>
<td>Vanilla mixup</td>
<td>32.6 ± 3.1%</td>
<td>3.1 ± 2.4%</td>
<td>81.0 ± 0.2%</td>
<td>56.2 ± 0.2%</td>
</tr>
<tr>
<td>Vanilla mixup + UW</td>
<td>72.2 ± 0.7%</td>
<td>71.8 ± 0.1%</td>
<td>92.1 ± 0.1%</td>
<td>85.6 ± 1.0%</td>
</tr>
<tr>
<td>In-group Group</td>
<td>33.6 ± 1.9%</td>
<td>24.0 ± 1.1%</td>
<td>88.7 ± 0.3%</td>
<td>68.0 ± 0.4%</td>
</tr>
<tr>
<td>In-group + UW</td>
<td>72.6 ± 0.1%</td>
<td>71.6 ± 0.2%</td>
<td>91.4 ± 0.6%</td>
<td>87.1 ± 0.6%</td>
</tr>
<tr>
<td>LISA (ours)</td>
<td>74.0 ± 0.1%</td>
<td><strong>73.3 ± 0.2%</strong></td>
<td>91.8 ± 0.3%</td>
<td><strong>89.2 ± 0.6%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CelebA</th>
<th></th>
<th>CivilComments</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>Worst</td>
<td>Avg.</td>
<td>Worst</td>
</tr>
<tr>
<td>ERM</td>
<td>94.9 ± 0.2%</td>
<td>47.8 ± 3.7%</td>
<td>92.2 ± 0.1%</td>
<td>56.0 ± 3.6%</td>
</tr>
<tr>
<td>Vanilla mixup</td>
<td>95.8 ± 0.0%</td>
<td>46.4 ± 0.5%</td>
<td>90.8 ± 0.8%</td>
<td>67.2 ± 1.2%</td>
</tr>
<tr>
<td>Vanilla mixup + UW</td>
<td>91.5 ± 0.2%</td>
<td>88.0 ± 0.3%</td>
<td>87.8 ± 1.2%</td>
<td>66.1 ± 1.4%</td>
</tr>
<tr>
<td>Within Group</td>
<td>95.2 ± 0.3%</td>
<td>58.3 ± 0.9%</td>
<td>90.8 ± 0.6%</td>
<td>69.2 ± 0.8%</td>
</tr>
<tr>
<td>Within Group + UW</td>
<td>92.4 ± 0.4%</td>
<td>87.8 ± 0.6%</td>
<td>84.8 ± 0.7%</td>
<td>69.3 ± 1.1%</td>
</tr>
<tr>
<td>LISA (ours)</td>
<td>92.4 ± 0.4%</td>
<td><strong>89.3 ± 1.1%</strong></td>
<td>89.2 ± 0.9%</td>
<td><strong>72.6 ± 0.1%</strong></td>
</tr>
</tbody>
</table>

the cells have been treated with. Concretely, the input \(x\) is an image of cells obtained by fluorescent microscopy, the label \(y\) indicates which of the 1,139 genetic treatments the cells received, and the domain \(d\) denotes the experimental batches. Here, 33 different batches of images are used for training, where each batch contains one sample for each class. The out-of-distribution validation set has images from 4 experimental batches. The out-of-distribution test set has 14 experimental batches. The average accuracy on out-of-distribution test set is reported.

**Amazon** Each task in the Amazon benchmark (Koh et al., 2021; Ni et al., 2019) is a multi-class sentiment classification task. The input \(x\) is the text of a review, the label \(y\) is the corresponding star rating ranging from 1 to 5, and the domain \(d\) is the corresponding reviewer. The training set has 245, 502 reviews from 1, 252 reviewers, while the out-of-distribution validation set has 100, 050 reviews from another 1, 334 reviewers. The out-of-distribution test set also has 100, 050 reviews from the rest 1, 252 reviewers. We evaluate the models by the 10th percentile of per-user accuracies in the test set.

**MetaShift** We use the MetaShift (Liang & Zou, 2021), which is derived from Visual Genome (Krishna et al., 2016). MetaShift leverages the natural heterogeneity of Visual Genome to provide many distinct data distributions for a given class (e.g. “cats with cars” or “cats in bathroom” for the “cat” class). A key feature of MetaShift is that it provides explicit explanations of the dataset correlation and a distance score to measure the degree of distribution shift between any pair of sets.

We adopt the “Cat vs. Dog” task in MetaShift, where we evaluate the model on the “dog(\textit{shelf})” domain with 306 images, and the “cat(\textit{shelf})” domain with 235 images. The training data for the “Cat” class is the cat(\textit{sofa + bed}), including cat(\textit{sofa}) domain and cat(\textit{bed}) domain. MetaShift provides 4 different sets of training data for the “Dog” class in an increasingly challenging order, i.e., increasing the amount of distribution shift. Specifically, dog(\textit{cabinet + bed}), dog(\textit{bag + box}), dog(\textit{bench + bike}), dog(\textit{boat + surfboard}) are selected for training, where their corresponding distances to dog(\textit{shelf}) are 0.44, 0.71, 1.12, 1.43.

### A.2.2. Training Details

Follow WILDS Koh et al. (2021), we adopt pre-trained DenseNet121 (Huang et al., 2017) for Camelyon17 and FMoW datasets, ResNet-50 (He et al., 2016) for RXRx1 and MetaShift datasets, and DistilBert (Sanh et al., 2019) for Amazon datasets.

In each training iteration, we first draw a batch of samples \(B_1\) from the training set. With \(B_1\), we then select another sample batch \(B_2\) with same labels as \(B_1\) for data interpolation. The interpolation ratio \(\lambda\) is drawn from the distribution \(\text{Beta}(2, 2)\). We use the same image transformers as Koh et al. (2021), and all other hyperparameters are selected via cross-validation and are listed in Table 12.
### Table 12. Hyperparameter settings for the domain shifts.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Camelyon17</th>
<th>FMoW</th>
<th>RxRx1</th>
<th>Amazon</th>
<th>MetaShift</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning rate</td>
<td>1e-4</td>
<td>1e-4</td>
<td>1e-3</td>
<td>2e-6</td>
<td>1e-3</td>
</tr>
<tr>
<td>Weight decay</td>
<td>0</td>
<td>0</td>
<td>1e-5</td>
<td>0</td>
<td>1e-4</td>
</tr>
<tr>
<td>Scheduler</td>
<td>n/a</td>
<td>n/a</td>
<td>Cosine Warmup</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>Batch size</td>
<td>32</td>
<td>32</td>
<td>72</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Type of mixup</td>
<td>CutMix</td>
<td>CutMix</td>
<td>CutMix</td>
<td>ManifoldMix</td>
<td>CutMix</td>
</tr>
<tr>
<td>Architecture</td>
<td>DenseNet121</td>
<td>DenseNet121</td>
<td>ResNet50</td>
<td>DistilBert</td>
<td>ResNet50</td>
</tr>
<tr>
<td>Optimizer</td>
<td>SGD</td>
<td>Adam</td>
<td>Adam</td>
<td>Adam</td>
<td>SGD</td>
</tr>
<tr>
<td>Maximum Epoch</td>
<td>2</td>
<td>5</td>
<td>90</td>
<td>3</td>
<td>100</td>
</tr>
<tr>
<td>Strategy sel. prob. $p_{sel}$</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### A.3. Strength of Spurious Correlation and Domain-free Intra-label LISA

In Section 2, the spurious correlation is defined as the association between the domain $d$ and label $y$, measured by Cramér’s $V$ (Cramér, 2016). Specifically, let $k_{y,d}$ be the number of samples from domain $d$ with label $y$. The Cramér’s $V$ is formulated as

$$V = \sqrt{\frac{\chi^2}{N \min(|Y-1|, |D-1|)}} = \sqrt{\frac{\sum_{y \in Y, d \in D} (k_{y,d} - \bar{k}_{y,d})^2}{N \min(|Y| - 1, |D| - 1)},}$$  \hspace{1cm} (6)

where $N$ represents the number of samples in the entire dataset and $\bar{k}_{y,d} = \frac{\sum_{y \in Y} k_{y,d} \sum_{d \in D} k_{y,d}}{\sum_{y \in Y} \sum_{d \in D} k_{y,d}}$. Cramér’s $V$ varies from 0 to 1 and higher Cramér’s $V$ represents stronger correlation.

According to Eq. (6), we calculate the strength of spurious correlations on all datasets used in the experiments and report the results in Table 13. Compared with other datasets, the Cramér’s $V$ on Camelyon17, FMoW and RxRx1 are significantly smaller, indicating weaker spurious correlations and may explaining efficacy of domain-free intra-label LISA.

<table>
<thead>
<tr>
<th>Subpopulation Shifts</th>
<th>CMNIST</th>
<th>Waterbirds</th>
<th>CelebA</th>
<th>CivilComments</th>
<th>Domain Shifts</th>
<th>FMoW</th>
<th>RxRx1</th>
<th>Amazon</th>
<th>MetaShift</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6000</td>
<td>0.8672</td>
<td>0.3073</td>
<td>0.8723</td>
<td>0.0004</td>
<td>0.1114</td>
<td>0.0067</td>
<td>0.3377</td>
<td>0.4932</td>
</tr>
</tbody>
</table>

### A.4. Results on Datasets without Spurious Correlations

In order to analyze the factors that lead to the performance gains of LISA, we conduct experiments on datasets without spurious correlations. To be more specific, we balance the number of samples for each group under the subpopulation shifts setting. The results of ERM, Vanilla mixup and LISA on CMNIST, Waterbirds and CelebA are reported in Table 14. The results show that LISA performs similarly compared with ERM when datasets do not have spurious correlations. If there exists any spurious correlation, LISA significantly outperforms ERM. Another interesting finding is that Vanilla mixup outperforms LISA and ERM without spurious correlations, while LISA achieves the best performance with spurious correlations. This finding strengthens our conclusion that the performance gains of LISA are from eliminating spurious correlations rather than simple data augmentation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CMNIST</th>
<th>Waterbirds</th>
<th>CelebA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>73.67%</td>
<td>88.07%</td>
<td>86.11%</td>
</tr>
<tr>
<td>Vanilla mixup</td>
<td>74.28%</td>
<td>88.23%</td>
<td>88.89%</td>
</tr>
<tr>
<td>LISA</td>
<td>73.18%</td>
<td>87.05%</td>
<td>87.22%</td>
</tr>
</tbody>
</table>
A.5. Additional Invariance Analysis

### A.5.1. Additional Metrics of Invariant Predictor Analysis

In Table 15, we report two additional metrics to measure the invariance of predictors – Risk Variance and Gradient Norm, which is defined as:

- **Risk Variance** ($IP_{\text{var}}$). Motivated by Krueger et al. (2021), we use the variance of test risks across all domains to measure the invariance, which is defined as $IP_{\text{var}} = \text{Var}(\{R_1(\theta), \ldots, R_D(\theta)\})$, where $D$ represents the number of test domains and $R_d(\theta)$ represents the risk of domain $d$.

- **Gradient Norm** ($IP_{\text{norm}}$). Follow IRMv1 (Arjovsky et al., 2019), we use the gradient norm of the classifier to measure the optimality of the dummy classifier at each domain $d$. Assume the classifier is parameterized by $w$, $IP_{\text{norm}}$ is defined as: $IP_{\text{norm}} = \frac{1}{|D|} \sum_{d \in D} \|\nabla_{w|w=1.0} R_d(\theta)\|^2$.

#### Table 15. Additional Invariance Metrics for Invariant Predictor Analysis. We report the results of risk variance ($IP_{\text{var}}$) and gradient norm ($IP_{\text{norm}}$), where smaller values indicate stronger invariance.

<table>
<thead>
<tr>
<th></th>
<th>CMNIST</th>
<th>Waterbirds</th>
<th>Camelyon17</th>
<th>MetaShift</th>
<th>CMNIST</th>
<th>Waterbirds</th>
<th>Camelyon17</th>
<th>MetaShift</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>12.0486</td>
<td>0.2456</td>
<td>0.0150</td>
<td>1.8824</td>
<td>1.1162</td>
<td>1.5780</td>
<td>1.2959</td>
<td>1.0914</td>
</tr>
<tr>
<td>Vanilla mixup</td>
<td>0.2769</td>
<td>0.1465</td>
<td>0.0180</td>
<td>0.2659</td>
<td>1.5347</td>
<td>1.8631</td>
<td>0.3993</td>
<td>0.1985</td>
</tr>
<tr>
<td>IRM</td>
<td>0.0112</td>
<td>0.1243</td>
<td>0.0201</td>
<td>0.8748</td>
<td>0.0908</td>
<td>0.9798</td>
<td>0.5266</td>
<td>0.2320</td>
</tr>
<tr>
<td>IB-IRM</td>
<td>0.0072</td>
<td>0.2069</td>
<td>0.0329</td>
<td>0.5680</td>
<td>0.6225</td>
<td>0.8814</td>
<td>0.6890</td>
<td>0.1683</td>
</tr>
<tr>
<td>V-REx</td>
<td>0.0056</td>
<td>0.1257</td>
<td>0.0106</td>
<td>0.4220</td>
<td>0.0290</td>
<td>0.8329</td>
<td>0.9641</td>
<td>0.3680</td>
</tr>
<tr>
<td><strong>LISA (ours)</strong></td>
<td><strong>0.0012</strong></td>
<td><strong>0.0016</strong></td>
<td><strong>9.97e-5</strong></td>
<td><strong>0.2387</strong></td>
<td><strong>0.0039</strong></td>
<td><strong>0.0538</strong></td>
<td><strong>0.3081</strong></td>
<td><strong>0.1354</strong></td>
</tr>
</tbody>
</table>

Comparing LISA to other invariant learning methods, the results of $IP_{\text{var}}$ and $IP_{\text{norm}}$ further confirm that LISA does indeed improve predictor invariance.

### A.5.2. Analysis of Learned Invariant Representations

In this section, we use pairwise divergence of representations ($IR_{kl}$) to measure representation-level invariance. Specifically, assume the representation before classifier of each sample $(x_i, y_i, d)$ is $h_{i,d}$, we compute the KL divergence of the distribution of representations. Similarly, kernel density estimation is also used to estimate the probability density function $P(h_y^D)$ of representations from domain $d$ with label $y$. Formally, $IR_{kl}$ is defined as $IR_{kl} = \frac{1}{|Y||D|} \sum_{y \in Y} \sum_{d',d \in D} \text{KL}(P(h_y^d | D = d) | P(h_y^{d'}) | D = d'))$. Smaller $IR_{kl}$ values indicate more invariant representations with respect to the labels. We report the results on CMNIST, Waterbirds, Camelyon17 and MetaShift in Table 16. Our key observations are: (1) Compared with ERM, LISA learns stronger representation-level invariance. The potential reason is that a stronger invariant predictor implicitly includes stronger invariance representation; (2) LISA provides more invariant representations than other regularization-based invariant predictor learning methods, i.e., IRM, IB-IRM, V-REx, showing its capability in learning stronger invariance.

#### Table 16. Results of representation-level invariance $IR_{kl}$ ($\times 10^8$ for CMNIST), where smaller $IR_{kl}$ value denotes stronger invariance.

<table>
<thead>
<tr>
<th></th>
<th>CMNIST</th>
<th>Waterbirds</th>
<th>Camelyon17</th>
<th>MetaShift</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>1.683</td>
<td>3.592</td>
<td>8.213</td>
<td>0.632</td>
</tr>
<tr>
<td>Vanilla mixup</td>
<td>4.392</td>
<td>3.935</td>
<td>7.786</td>
<td>0.634</td>
</tr>
<tr>
<td>IRM</td>
<td>1.905</td>
<td>2.413</td>
<td>8.169</td>
<td>0.627</td>
</tr>
<tr>
<td>IB-IRM</td>
<td>3.178</td>
<td>3.306</td>
<td>8.824</td>
<td>0.646</td>
</tr>
<tr>
<td>V-REx</td>
<td>3.169</td>
<td>3.414</td>
<td>8.838</td>
<td>0.617</td>
</tr>
<tr>
<td><strong>LISA (ours)</strong></td>
<td><strong>0.421</strong></td>
<td><strong>1.912</strong></td>
<td><strong>7.570</strong></td>
<td><strong>0.585</strong></td>
</tr>
</tbody>
</table>

Besides the quantitative analysis, follow Appendix C in Lee et al. (2019), we visualize the hidden representations for all test samples and the decision boundary on Waterbirds and illustrate the results in Figure 3. Compared with other methods, the representations of samples with the same label that learned by LISA are closer regardless of their domain information, which further demonstrates the promise of LISA in producing invariant representations.
Improving Out-of-Distribution Robustness via Selective Augmentation

![Figure 3. Visualization of sample representations and decision boundaries on Waterbirds dataset.](image)

**A.6. Full Results of WILDS data**

Follow Koh et al. (2021), we reported more results on WILDS datasets in Table 17 - Table 20, including validation performance and the results of other metrics. According to these additional results, we could see that LISA outperforms other baseline approaches in all scenarios. Particularly, we here discuss two additional findings: (1) In Camelyon dataset, the test data is much more visually distinctive compared with the validation data, resulting in the large gap (∼10%) between validation and test performance of ERM (see Table 17). However, LISA significantly reduces the performance gap between the validation and test sets, showing its promise in improving OOD robustness; (2) In Amazon dataset, though LISA performs worse than ERM in average accuracy, it achieves the best accuracy at the 10th percentile, which is regarded as a more common and important metric to evaluate whether models perform consistently well across all users (Koh et al., 2021).

**Table 17. Full Results of Camelyon17.** We report both validation accuracy and test accuracy.

<table>
<thead>
<tr>
<th>Method</th>
<th>Validation Acc.</th>
<th>Test Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>84.9 ± 3.1%</td>
<td>70.3 ± 6.4%</td>
</tr>
<tr>
<td>IRM</td>
<td><strong>86.2 ± 1.4%</strong></td>
<td>64.2 ± 8.1%</td>
</tr>
<tr>
<td>IB-IRM</td>
<td>80.5 ± 0.4%</td>
<td>68.9 ± 6.1%</td>
</tr>
<tr>
<td>V-REx</td>
<td>82.3 ± 1.3%</td>
<td>71.5 ± 8.3%</td>
</tr>
<tr>
<td>Coral</td>
<td><strong>86.2 ± 1.4%</strong></td>
<td>59.5 ± 7.7%</td>
</tr>
<tr>
<td>GroupDRO</td>
<td>85.5 ± 2.2%</td>
<td>68.4 ± 7.3%</td>
</tr>
<tr>
<td>DomainMix</td>
<td>83.5 ± 1.1%</td>
<td>69.7 ± 5.5%</td>
</tr>
<tr>
<td>Fish</td>
<td>83.9 ± 1.2%</td>
<td>74.7 ± 7.1%</td>
</tr>
<tr>
<td>LISA (ours)</td>
<td>81.8 ± 1.3%</td>
<td><strong>77.1 ± 6.5%</strong></td>
</tr>
</tbody>
</table>

**B. Proofs of Theorem 1 and Theorem 2**

**Outline of the proof.** We will first find the mis-classification errors based on the population version of OLS with different mixup strategies. Next, we will develop the convergence rate of the empirical OLS based on $n$ samples towards its population version. These two steps together give us the empirical mis-classification errors of different methods. We will separately show that the upper bounds in Theorem 1 and Theorem 2 hold for two selective augmentation strategies of LISA and hence hold for any $p_{sel} \in [0, 1]$. Let LL denote intra-label LISA and LD denote intra-domain LISA.

Let $\pi_1 = P(y_i = 1)$ and $\pi_0 = P(y_i = 0)$ denote the marginal class proportions in the training samples. Let $\pi_R = P(d_i = R)$ and $\pi_G = P(d_i = G)$ denote the marginal subpopulation proportions in the training samples. Let $\pi_G|1 = P(d_i = G|y_i = 1)$ and define $\pi_G|0$, $\pi_R|1$, and $\pi_R|0$ similarly.
Table 18. Full Results of FMoW. Here, we report the average accuracy and the worst-domain accuracy on both validation and test sets.

<table>
<thead>
<tr>
<th></th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>59.5 ± 0.37%</td>
<td>48.9 ± 0.62%</td>
</tr>
<tr>
<td>IRM</td>
<td>57.4 ± 0.37%</td>
<td>47.5 ± 1.57%</td>
</tr>
<tr>
<td>IB-IRM</td>
<td>56.1 ± 0.48%</td>
<td>45.0 ± 0.62%</td>
</tr>
<tr>
<td>V-REx</td>
<td>55.3 ± 1.75%</td>
<td>44.7 ± 1.31%</td>
</tr>
<tr>
<td>Coral</td>
<td>56.9 ± 0.25%</td>
<td>47.1 ± 0.43%</td>
</tr>
<tr>
<td>GroupDRO</td>
<td>58.8 ± 0.19%</td>
<td>46.5 ± 0.25%</td>
</tr>
<tr>
<td>DomainMix</td>
<td>58.6 ± 0.29%</td>
<td>48.9 ± 1.15%</td>
</tr>
<tr>
<td>Fish</td>
<td>57.8 ± 0.15%</td>
<td>49.5 ± 2.34%</td>
</tr>
<tr>
<td><strong>LISA (ours)</strong></td>
<td>58.7 ± 0.92%</td>
<td>48.7 ± 0.74%</td>
</tr>
</tbody>
</table>

Table 19. Full Results of RxRx1. ID: in-distribution; OOD: out-of-distribution.

<table>
<thead>
<tr>
<th></th>
<th>Validation Acc.</th>
<th>Test ID Acc.</th>
<th>Test OOD Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>19.4 ± 0.2%</td>
<td>35.9 ± 0.4%</td>
<td>29.9 ± 0.4%</td>
</tr>
<tr>
<td>IRM</td>
<td>5.6 ± 0.4%</td>
<td>9.9 ± 1.4%</td>
<td>8.2 ± 1.1%</td>
</tr>
<tr>
<td>IB-IRM</td>
<td>4.3 ± 0.7%</td>
<td>7.9 ± 0.5%</td>
<td>6.4 ± 0.6%</td>
</tr>
<tr>
<td>V-REx</td>
<td>5.2 ± 0.6%</td>
<td>9.3 ± 0.9%</td>
<td>7.5 ± 0.8%</td>
</tr>
<tr>
<td>Coral</td>
<td>18.5 ± 0.4%</td>
<td>34.0 ± 0.3%</td>
<td>28.4 ± 0.3%</td>
</tr>
<tr>
<td>GroupDRO</td>
<td>15.2 ± 0.1%</td>
<td>28.1 ± 0.3%</td>
<td>23.0 ± 0.3%</td>
</tr>
<tr>
<td>DomainMix</td>
<td>19.3 ± 0.7%</td>
<td>39.8 ± 0.2%</td>
<td>30.8 ± 0.4%</td>
</tr>
<tr>
<td>Fish</td>
<td>7.5 ± 0.6%</td>
<td>12.7 ± 1.9%</td>
<td>10.1 ± 1.5%</td>
</tr>
<tr>
<td><strong>LISA (ours)</strong></td>
<td><strong>20.1 ± 0.4%</strong></td>
<td><strong>41.2 ± 1.0%</strong></td>
<td><strong>31.9 ± 0.8%</strong></td>
</tr>
</tbody>
</table>

We consider the setting where $\alpha := \pi^{(1,G)} = \pi^{(0,R)}$ is relatively small and $\pi^{(1)} = \pi^{(0)} = \pi^{(G)} = \pi^{(R)} = 1/2$.

### B.1. Decomposing the loss function

Recall that $\Delta = \mu^{(1,G)} - \mu^{(0,G)} = \mu^{(1,R)} - \mu^{(0,R)}$. We further define $\tilde{\Delta} = \mu^{(1)} - \mu^{(0)}$, $\theta^{(G)} = \mu^{(0,G)} - \mathbb{E}[x_i]$, and $\theta^{(R)} = \mu^{(0,R)} - \mathbb{E}[x_i]$.

For the mixup estimators, we will repeatedly use the fact that $\lambda_i$ has a symmetric distribution with support $[0, 1]$.

For ERM estimator based on $(X, y)$, where $b_0 = \frac{1}{2} - \mathbb{E}[x_i]^T b$, we have

$$
(\mu^{(0,G)})^T b + b_0 = (\mu^{(0,G)} - \mathbb{E}[x_i])^T b + \frac{1}{2} \\
= (\theta^{(G)})^T b + \mathbb{E}[y_i] \\
(\mu^{(1,G)})^T b + b_0 = (\mu^{(1,G)} - \mathbb{E}[x_i])^T b + \frac{1}{2} \\
= \Delta^T b + (\theta^{(G)})^T b + \mathbb{E}[y_i],
$$

Notice that based on the estimator $b, b_0$, for $d \in \{G, R\}$,

$$
E^{(1,d)}(b, b_0) = \Phi(\frac{-\Delta^T b - (\theta^{(d)})^T b}{\sqrt{b^T \Sigma b}}) \quad \text{and} \quad E^{(0,d)}(b, b_0) = \Phi(\frac{(\theta^{(d)})^T b}{\sqrt{b^T \Sigma b}}),
$$

(7)

In the extreme case where $\pi_{0,R} = \pi_{1,G} = 0$, we have

$$
\tilde{\Delta} = \mu^{(1,R)} - \mu^{(0,G)}, \theta^{(G)} = -\frac{1}{2} \Delta, \theta^{(R)} = \frac{1}{2} \Delta - \Delta, \quad \text{and} \quad \Delta_0 := \mu^{(0,G)} - \mu^{(0,R)} = \Delta - \Delta.
$$
Table 20. Full Results of Amazon. Both the average accuracy and the 10th Percentile accuracy are reported.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>72.7 ± 0.1%</td>
<td>55.2 ± 0.7%</td>
<td>71.9 ± 0.1%</td>
<td>53.8 ± 0.8%</td>
</tr>
<tr>
<td>IRM</td>
<td>71.5 ± 0.3%</td>
<td>54.2 ± 0.8%</td>
<td>70.5 ± 0.3%</td>
<td>52.4 ± 0.8%</td>
</tr>
<tr>
<td>IB-IRM</td>
<td>72.4 ± 0.4%</td>
<td>55.1 ± 0.6%</td>
<td>72.2 ± 0.3%</td>
<td>53.8 ± 0.7%</td>
</tr>
<tr>
<td>V-REx</td>
<td>72.7 ± 1.2%</td>
<td>53.8 ± 0.7%</td>
<td>71.4 ± 0.4%</td>
<td>53.3 ± 0.0%</td>
</tr>
<tr>
<td>Coral</td>
<td>72.0 ± 0.3%</td>
<td>54.7 ± 0.0%</td>
<td>70.0 ± 0.6%</td>
<td>52.9 ± 0.8%</td>
</tr>
<tr>
<td>GroupDRO</td>
<td>70.7 ± 0.6%</td>
<td>54.7 ± 0.0%</td>
<td>70.0 ± 0.6%</td>
<td>53.3 ± 0.0%</td>
</tr>
<tr>
<td>DomainMix</td>
<td>71.9 ± 0.2%</td>
<td>54.7 ± 0.0%</td>
<td>71.1 ± 0.1%</td>
<td>53.3 ± 0.0%</td>
</tr>
<tr>
<td>Fish</td>
<td>72.5 ± 0.0%</td>
<td>54.7 ± 0.0%</td>
<td>71.7 ± 0.1%</td>
<td>53.3 ± 0.0%</td>
</tr>
<tr>
<td>LISA (ours)</td>
<td>71.6 ± 0.4%</td>
<td>55.1 ± 0.6%</td>
<td>70.8 ± 0.3%</td>
<td>54.7 ± 0.0%</td>
</tr>
</tbody>
</table>

Hence,

$$E_0^{(wst)} = \max \{ \Phi \left( \frac{1}{2} \Delta - \Delta \right)^T b \}, \Phi \left( \frac{1}{2} \Delta^T b \right).$$ \quad (8)

B.2. Classification errors of four methods with infinite training samples

We first provide the limit of the classification errors when $n \to \infty$.

B.2.1. Baseline method: ERM

For the training data, it is easy to show that

$$\text{var}(x) = \mathbb{E}[\text{var}(x|y)] + \text{var}(\mathbb{E}[x|y])$$

$$= \Sigma + \mathbb{E}[\text{var}(\mathbb{E}[x|y, D]|y)] + \text{var}(\mu^{(1)} - \mu^{(0)}) y$$

$$= \Sigma + \mathbb{E}[\text{var}(\mu^{(0,R)} - \mu^{(0,G)}) I(D = R)|y] + \Delta^{\otimes 2} \pi^{(1)} \pi^{(0)}$$

$$= \Sigma + \frac{1}{2} (\mu^{(0,R)} - \mu^{(0,G)}) \otimes 2 (\pi^{(1)} \pi^{(1)} + \pi^{(0)} \pi^{(0)})$$

$$\text{cov}(x, y) = \text{cov}(\mathbb{E}[x|y], y)$$

$$= \text{cov}(\mu^{(0)} + \Delta y, y)$$

$$= \text{cov}(\Delta y, y) = \Delta \pi^{(1)} \pi^{(0)}$$

For $a_0 = \frac{1}{2} (\pi^{(1)} \pi^{(1)} + \pi^{(0)} \pi^{(0)})$ and $\Delta_0 = \mu^{(0,G)} - \mu^{(0,R)}$, the ERM has slope and intercept being

$$b = \text{var}(x)^{-1} \text{cov}(x, y)$$

$$\propto (\Sigma + a_0 \Delta_0^{\otimes 2})^{-1} \Delta$$

$$= \Sigma^{-1} \Delta - \Sigma^{-1} \Delta_0 \frac{a_0 \Delta^T \Sigma^{-1} \Delta_0}{1 + a_0 \Delta_0^T \Sigma^{-1} \Delta_0}$$

$$b_0 = \mathbb{E}[y] - \mathbb{E}[x^T b].$$

B.2.2. Baseline method: Vanilla mixup

The vanilla mixup does not use the group information. Let $i_1$ be a random draw from $\{1, \ldots, n\}$. Let $i_2$ be a random draw from $\{1, \ldots, n\}$ independent of $i_1$. Let

$$\tilde{y}_i = \lambda_1 y_{i_1} + (1 - \lambda_1) y_{i_2}$$

and

$$\tilde{x}_i = \lambda_1 x_{i_1} + (1 - \lambda_1) x_{i_2}.$$
We can find that
\[
\text{cov}(\tilde{x}_i, \tilde{y}_i) = \text{cov}(\lambda_i x_{i_1} + (1 - \lambda_i) x_{i_2}, \lambda_i y_{i_1} + (1 - \lambda_i) y_{i_2})
\]
\[
= \text{cov}(\lambda_i x_{i_1}, \lambda_i y_{i_1}) + \text{cov}((1 - \lambda_i) x_{i_2}, (1 - \lambda_i) y_{i_2})
\]
\[
= (\mathbb{E}[\lambda_i^2] + \mathbb{E}[(1 - \lambda_i)^2]) \text{cov}(x_i, y_i).
\]
\[
\text{cov}(\tilde{x}_i) = \mathbb{E}[\lambda_i^2] + \mathbb{E}[(1 - \lambda_i)^2] \text{cov}(x_i).
\]

Hence, the population-level slope is the same as the slope in the benchmark method. It is easy to show that the population-level intercept is also the same. Hence,
\[
E_{\text{mix}}^{(wst)} = E_0^{(wst)}.
\]

**B.3. Intra-label LISA (LISA-L): mixup across domain**

Define
\[
x_i^{(\lambda)} = \lambda_i x_i^{(y_i,G)} + (1 - \lambda_i) x_i^{(y_i,R)},
\]
where \(i_1\) is a random draw from \(\{l : y_l = y_i, D_l = G\}\) and \(i_2\) is a random draw from \(\{l : y_l = y_i, D_l = R\}\). Then we perform OLS based on \((x_i^{(\lambda)}, y_i), i = 1, \ldots, n\).

We can calculate that
\[
\text{cov}(x_i^{(\lambda)}, y_i) = \text{cov}(\mathbb{E}[x_i^{(\lambda)}|y_i], y_i) = \text{cov}\left(\frac{1}{2} \mu^{(y_i,G)}, \frac{1}{2} \mu^{(y_i,R)}, y_i\right)
\]
\[
= \text{var}(y_i) \Delta = \pi^{(1)} \pi^{(0)} \Delta
\]
\[
\text{cov}(x_i^{(\lambda)}) = \mathbb{E}[\text{cov}(x_i^{(\lambda)}|y_i, \lambda_i)] + \text{cov}(\mathbb{E}[x_i^{(\lambda)}|y_i, \lambda_i])
\]
\[
= 2 \mathbb{E}[\lambda_i^2] \Sigma + \text{cov}(\lambda_i (\mu^{(0,G)} - \mu^{(0,R)}) + \Delta y_i)
\]
\[
= 2 \mathbb{E}[\lambda_i^2] \Sigma + \text{var}(\lambda_i) (\mu^{(0,G)} - \mu^{(0,R)}) \otimes^2 + \pi^{(1)} \pi^{(0)} \Delta \otimes^2.
\]

**B.4. Intra-domain LISA (LISA-D): mixup within each domain**

The interpolated sample can be written as
\[
(\tilde{y}_i, \tilde{x}_i) = (\lambda_i, \lambda_i x_{i_1}^{(y_i,G)} + (1 - \lambda_i) x_{i_2}^{(y_i,G)}) \text{ if } d_i = G
\]
\[
(\tilde{y}_i, \tilde{x}_i) = (\lambda_i, \lambda_i x_{i_1}^{(y_i,R)} + (1 - \lambda_i) x_{i_2}^{(y_i,R)}) \text{ if } d_i = R,
\]
where \(i_1\) is a random draw from \(\{l : d_l = d_i, y_l = 1\}\) and \(i_2\) is a random draw from \(\{l : d_l = d_i, y_l = 0\}\).

We consider regress \(\tilde{y}_i\) on \(\tilde{x}_i\).
\[
\text{cov}(\tilde{x}_i, \tilde{y}_i|d_i = G) = \text{cov}(\mathbb{E}[\tilde{x}_i|\tilde{y}_i, d_i = G], \tilde{y}_i|d_i = G) = \text{var}(\tilde{y}_i)(\mu^{(1,G)} - \mu^{(0,G)})
\]
\[
\text{var}(\tilde{x}_i|d_i = G) = \mathbb{E}[(\text{var}(\tilde{x}_i|\lambda_i, D_i = G)|d_i = G) + \text{var}(\mathbb{E}[\tilde{x}_i|\lambda_i, d_i = G]|D_i = G]
\]
\[
= 2 \mathbb{E}[\lambda_i^2] \Sigma + \text{var}(\lambda_i \mu^{(1,G)} + (1 - \lambda_i) \mu^{(0,G)}|d_i = G)
\]
\[
= 2 \mathbb{E}[\lambda_i^2] \Sigma + \text{var}(\tilde{y}_i) \Delta \otimes^2.
\]

We further have
\[
\text{cov}(\tilde{x}_i, \tilde{y}_i) = \mathbb{E}[(\text{cov}(\tilde{x}_i, \tilde{y}_i|d_i)) + \text{cov}(\mathbb{E}[\tilde{x}_i|d_i], \mathbb{E}[\tilde{y}_i|d_i])
\]
\[
= \text{cov}(\tilde{x}_i^{(G)}, \tilde{y}_i^{(G)}) \pi^{(G)} + \text{cov}(\tilde{x}_i^{(R)}, \tilde{y}_i^{(R)}) \pi^{(R)}
\]
\[
= \text{var}(\tilde{y}_i) (\mu^{(1,G)} - \mu^{(0,G)}) \pi^{(G)} + \text{var}(\tilde{y}_i) (\mu^{(1,R)} - \mu^{(0,R)}) \pi^{(R)}
\]
\[
= \text{var}(\tilde{y}_i) \Delta.
\]
Moreover,

\[
\text{var}(\tilde{x}_i) = \mathbb{E}\left[\text{var}(\tilde{x}_i|d_i)\right] + \text{var}(\mathbb{E}[\tilde{x}_i|d_i]) \\
= \text{var}(\tilde{x}_i^{(G)})\pi^{(G)} + \text{var}(\tilde{x}_i^{(R)})\pi^{(R)} + \left(\mathbb{E}[\tilde{x}_i^{(G)}] - \mathbb{E}[\tilde{x}_i^{(R)}]\right)^2\pi^{(G)}\pi^{(R)} \\
= 2\mathbb{E}[\lambda_i^2]\Sigma + \text{var}(\lambda_i)\Sigma^2 + \left(\mu^{(0,G)} - \mu^{(0,R)}\right)^2\pi^{(G)}\pi^{(R)}. 
\]

Slope:

\[
b = var(\tilde{x}_i)^{-1}\text{cov}(\tilde{x}_i, \tilde{y}_i) \\
\propto (\Sigma + a_{LD}\Delta_0^2)^{-1}\Delta \\
= \Sigma^{-1}\Delta - \Sigma^{-1}\Delta_0 \cdot \frac{a_{LD}(\Delta_0)^T\Sigma^{-1}\Delta}{1 + a_{LD}(\Delta_0)^T\Sigma^{-1}\Delta_0} \\
\propto \Sigma^{-1}\Delta + c_{LD}\Sigma^{-1}\Delta, 
\]

where \(a_{LD} = \frac{\pi^{(G)}}{2\mathbb{E}[\lambda_i^2]}\) and

\[
c_{LL} = \frac{1 + a_{LD}\Delta_0^T\Sigma^{-1}\Delta_0 - a_{LD}\Delta^T\Sigma^{-1}\Delta_0}{a_{LD}\Delta^T\Sigma^{-1}\Delta_0}.
\]

Moreover, \(b_0 = \mathbb{E}[\tilde{y}_i] - \mathbb{E}[\tilde{x}_i]^Tb = \frac{1}{2} - \mathbb{E}[\tilde{x}_i]^Tb\). Notice that

\[
\mathbb{E}[\tilde{x}_i] = \frac{1}{4}(\mu^{(0,G)} + \mu^{(1,G)} + \mu^{(0,R)} + \mu^{(1,R)}) \\
= \frac{1}{4}(2\mu^{(0,G)} + \Delta + 2\mu^{(1,R)} - \Delta) \\
= \frac{1}{2}(\mu^{(0,G)} + \mu^{(1,R)}) = \mathbb{E}[x_i].
\]

**Method comparison** We only need to compare \(E_{\text{ERM}}^{(\text{wst})}, E_{LL}^{(\text{wst})}, E_{LD}^{(\text{wst})}\).

For the ERM, \(0 \leq a_0 \leq 2\alpha\) and

\[
b_{\text{ERM}} = \left(1 + \frac{a_0\Delta^T\Sigma^{-1}\Delta_0}{1 + a_0\Delta^T\Sigma^{-1}\Delta_0}\right)\Sigma^{-1}\Delta - \frac{a_0\Delta^T\Sigma^{-1}\Delta_0}{1 + a_0\Delta^T\Sigma^{-1}\Delta_0}\Sigma^{-1}\Delta \\
\propto \Sigma^{-1}\Delta - \frac{a_0\Delta^T\Sigma^{-1}\Delta_0}{1 + a_0\Delta^T\Sigma^{-1}\Delta_0 + a_0\Delta^T\Sigma^{-1}\Delta_0}\Sigma^{-1}\Delta \\
\propto \Sigma^{-1}\Delta - \frac{a_0\Delta^T\Sigma^{-1}\Delta_0}{1 + a_0\Delta^T\Sigma^{-1}\Delta_0}\Sigma^{-1}\Delta.
\]

Let \(c_0 = \frac{a_0\Delta^T\Sigma^{-1}\Delta_0}{1 + a_0\Delta^T\Sigma^{-1}\Delta_0}\) and \(c_1 = |c_0||\Delta|/||\Delta||\Sigma|.\) For simplicity, let \(|v| = v^T\Sigma^{-1}v\). We first lower bound it via

\[
\text{cor}(b_{\text{ERM}}, \Delta) = \frac{b^T\Delta}{||\Delta||_{\Sigma}/b^T\Sigma b} = \frac{\Delta^T\Sigma^{-1}\Delta - c_0\Delta^T\Sigma^{-1}\Delta}{||\Delta||_{\Sigma}/b^T\Sigma b} \\
\geq \frac{\Delta^T\Sigma^{-1}\Delta}{||\Delta||_{\Sigma}/|\Delta| + |c_0||\Delta||_{\Sigma}} - \frac{c_0\xi||\Delta||_{\Sigma}}{b^T\Sigma b} \\
\geq \frac{1}{1 + |c_0||\Delta||_{\Sigma}/|\Delta|} - ||\Delta||_{\Sigma}/c_0||\Delta||_{\Sigma} \\
\geq \frac{1 - (1 + \xi)c_1 - c_1^2}{1 - c_1^2} = 1 - C\alpha.
\]
Similarly, we have
\[
\text{cor}(b_{\text{ERM}}, \Delta) = \frac{b^T \Delta}{\|\Delta\| \sqrt{b^T \Sigma b}} = \frac{\Delta^T \Sigma^{-1} \Delta - c_0 \Delta^T \Sigma^{-1} \Delta}{\|\Delta\| \sqrt{b^T \Sigma b}} \\
\leq \frac{1}{1 - c_0 \|\Delta\| / \|\Delta\|} \xi + \frac{c_0 \|\Delta\| / \|\Delta\|}{1 - c_0 \|\Delta\| / \|\Delta\|} \leq (\frac{\xi}{1 - c_1} - \frac{1}{1 - c_1}) \|\Delta\|.
\]
Hence,
\[
E_{\text{ERM}}^{(u)} \geq \max \left\{ \Phi((\frac{1}{2} - C\alpha)\|\Delta\| - (\xi + C\alpha)\|\Delta\|) \right\} \Phi((-\frac{1}{2} - C\alpha)\|\Delta\|) \tag{9}
\]
for some constant $C$ depending on the true parameters.

For method LISA-L, using the fact that $\Delta_0 = \Delta - \tilde{\Delta}$, for $a_{\text{LL}} = \text{var}(\lambda_i) / (2\mathbb{E}(|\lambda_i|^2))$,
\[
b_{\text{LL}} \propto \Sigma^{-1} \Delta + \frac{-a_{\text{LL}} \Delta^T \Sigma^{-1} \Delta_0}{1 + a_{\text{LL}} \Delta^T \Sigma^{-1} \Delta_0} \Sigma^{-1} \tilde{\Delta} \propto \Sigma^{-1} \Delta + c_{\text{LL}} \Sigma^{-1} \Delta
\]
for
\[
c_{\text{LL}} = \frac{1 + a_{\text{LL}} \Delta_0^T \Sigma^{-1} \Delta_0 - a_{\text{LL}} \Delta^T \Sigma^{-1} \Delta_0}{a_{\text{LL}} \Delta^T \Sigma^{-1} \Delta_0} = \frac{1 - a_{\text{LL}} \tilde{\Delta}^T \Sigma^{-1} \Delta_0}{a_{\text{LL}} \Delta^T \Sigma^{-1} \Delta_0}.
\]
Hence,
\[
\text{cor}(b_{\text{LL}}, \tilde{\Delta}) = \frac{\tilde{\Delta}^T b_{\text{LL}}}{\|\Delta\| \sqrt{b_{\text{LL}}^T \Sigma b_{\text{LL}}}} = \frac{\|\tilde{\Delta}\| \|\eta\| \|\Delta\|}{\|\Delta + c_{\text{LL}} \Delta\|}
\text{cor}(b_{\text{LL}}, \Delta) = \frac{b_{\text{LL}}^T \Delta}{\|\Delta\| \sqrt{b_{\text{LL}}^T \Sigma b_{\text{LL}}}} = \frac{\|\tilde{\Delta}\| \|\eta\| \|\Delta\|}{\|\Delta + c_{\text{LL}} \Delta\|}
\]
To have $E_{\text{LL}}^{(u)} < E_{\text{ERM}}^{(u)}$, it suffices to require that $(-\frac{1}{2} - C\alpha)\|\tilde{\Delta}\| < (\frac{1}{2} - C\alpha)\|\tilde{\Delta}\| - (\xi + C\alpha)\|\Delta\|$ and
\[
\frac{1}{2} \text{cor}(b_{\text{LL}}, \tilde{\Delta}) \|\tilde{\Delta}\| - \text{cor}(b_{\text{LL}}, \Delta) \|\Delta\| \leq (\frac{1}{2} - C\alpha)\|\tilde{\Delta}\| - (\xi + C\alpha)\|\Delta\|
\]
and
\[
A \text{ sufficient condition is}
\xi < (\frac{1}{2} + \frac{1}{2} \text{cor}(b_{\text{LL}}, \tilde{\Delta})) \|\tilde{\Delta}\| - C\alpha, \text{ cor}(b_{\text{LL}}, \Delta) \geq \xi + C\alpha, \text{ cor}(b_{\text{LL}}, \tilde{\Delta}) \leq 1 - 2C\alpha.
\]
We can find that a further sufficient condition is
\[
\xi < \frac{\|\tilde{\Delta}\|}{\|\Delta\|} - C\alpha, \text{ c}_{\text{LL}} > 0, \xi \leq \frac{\|\tilde{\Delta} + c_{\text{LL}} \Delta\| - \|\tilde{\Delta}\|}{c_{\text{LL}} \|\Delta\|} - \epsilon_1 \alpha \tag{10}
\]
\[
\|\tilde{\Delta} + c_{\text{LL}} \Delta\|\geq \|\tilde{\Delta}\|, \xi \leq \frac{c_{\text{LL}} \|\Delta\|}{\|\tilde{\Delta} + c_{\text{LL}} \Delta\| - \|\tilde{\Delta}\|} - \epsilon_1 \alpha \tag{11}
\]
\[
\xi \leq (\frac{1}{2} + \frac{1}{2} \text{cor}(b_{\text{LL}}, \tilde{\Delta})) \|\tilde{\Delta}\| - C\alpha. \tag{12}
\]
We first find sufficient conditions for the statements in (10) and (11). Parameterizing $t = c_{LL} \|\Delta\|_{\Sigma}/\|\tilde{\Delta}\|_{\Sigma}$, we further simplify the condition in (10) and (11) as

$$\xi < \min \left\{ \frac{\|\Delta\|_{\Sigma}}{\|\tilde{\Delta}\|_{\Sigma}}, 1 \right\} - C\alpha, t(t + 2\xi) > 0$$

$$\xi \leq \frac{\sqrt{1 + t^2 + 2t\xi} - 1}{t} - \epsilon_1\alpha, \quad \xi \leq \frac{1 + \sqrt{1 + t^2 + 2t\xi}}{t + 2\xi} - \epsilon_1\alpha.$$ 

We only need to require

$$t \geq \max \{0, -2\xi\} \text{ and } \xi \leq \min \left\{ \frac{\|\Delta\|_{\Sigma}}{\|\tilde{\Delta}\|_{\Sigma}}, 1 \right\} - C\alpha.$$ 

Some tedious calculation shows that $t \geq \max \{0, -2\xi\}$ can be guaranteed by

$$a_{LL} \leq \frac{1}{\|\Delta\|_{\Sigma}^2 + \|\Delta\|_{\Sigma}\|\tilde{\Delta}\|_{\Sigma}} \text{ and } \xi \leq \frac{\|\tilde{\Delta}\|_{\Sigma}}{\|\tilde{\Delta}\|_{\Sigma}}.$$ 

It is left to consider the constraint in (12). Notice that it holds for any $\xi \leq 0$. When $\xi > 0$, we can see

$$\text{cor}(b_{LL}, \tilde{\Delta}) = \frac{\|\tilde{\Delta}\|_{\Sigma} + \xi c_{LL} \|\Delta\|_{\Sigma}}{\|\Delta + c_{LL}\Delta\|_{\Sigma}} = \frac{1 + t\xi}{\sqrt{1 + t^2 + 2t\xi}}$$

$$\geq \frac{1 + t\xi}{1 + t} \geq \xi.$$

Hence, it suffices to guarantee that

$$(1 - \frac{1}{2} \frac{\|\tilde{\Delta}\|_{\Sigma}}{\|\Delta\|_{\Sigma}})\xi < \frac{1}{2} \frac{\|\tilde{\Delta}\|_{\Sigma}}{\|\Delta\|_{\Sigma}} - C\alpha.$$ 

If $\|\tilde{\Delta}\|_{\Sigma}/\|\Delta\|_{\Sigma} \geq 2$, then LHS is negative and it holds. If $1 \leq \|\tilde{\Delta}\|_{\Sigma}/\|\Delta\|_{\Sigma} < 2$, then the inequality becomes $\xi < 1 - C\alpha$. If $\|\tilde{\Delta}\|_{\Sigma}/\|\Delta\|_{\Sigma} < 1$, then the inequality becomes $\xi \leq \frac{\|\tilde{\Delta}\|_{\Sigma}}{\|\Delta\|_{\Sigma}} - C\alpha$. Because we have required $\xi < \min \left\{ \frac{\|\tilde{\Delta}\|_{\Sigma}}{\|\Delta\|_{\Sigma}}, 1 \right\} - C\alpha$ for some large enough $C$, the constraint (12) always holds. To summarize, $E_{LL} < E_{ERM}$ given that $\xi \leq \min \left\{ \frac{\|\tilde{\Delta}\|_{\Sigma}}{\|\Delta\|_{\Sigma}}, \frac{\|\tilde{\Delta}\|_{\Sigma}}{\|\Delta\|_{\Sigma}} \right\} - C\alpha$ for some large enough $C$ and $a_{LL} \leq 1/(\|\tilde{\Delta}\|_{\Sigma}^2 + \|\tilde{\Delta}\|_{\Sigma}\|\tilde{\Delta}\|_{\Sigma})$.

For method LISA-D, we can similarly show that $E_{LD} < E_{ERM}$ given that $\xi \leq \min \left\{ \frac{\|\tilde{\Delta}\|_{\Sigma}}{\|\Delta\|_{\Sigma}}, \frac{\|\tilde{\Delta}\|_{\Sigma}}{\|\Delta\|_{\Sigma}} \right\} - C\alpha$ for some large enough $C$ and $a_{LD} \leq 1/(\|\tilde{\Delta}\|_{\Sigma}^2 + \|\tilde{\Delta}\|_{\Sigma}\|\tilde{\Delta}\|_{\Sigma})$.

### B.5. Finite sample analysis

The empirical loss can be written as

$$\mathbb{P}(1(\langle x_i^{(G)} \rangle^T \hat{b} + \hat{b}_0 > \frac{1}{2}) \neq y_i^{(G)})$$

$$= \frac{1}{2} \mathbb{P}(\langle x_i^{(G)} \rangle^T \hat{b} + \hat{b}_0 > \frac{1}{2} | y_i^{(G)} = 0) + \frac{1}{2} \mathbb{P}(\langle x_i^{(G)} \rangle^T \hat{b} + \hat{b}_0 < \frac{1}{2} | y_i^{(G)} = 1),$$

where

$$\mathbb{P}(\langle x_i^{(G)} \rangle^T \hat{b} + \hat{b}_0 > \frac{1}{2} | y_i^{(G)} = 0) = \Phi(-\frac{1 - (\mu^{(0,G)})^T \hat{b} - \hat{b}_0}{\sqrt{\hat{b}_0 \Sigma}}),$$

$$\mathbb{P}(\langle x_i^{(G)} \rangle^T \hat{b} + \hat{b}_0 < \frac{1}{2} | y_i^{(G)} = 1) = \Phi(\frac{1 - (\mu^{(1,G)})^T \hat{b} - \hat{b}_0}{\sqrt{\hat{b}_0 \Sigma}}).$$

First notice that

$$\hat{b}_0 = \hat{y} - \hat{x}^T \hat{b}.$$
We have

\[
\begin{align*}
\mu^{(0,G)} T \hat{b} + \hat{b}_0 &= (\mu^{(0,G)} - \bar{x}) T \hat{b} + \bar{y} \\
&= (\mu^{(0,G)} - \mathbb{E}[x_i]) T \hat{b} + \frac{1}{2} \left( (\bar{y} - \bar{x} T \hat{b}) - (\mathbb{E}[y_i] - \mathbb{E}[x_i] T \hat{b}) \right) \\
\mu^{(1,G)} T \hat{b} + \hat{b}_0 &= (\mu^{(1,G)} - \bar{x}) T \hat{b} + \bar{y} \\
&= \Delta T \hat{b} + (\mu^{(0,G)} - \mathbb{E}[x_i]) T \hat{b} + \frac{1}{2} + R_1.
\end{align*}
\]

Therefore, according to (13),

\[
\frac{1}{2} \Phi\left( \frac{1}{\sqrt{b^T \Sigma b}} \left( \frac{1}{2} - (\mu^{(0,G)}) T \hat{b} - \hat{b}_0 \right) \right) + \frac{1}{2} \Phi\left( \frac{1}{\sqrt{b^T \Sigma b}} (\mu^{(1,G)} T \hat{b} - \hat{b}_0) \right)
\]

\[
= \frac{1}{2} \Phi\left( \frac{\left( \theta^{(G)} \right) T \hat{b} + R_1}{\sqrt{b^T \Sigma b}} \right) + \frac{1}{2} \Phi\left( \frac{\left( \theta^{(G)} \right) T \hat{b} + R_1}{\sqrt{b^T \Sigma b}} \right)
\]

\[
= \frac{1}{2} \Phi\left( \frac{\left( \theta^{(G)} \right) T \hat{b} + R_1}{\sqrt{b^T \Sigma b}} \right) + \frac{1}{2} \Phi\left( \frac{-\Delta + (\Theta^{(G)}) T \hat{b} + R_1}{\sqrt{b^T \Sigma b}} \right)
\]

\[
= \frac{1}{2} - \frac{1}{2} \left( \Phi\left( \frac{\left( \theta^{(G)} \right) T \hat{b} + R_1}{\sqrt{b^T \Sigma b}} \right) - \Phi\left( \frac{(\theta^{(G)}) T \hat{b} - \Delta T \hat{b} + R_1}{\sqrt{b^T \Sigma b}} \right) \right).
\]

Then the mis-classification error can be written as

\[
\frac{1}{2} - \frac{1}{2} \left( \Phi\left( \frac{\left( \theta^{(G)} \right) T \hat{b} + R_1}{\sqrt{b^T \Sigma b}} \right) - \Phi\left( \frac{(\theta^{(G)}) T \hat{b} - \Delta T \hat{b} + R_1}{\sqrt{b^T \Sigma b}} \right) \right).
\]

Larger the \(\hat{L}(\hat{b})\), smaller the mis-classification error.

We first find that

\[
\hat{L}(\hat{b}) - L(b) \leq C \left| \frac{(\theta^{(G)}) T \hat{b} + R_1}{\sqrt{b^T \Sigma b}} - \frac{(\theta^{(G)}) T \hat{b}}{\sqrt{b^T \Sigma b}} \right| + C \left| \frac{(\theta^{(G)}) T \hat{b} - \Delta T \hat{b} + R_1}{\sqrt{b^T \Sigma b}} - \frac{(\theta^{(G)}) T \hat{b} - \Delta T \hat{b}}{\sqrt{b^T \Sigma b}} \right|
\]

In the event that

\[
\|\Sigma^{1/2}(\hat{b} - b)\|_2 = o(1) \max_{y,d} \|\mu^{(y,d)}\|_2 \leq C, \Sigma \text{ is positive definite.}
\]

for the denominator, we have

\[
|b^T \Sigma b - \bar{b}^T \Sigma \bar{b}| \leq (2\|\Sigma^{1/2} b\|_2 + \|\Sigma^{1/2}(\hat{b} - b)\|_2)\|\Sigma^{1/2}(\hat{b} - b)\|_2 \\
\leq 2(1 + o(1))\|\Sigma^{1/2} b\|_2\|\Sigma^{1/2}(\hat{b} - b)\|_2
\]

\[
\|\sqrt{b^T \Sigma b} - \sqrt{b^T \Sigma b}\| \leq \frac{|\bar{b}^T \Sigma b - b^T \Sigma b|}{\sqrt{b^T \Sigma b} + \sqrt{b^T \Sigma b}} \\
\leq 2(1 + o(1))\|\Sigma^{1/2}(\hat{b} - b)\|_2.
\]

For the numerator, we have

\[
\frac{1}{2} \Delta T \hat{b} + R_1 - \frac{1}{2} \Delta T \hat{b} \leq |R_1| + \frac{1}{2}\|\Sigma^{-1/2} \Delta\|_2\|\Sigma^{1/2}(\hat{b} - b)\|_2.
\]
We arrive at
\[ T_1 \leq (1 + o(1)) \frac{|R_1| + \frac{1}{2} \|\Sigma^{-1/2} \Delta\|_2 \|\Sigma^{1/2}(\hat{b} - b)\|_2}{\|\Sigma^{1/2} b\|_2} + (1 + o(1)) \frac{\|\Delta^T b\| \|\Sigma^{1/2}(\hat{b} - b)\|_2}{\sqrt{b^T \Sigma b}}. \]

Moreover
\[ R_1 \leq \|\hat{b} - b\| + O_P(\frac{1}{\sqrt{n}}). \]

To summarize,
\[ \|\hat{b} - b\| \lesssim (1 + o(1)) (\|\hat{b} - b\|_2 + \frac{1}{\sqrt{n}}). \]

In the following, we will upper bound \( \|\hat{b} - b\|_2 \) for each method. For the ERM method, \( \hat{b} = \{(X - \bar{X})^T (X - \bar{X})\}^{-1} (X - \bar{X})^T (y - \bar{y}). \)

It is easy to show that \( \|\hat{b} - b\|_2^2 = O_P(\frac{\sum_{i=1}^N \text{var}(y_i|x_i)}{N^2}) = O_P(\frac{p}{N}). \)

For the vanilla mixup method, we first see that
\[ \frac{1}{n} \sum_{i=1}^n \bar{x}_i = \frac{1}{n} \sum_{i=1}^n (\lambda_i x_{i_1} + (1 - \lambda_i) x_{i_2}) = \bar{x} + O_P(n^{-1/2}) = \mu + O_P(n^{-1/2}) \]
\[ \frac{1}{n} \sum_{i=1}^n \bar{y}_i = \pi^{(1)} + O_P(n^{-1/2}). \]

Next,
\[ \frac{1}{n} \sum_{i=1}^n \bar{x}_i \bar{y}_i = \frac{1}{n} \sum_{i=1}^n \{\lambda_i^2 x_{i_1} y_{i_1} + (1 - \lambda_i)^2 x_{i_2} y_{i_2} + \lambda_i(1 - \lambda_i) x_{i_1} y_{i_2} + \lambda_i(1 - \lambda_i) x_{i_2} y_{i_1}\} \]
\[ \frac{1}{n} \sum_{i=1}^n \bar{x}_i \bar{y}_i - \mathbb{E}[\bar{x}_i \bar{y}_i] = \frac{1}{n} \sum_{i=1}^n \bar{x}_i \bar{y}_i - \mathbb{E}[\bar{x}_i \bar{y}_i|X, y] + \mathbb{E}[\bar{x}_i \bar{y}_i|X, y] - \mathbb{E}[\bar{x}_i \bar{y}_i]. \]

For \( E_2 \),
\[ E_2 = \frac{2\mathbb{E}[\lambda_i^2]}{n} \sum_{i=1}^n x_i y_i - \mathbb{E}[\bar{x}_i \bar{y}_i] = 2\mathbb{E}[\lambda_i^2] \mathbb{E}[x_i y_i]. \]

Hence,
\[ \|E_2\|_2^2 = O_P(\frac{p}{n}). \]

For \( E_1 \), conditioning on \((X, y), \lambda_i^2 x_{i_1} y_{i_1} - \mathbb{E}[\lambda_i^2] \sum_{i=1}^n x_i y_i\) are independent sub-Gaussian vectors. The sub-Gaussian norm of \( \frac{1}{N} \sum_{i=1}^n \lambda_i^2 x_{i_1} y_{i_1} - \mathbb{E}[\lambda_i^2] \sum_{i=1}^n x_i y_i\) (conditioning on \((X, y)\)) can be upper bounded by \( c \max_{i \leq N} |x_{i,j}| / \sqrt{n} \). Hence
\[ \mathbb{P}(\|E_1\|_2 \geq t|X, y) \leq 2 \exp\left\{ -\frac{c_2 n t^2}{\max_{j=1}^p \max_{i \leq N} x_{i,j}^2} \right\}. \]
As $x_{i,j}$ are Gaussian distributed, we know that
\[
\mathbb{P}\left(\sum_{j=1}^{p} \max_{i \leq n} x_{i,j}^2 \geq p \log n \right) \leq \exp\{-c_3 \log n\}.
\]

Hence, with probability at least $1 - \exp(-c_1 \log n)$,
\[
E_1 \leq \frac{Cp \log n}{n}.
\]

To summarize,
\[
\left\| \frac{1}{n} \sum_{i=1}^{n} \hat{x}_i \hat{y}_i - \left( \frac{1}{n} \sum_{i=1}^{n} \hat{x}_i \right) \left( \frac{1}{n} \sum_{i=1}^{n} \hat{y}_i \right) - \text{cov}(\hat{x}_i, \hat{y}_i) \right\|_2^2 = O_p\left( \frac{p \log n}{n} \right).
\]

Similarly, we can show that
\[
\left\| \frac{1}{n} \sum_{i=1}^{n} \hat{x}_i \hat{x}_i^T - \left( \frac{1}{n} \sum_{i=1}^{n} \hat{x}_i \right) \left( \frac{1}{n} \sum_{i=1}^{n} \hat{x}_i^T \right) - \text{cov}(\hat{x}_i) \right\|_2^2 = O_p\left( \frac{p \log n}{n} \right).
\]

Hence,
\[
\| \hat{b} - b \|_2^2 = O_p\left( \frac{p \log n}{n} \right).
\]

For the LISA-L, we first see that
\[
\frac{1}{n} \sum_{i=1}^{n} x_{i}^{(\lambda)} = \frac{1}{n} \sum_{i=1}^{n} (\lambda_i x_{i1}^{(1,G)} + (1 - \lambda_i)x_{i2}^{(1,R)}) + \frac{1}{n} \sum_{i=1}^{n} (\lambda_i x_{i1}^{(0,G)} + (1 - \lambda_i)x_{i2}^{(0,R)})
\]
\[
= \frac{1}{2} (\bar{x}^{(1,G)} + \bar{x}^{(1,R)}) \tilde{\pi}_1 + \frac{1}{2} (\bar{x}^{(0,G)} + \bar{x}^{(0,R)}) \tilde{\pi}_0
\]

We have
\[
\frac{1}{n} (X^{(\lambda)})^T y - \bar{y} \frac{1}{n} \sum_{i=1}^{n} x_{i}^{(\lambda)} - \text{cov}(x_{i}^{(\lambda)}, y_i) = \frac{1}{n} (X^{(\lambda)})^T y - \bar{y} \frac{1}{n} \sum_{i=1}^{n} x_{i}^{(\lambda)} - \text{cov}(x_{i}^{(\lambda)}, y_i) \underbrace{E_1}_{E_2} + \text{cov}(x_{i}^{(\lambda)}, y_i | X, y) - \text{cov}(x_{i}^{(\lambda)}, y_i)
\]

For $E_2$,
\[
E_2 = \frac{\tilde{\pi}_1}{2} (\bar{x}^{(1,G)} + \bar{x}^{(1,R)}) - \tilde{\pi}_1 \left( \frac{1}{2} (\bar{x}^{(1,G)} + \bar{x}^{(1,R)}) \tilde{\pi}_1 + \frac{1}{2} (\bar{x}^{(0,G)} + \bar{x}^{(0,R)}) \tilde{\pi}_0 - \text{cov}(x_{i}^{(\lambda)}, y_i) \right)
\]
\[
= \frac{1}{2} (\bar{x}^{(1,G)} + \bar{x}^{(1,R)} - \bar{x}^{(0,G)} - \bar{x}^{(0,R)}) \tilde{\pi}_1 \tilde{\pi}_0 - \pi^{(1)} \pi^{(0)} \Delta.
\]

It is easy to show that
\[
\| E_2 \|_2^2 = O_P\left( \frac{p}{\min_{y,e} \| y | e \|_n} \right).
\]

For $E_1$, conditioning on $X$ and $y$, $x_{i}^{(\lambda)} y_i - \mathbb{E}[x_{i}^{(\lambda)} y_i | X, y]$ are independent sub-Gaussian vectors with mean zero. The sub-Gaussian norm of $\frac{1}{n} \sum_{i=1}^{n} x_{i,j}^{(\lambda)} y_i$ (conditioning on $X$ and $y$) can be upper bounded by $c \max_{i \leq n} |x_{i,j}| / \sqrt{N}$.

\[
\mathbb{P}(\| E_1 \|_2 \geq t | X, y) = \frac{1}{\sqrt{p}} \sum_{j=1}^{p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left( x_{i,j}^{(\lambda)} y_i - \mathbb{E}[x_{i,j}^{(\lambda)} y_i | X, y] \right)^2 \right\}^{1/2} \geq t^2 | X, y \}
\]
\[
\leq 2 \exp\left\{ - \frac{c_2 n t^2}{\sum_{j=1}^{p} \max_{i \leq n} x_{i,j}^{2}} \right\}.
\]
The sub-exponential norm of $n \log n$. Hence,

$$E_1 = O_p\left(\sqrt{\frac{\sum_{i=1}^{p} \max_{x_i} \|x_i\|^2}{n}}\right) = O_p\left(\frac{p \log n}{n}\right).$$

To summarize,

$$\left\| \frac{1}{n} (X(\lambda))^T y - \mathbb{E}[x_i(y_i)] \right\|^2 = O_p\left(\frac{p}{\min_{y, e, \epsilon} n(y, e)} + \frac{p \log n}{n}\right).$$

We can use similar analysis to bound

$$\left\| \frac{1}{n} (X(\lambda))^T X(\lambda) - \mathbb{E}[x_i(x_i)^T] \right\|_2.$$ 

The sub-exponential norm of $\frac{1}{n} \sum_{i=1}^{N} x_{i,j} x_{i,k}$ (conditioning on $X$) can be upper bounded by $\max_{i \leq N} \|x_i\| x_{i,k}/\sqrt{N}$. We can show that

$$\left\| \frac{1}{n} (X(\lambda))^T X(\lambda) - \mathbb{E}[x_i(x_i)^T] \right\|_2 = O_p\left(\frac{p}{\min_{y, e, \epsilon} n(y, e)} + \frac{p \log n}{n}\right).$$

For the LISA-D, we first see that

$$\frac{1}{n} \sum_{i=1}^{n} \tilde{x}_i = \frac{1}{n} \sum_{D_i = G} (\lambda_i x_{i1}^{(1,G)} + (1 - \lambda_i) x_{i2}^{(0,G)}) + \frac{1}{n} \sum_{D_i = R} (\lambda_i x_{i1}^{(1,R)} + (1 - \lambda_i) x_{i2}^{(0,R)})$$

$$= \frac{1}{2} (\bar{x}^{(1,G)} + \bar{x}^{(0,G)}) \bar{\pi}_G + \frac{1}{2} (\bar{x}^{(1,R)} + \bar{x}^{(0,R)}) \bar{\pi}_R$$

$$\tilde{y} = \frac{1}{2}.$$ 

Next,

$$\frac{1}{n} \sum_{i=1}^{n} \tilde{x}_i \tilde{y}_i = \frac{1}{n} \sum_{D_i = G} \lambda_i^2 x_{i1}^{(1,G)} + \lambda_i (1 - \lambda_i) x_{i2}^{(0,G)} + \frac{1}{n} \sum_{D_i = R} \lambda_i^2 x_{i1}^{(1,R)} + \lambda_i (1 - \lambda_i) x_{i2}^{(0,R)}$$

$$= \frac{1}{2} (\bar{x}^{(1,G)} + \bar{x}^{(0,G)}) \bar{\pi}_G + \frac{1}{2} (\bar{x}^{(1,R)} + \bar{x}^{(0,R)}) \bar{\pi}_R - \text{cov}(\tilde{x}, \tilde{y}) \text{cov}(\tilde{x}, \tilde{y}) + \text{cov}(\tilde{x}_i, \tilde{y}_i | X, y) + \text{cov}(\tilde{x}_i, \tilde{y}_i | X, y) - \text{cov}(\tilde{x}, \tilde{y}) \text{cov}(\tilde{x}, \tilde{y}).$$

For $E_2$,

$$E_2 = \pi^{(G)} \mathbb{E}[\lambda_i^2] (\bar{x}^{(1,G)} - \bar{x}^{(0,G)}) + \frac{1}{2} \bar{x}^{(0,G)} + \pi^{(R)} \mathbb{E}[\lambda_i^2] (\bar{x}^{(1,R)} - \bar{x}^{(0,R)}) + \frac{1}{2} \bar{x}^{(0,R)} - \pi^{(G)} \pi^{(R)} \text{var}(\lambda_i) \Delta$$

Next, $E_2$ is a sub-Gaussian vector with sub-Gaussian norm upper bounded by

$$\bar{\pi}_G^2 + \bar{\pi}_R^2 + \frac{\pi_d}{\pi_{y,d}} + \frac{\pi_d}{\pi_{y,d}} \leq 4 \max_{y,d} \frac{\pi_d}{\pi_{y,d}}.$$ 

Using sub-Gaussian concentration, we can show that

$$E_2 = O_p\left(\sqrt{\frac{p \max_{y,d} \pi_d}{\pi_{y,d}}} \right).$$
Notice that $\max_{y,d} \frac{\pi_{yd}}{\pi_{y|d}} \geq 1$. For $E_1$, conditioning on $X$ and $y \tilde{x}_i y_i - \mathbb{E}[\tilde{x}_i y_i | X, y]$ are independent sub-Gaussian vectors with mean zero. The sub-Gaussian norm of $\frac{1}{n} \sum_{i=1}^n \tilde{x}_i y_i$ conditioning on $X$ and $y$ can be upper bounded by $c \max_{i,j} |x_{i,j}|$. Similar analysis on $E_1$ leads to

$$\frac{1}{n} \sum_{i=1}^n \tilde{x}_i y_i - \mathbb{E}[\tilde{x}_i y_i] = O_P\left(\sqrt{\frac{p \log n}{n}} + \sqrt{\frac{p \max_{y,d} \pi_{yd}}{n \max_{y,d} \pi_{y|d}}} \right).$$

For the sample covariance matrix, we can also show that

$$\left\| \frac{1}{n} \sum_{i=1}^n \tilde{x}_i \tilde{x}_i^T - \frac{1}{n} \sum_{i=1}^n \tilde{x}_i \tilde{x}_i^T - \mathbb{E}[\tilde{x}_i \tilde{x}_i^T] \right\|_2 = O_P\left(\sqrt{\frac{p \log n}{n}} + \sqrt{\frac{p \max_{y,d} \pi_{yd}}{n \max_{y,d} \pi_{y|d}}} \right).$$

**B.6. A $\xi$-dependent lower bound for $E_{\text{ERM}}^{(wst)} - E_{\text{LL}}^{(wst)}$**

Next, we provide a $\xi$-dependent lower bound for $E_{\text{ERM}}^{(wst)} - E_{\text{LL}}^{(wst)}$. Based on our previous analysis

$$E_{\text{ERM}}^{(wst)} - E_{\text{LL}}^{(wst)} \geq c_1 \min \left\{ \left( \frac{1}{2} - C\alpha \right) \right\} \Delta \Delta_{\Sigma} + (\text{cor}(b_{\text{LL}}, \tilde{\Delta}) - \xi - C\alpha) \Delta \Delta_{\Sigma},$$

$$\left( \frac{1}{2} - C\alpha + \frac{1}{2} \text{cor}(b_{\text{LL}}, \tilde{\Delta}) \right) \Delta \Delta_{\Sigma} - (\xi + C\alpha) \Delta \Delta_{\Sigma},$$

where $c_1$ is a positive constant given by the derivative of $\Phi(\cdot)$. Plugging in the expression of $\text{cor}(b_{\text{LL}}, \tilde{\Delta})$ and $\text{cor}(b_{\text{LL}}, \Delta)$, we have for the first term of $E_{\text{ERM}}^{(wst)} - E_{\text{LL}}^{(wst)}$, it is no smaller than

$$\frac{1}{2} \left( 1 - 2C\alpha - \frac{1 + \xi t}{\sqrt{1 + t^2 + 2t\xi}} \right) \Delta \Delta_{\Sigma} + \left( \frac{\xi + t}{\sqrt{1 + t^2 + 2t\xi}} - \xi - C\alpha \right) \Delta \Delta_{\Sigma} \geq \frac{1}{2} \left( \frac{t^2}{1 + t^2} - 2\xi \right) \Delta \Delta_{\Sigma} + \frac{t^2}{1 + t^2} \left( \frac{1 - \xi}{\Delta \Delta_{\Sigma}} \right) \Delta \Delta_{\Sigma} - C\alpha(\Delta \Delta_{\Sigma} + \Delta \Delta_{\Sigma}),$$

where the last step is due to the current constraint that $t > \max\{0, -2\xi\}$. For the second term, it is no smaller than

$$\Delta \Delta_{\Sigma} - \xi \Delta \Delta_{\Sigma} - C\alpha(\Delta \Delta_{\Sigma} + \Delta \Delta_{\Sigma}).$$

Notice that $t^2/(1 + t^2) \geq \min\{\frac{t^2}{4}, \frac{1}{2}\}$. We can show that $t \geq \frac{\Delta \Delta_{\Sigma}}{\Delta \Delta_{\Sigma}}$. Then

$$E_{\text{ERM}}^{(wst)} - E_{\text{LL}}^{(wst)} \geq c_3 \min \left\{ \left( \Delta \Delta_{\Sigma} - \xi \right) \Delta \Delta_{\Sigma}, \left( 1 - \xi \right) \Delta \Delta_{\Sigma}, \left( 1 - \xi \right) \Delta \Delta_{\Sigma}^2 / \Delta \Delta_{\Sigma}^2 \right\} - c_4 \alpha(\Delta \Delta_{\Sigma} + \Delta \Delta_{\Sigma}).$$

**B.7. Domain shifts: Proof of Theorem 2**

It still holds that $\Delta^* = 2(\mu^{(0,s)} - \mathbb{E}[x_i^{(1)}]) = 2(\mu^{(0,s)} - \mathbb{E}[x_i])$. It is easy to show that the worst group mis-classification error for this new environment is

$$E_{A}^{(wst,s)} = \max \left\{ \Phi \left( -\frac{1}{2} \Delta^* \Sigma b_{A} \right), \Phi \left( \frac{1}{2} \gamma \Delta^* \Sigma b_{A} - \Delta^* T b_{A} \right) \right\},$$

where $A \in \{\text{ERM}, \text{mix}, \text{LL}, \text{LD}\}$. Notice that

$$\Delta^* = 2\mu^{(0,s)} - (\mu^{(0,G)} + \mu^{(1,R)}) = \Delta + \mu^{(0,s)} - \mu^{(0,G)}.$$
Hence,
\[
E_{\text{ERM}}^{(\text{wst})} \geq \max \left\{ \Phi\left(\frac{\gamma}{2} - C\alpha\right) \|\Delta\|_\Sigma - \left(\xi - C\alpha\right) \|\Delta\|_\Sigma\right\}, \Phi\left(\frac{-\gamma}{2} - C\alpha\right) \|\Delta\|_\Sigma \right\}
\]
(16)
for some constant \(C\) depending on the true parameters.

Hence,
\[
cor(b_{LL, \Delta^*}) = \frac{(\Delta^*)^T b_{LL}}{\|\Delta^*\|_\Sigma \sqrt{b_{LL}^T \Sigma b_{LL}}} = \frac{\gamma \|\Delta\|_\Sigma + c_{LL} \xi^* \|\Delta\|_\Sigma}{\|\Delta + c_{LL} \Delta\|_\Sigma}.
\]

To have \(E_{\text{LL}}^{(\text{wst})} < E_{\text{ERM}}^{(\text{wst})}\), it suffices to require that \((-\frac{\gamma}{2} - C\alpha) \|\Delta\|_\Sigma < \left(\frac{\gamma}{2} - C\alpha\right) \|\Delta\|_\Sigma - (\xi + C\alpha) \|\Delta\|_\Sigma\) and
\[
\frac{1}{2} \text{cor}(b_{LL, \Delta^*}) \|\Delta\|_\Sigma - \text{cor}(b_{LL, \Delta}) \|\Delta\|_\Sigma \leq \left(\frac{\gamma}{2} - C\alpha\right) \|\Delta\|_\Sigma - (\xi + C\alpha) \|\Delta\|_\Sigma;
\]
\[
- \frac{1}{2} \text{cor}(b_{LL, \Delta^*}) \|\Delta\|_\Sigma \leq \left(\frac{\gamma}{2} - C\alpha\right) \|\Delta\|_\Sigma - (\xi + C\alpha) \|\Delta\|_\Sigma.
\]

A sufficient condition is
\[
\xi < \left(\frac{\gamma}{2} + \frac{1}{2} \text{cor}(b_{LL, \Delta^*}) \right) \|\Delta\|_\Sigma - C\alpha, \text{cor}(b_{LL, \Delta}) \geq \xi + C\alpha, \text{cor}(b_{LL, \Delta^*}) \leq \gamma - 2C\alpha.
\]

We can find that a further sufficient condition is
\[
\xi < \frac{1 + \gamma}{2} \frac{\|\Delta\|_\Sigma}{\|\Delta\|_\Sigma} - C\alpha, c_{LL} > 0, \xi^* \leq \frac{\gamma(\|\Delta + c_{LL} \Delta\|_\Sigma - \|\Delta\|_\Sigma)}{c_{LL} \|\Delta\|_\Sigma} - \epsilon_1 \alpha
\]
(17)
\[
\|\Delta + c_{LL} \Delta\|_\Sigma \geq \|\Delta\|_\Sigma, \xi \leq \frac{c_{LL} \|\Delta\|_\Sigma}{\|\Delta + c_{LL} \Delta\|_\Sigma - \|\Delta\|_\Sigma} - \epsilon_1 \alpha
\]
(18)
\[
\xi \leq \left(\frac{\gamma}{2} + \frac{1}{2} \text{cor}(b_{LL, \Delta^*}) \right) \frac{\|\Delta\|_\Sigma}{\|\Delta\|_\Sigma} - C\alpha.
\]
(19)

We first find sufficient conditions for the statements in (10) and (11). Parameterizing \(t = c_{LL} \|\Delta\|_\Sigma / \|\Delta\|_\Sigma\), we further simplify the condition in (17) and (18) as
\[
\xi < \frac{1 + \gamma}{2} \frac{\|\Delta\|_\Sigma}{\|\Delta\|_\Sigma} - C\alpha, t > 0, \xi^* \leq \frac{\gamma(\sqrt{1 + t^2 + 2t \xi} - 1)}{t} - \epsilon_1 \alpha,
\]
\[
- \frac{t}{2} \leq \xi \leq t, \xi \leq \frac{1 + \sqrt{1 + t^2 + 2t \xi}}{t + 2\xi} - \epsilon_1 \alpha.
\]

We only need to require
\[
t \geq \max\{0, -2\xi\} \text{ and } \xi < \min\{\frac{1 + \gamma}{2} \frac{\|\Delta\|_\Sigma}{\|\Delta\|_\Sigma}, 1\} - C\alpha, \xi^* \leq \gamma \xi.
\]

Some tedious calculation shows that \(t \geq \max\{0, -2\xi\}\) can be guaranteed by
\[
a_{LL} \leq \frac{1}{\|\Delta\|_\Sigma^2 + \|\Delta\|_\Sigma \|\Delta\|_\Sigma} \text{ and } \xi \leq \frac{\|\Delta\|_\Sigma}{\|\Delta\|_\Sigma}
\]

It is left to consider the constraint in (19). Notice that it holds for any \(\xi \leq 0\). When \(\xi > 0\), we can see
\[
cor(b_{LL, \Delta^*}) = \frac{\gamma \|\Delta\|_\Sigma + \xi^* c_{LL} \|\Delta\|_\Sigma}{\|\Delta + c_{LL} \Delta\|_\Sigma} = \frac{\gamma + t \xi^*}{\sqrt{1 + t^2 + 2t \xi}}
\]
\[
\geq \frac{\gamma + t \xi^*}{1 + t}.
\]
Hence, it suffices to guarantee that
\[ \xi^* + \gamma \geq \frac{2\|\Delta\|_\Sigma}{\|\Delta\|_\Sigma} \xi + C\alpha. \]

To summarize, it suffices to require
\[ a_{LL} \leq \frac{1}{\|\Delta\|_\Sigma^2 + \|\Delta\|_\Sigma \|\Delta\|_\Sigma}, \ 0 \leq \xi^* \leq \gamma \xi, \xi < \min\left\{ \frac{\gamma}{2} \frac{\|\Delta\|_\Sigma}{\|\Delta\|_\Sigma}, \frac{\|\Delta\|_\Sigma}{\|\Delta\|_\Sigma} \right\} - C\alpha. \]

For LISA-D, we can similarly show that \( E_{LD}^{(wst)} < E_{ERM}^{(wst)} \) given that
\[ a_{LD} \leq \frac{1}{\|\Delta\|_\Sigma^2 + \|\Delta\|_\Sigma \|\Delta\|_\Sigma}, \ 0 \leq \xi^* \leq \gamma \xi, \xi < \min\left\{ \frac{\gamma}{2} \frac{\|\Delta\|_\Sigma}{\|\Delta\|_\Sigma}, \frac{\|\Delta\|_\Sigma}{\|\Delta\|_\Sigma} \right\} - C\alpha. \]