Spuriosity Didn’t Kill the Classifier: Using Invariant Predictions to Harness Spurious Features

Cian Eastwood 1,2 Shashank Singh 1
Andrei L. Nicolicioiu 1 Marin Vlastelica 1
Julius von Kügelgen 1,3 Bernhard Schölkopf 1

Abstract

To avoid failures on out-of-distribution data, recent works have sought to use only features with an invariant or stable relationship with the label across domains, discarding “spurious” or unstable features whose relationship with the label changes across domains. However, unstable features often carry complementary information about the label that could boost performance if used correctly in the test domain. Our main contribution is to show that it is possible to learn how to use these unstable features in the test domain without labels. We prove that pseudo-labels based on stable features provide sufficient guidance for doing so, provided that stable and unstable features are conditionally independent given the label. Based on this insight, we propose Stable Feature Boosting (SFB), an algorithm for: (i) learning stable and conditionally-independent unstable features; and (ii) using the stable-feature predictions to adapt the unstable-feature predictions to the test domain. Theoretically, we prove that SFB can learn an asymptotically-optimal predictor without test-domain labels. Empirically, we demonstrate the effectiveness of SFB on real and synthetic data.

1. Introduction and Related Work

Machine learning systems often rely on “spurious” features whose relationship with the label changes across domains, leading to poor performance in test domains of interest (Geirhos et al., 2020). Recent works have thus sought predictors which do not rely on these spurious or unstable relationships, but instead leverage relationships which are invariant or stable across multiple domains (Peters et al., 2016; Arjovsky et al., 2020; Krueger et al., 2021; Eastwood et al., 2022). However, despite their instability, spurious features can often provide additional or complementary information about the target label. Thus, if a predictor could be adjusted to use spurious features optimally in the test domain, it would boost performance substantially. That is, perhaps we don’t need to discard spurious features at all, but rather, use them in the right way.

As a simple but illustrative example, consider the CMNIST or ColorMNIST dataset (Arjovsky et al., 2020). This transforms the original MNIST dataset into a binary classification task (digit in 0–4 or 5–9) and then: (i) flips the label with probability 0.25, meaning that, across all 3 domains, digit shape correctly determines the label with probability 0.75; and (ii) colorizes the digit such that digit color (red or green) is a more informative but spurious feature (see Fig. 5 of Appendix E). Prior work focused on learning invariant predictors that use only shape and avoid using color—a spurious feature whose relationship with the label changes across domains. However, as shown in Fig. 1a, the invariant predictor is not Bayes-optimal in many test domains since spurious features can be used in a domain-specific manner to improve performance. Hence, we ask: when and how can such informative but spurious features be reliably harnessed without labels?

Related work. Table 1 summarises related work. For space reasons, all other related work is deferred to Appendix H.
2. Harnessing Unstable Features

Problem setup. We consider the problem of domain generalization (DG) where predictors are trained on data from multiple training domains and with the goal of performing well on data from unseen test domains. More formally, we consider datasets \( D^e = \{ (X^e_i, Y^e_i) \}_{i=1}^n \) collected from \( m \) different training domains or environments \( \mathcal{E}_u := \{ E_1, \ldots, E_m \} \), with each dataset \( D^e \) containing data pairs \( (X^e_i, Y^e_i) \) sampled i.i.d. from \( \mathcal{P}(X^e, Y^e) \).\(^1\) The goal is then to learn a predictor \( f \) that performs well on data from a new test domain from a larger set of all possible domains \( \mathcal{E}_{all} \supset \mathcal{E}_u \).

Stable and unstable features. To avoid failures on OOD data, recent works in DG have sought robust predictors that only use stable or invariant features, i.e., those which have a stable or invariant relationship with the label across domains (Peters et al., 2016). In particular, Arjovsky et al. (2020) learn features which have an invariant functional relationship with the label by enforcing that the classifier on top of these features is optimal for all domains simultaneously. We henceforth use stable features and \( X_S \) to refer to these features, and stable predictors to refer to predictors which use only these features. Analogously, we use unstable features and \( X_U \) to refer to features with an unstable or “spurious” relationship with the label across domains. Formal definitions for both stable and unstable features are provided in § 3. Note that \( X_S \) and \( X_U \) form a partition of the components of \( X \) which are informative about \( Y \), as depicted in Fig. 1b.

Harnessing unstable features with labels. A stable predictor \( f_S \) is unlikely to be the best predictor in any given domain. As illustrated in Fig. 1, this is because it excludes unstable features \( X_U \) which are informative about \( Y \) and can boost performance if used in an appropriate, domain-specific manner. Assuming that we can indeed learn a stable predictor using prior methods, e.g., IRM (Arjovsky et al., 2020), we now show how \( X_U \) can be harnessed with labels to boost performance. To begin, note that we need only update the \( X_U \)-\( Y \) relation since, by definition, the \( X_S \)-\( Y \) relation is stable across domains. We thus seek a feature space which separates \( X_S \) and \( X_U \), allowing only the unstable \( X_U \)-\( Y \) relation to be updated. To do so, we decompose a predictor \( f = h \circ \Phi \) into feature representation \( \Phi \) and classifier \( h \) and then describe the boosted joint predictor \( f^e \) in domain \( e \) as:

\[
f^e(X) = f_S(X) + f^e(X) = h_S(\Phi_S(X)) + h^e(\Phi_U(X)) = h_S(X_S) + h^e(X_U).
\]

Here, both \( f_S \) and \( f^e \) produce logits, with \( f^e \) adding a domain-specific adjustment to \( f_S \) in logit space. As illustrated by Eqs. (2.2) and (2.3), the role of \( \Phi_S \) and \( \Phi_U \) is to extract \( X_S \) and \( X_U \), respectively, from the observed features \( X \). Note that the stable predictor \( f_S \) and classifier \( h_S \), as well as the feature extractors \( \Phi_S \) and \( \Phi_U \), are shared across domains \( e \), whereas the unstable classifier \( h^e_U \) is not. In principle, \( h^e_U \) could take any form and we could learn completely separate \( \Phi_S, \Phi_U \). In practice, however, we generally take \( h^e_U \) to be a linear classifier and simply split the output features of a shared \( \Phi(X) = (\Phi_S(X), \Phi_U(X)) \) into two parts.

Given a new domain \( e \) and with labels \( Y^e \), we can then boost performance by adapting \( h^e_U \). More specifically, letting \( \ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R} \) be a loss function (e.g., cross-entropy) and \( R^e(f) = \mathbb{E}_{(X,Y)} [\ell(Y, f(X)) \mid E = e] \) the risk of a predictor \( f : \mathcal{X} \to \mathcal{Y} \) in domain \( e \), we can adapt \( h^e_U \) to solve:

\[
\min_{h_U} \sum_{e \in \mathcal{E}_u} R^e(\sigma \circ ((h_S \circ \Phi_S) + (h_U \circ \Phi_U)))
\]

Harnessing unstable features without labels. We now consider the main question of this work—can we reliably harness \( X_U \) without test-domain labels? To begin, note that, while we don’t have labels in the test domain, we do have stable predictions. By definition, these are imperfect (i.e., noisy) but robust, and can be used to form pseudo-labels \( \hat{Y}_i = \arg \max_j f_S(X_i)_j \), with \( f_S(X_i)_j \) denoting the \( j \)th logit of the stable prediction for \( X_i \). Can we somehow use these noisy but robust pseudo-labels to guide our updating of \( h^e_U \), and, ultimately, our use of \( X_U \) in the test domain? Unfortunately, if we try to use our robust pseudo-labels as if they were true labels—updating \( h^{e_U} \) to minimize the joint risk as in Eq. (2.4)—we get a trivial solution of \( h^{e_U} = 0 \) achieves 100% accuracy. For cross-entropy, the same applies (see Prop. D.1 of App. D). Thus, we cannot minimize a joint loss involving \( f_S \)’s predictions when using \( f_S \)’s pseudo-labels. Instead, we consider minimizing the unstable-only risk \( R^e(\sigma \circ h_S \circ \Phi_S) \). While this could work, it raises many questions about \( h_S \) when it will work. We now summarise these questions before addressing them in § 3:

1. When can we minimize the unstable-predictor risk alone/separately? When does this lead to the optimal joint predictor? This won’t always work; e.g., for independent \( X_S, X_U \sim \text{Bernoulli}(1/2) \) and \( Y = X_S \text{ XOR } X_U \), \( Y \) is independent of each of \( X_S \) and \( X_U \) and hence cannot be predicted from either alone.

\(^1\)We drop the superscript \( e \) when referring to any environment.
2. Can we just add the logits as before, in Eq. (2.3)? Intuitively, doing so would require them both to be “of the same scale”, or, more precisely, properly calibrated. Do we have any reason to believe that, after training on the pseudo-labels, \( h_t^{(E)} \) will be properly calibrated?

3. Can the student outperform the teacher? Stable predictors likely make mistakes—indeed, this is the motivation for trying to improve them. Is it possible to correct these mistakes with \( X_U \)? Is it possible to learn an unstable “student” predictor that outperforms its own supervision signal or “teacher”?

3. Theory: When is it possible without labels?

Suppose we have already identified a stable feature \( X_S \) and a potentially unstable feature \( X_U \). We now analyze how to use \( X_S \) to leverage \( X_U \) without labels in the test domain. To do so, we first state a population-level model of our domain generalization setup. Let \( E \) be a random variable denoting the environment. Given environment \( E \), the stable feature \( X_S \), the unstable feature \( X_U \), and the label \( Y \) are distributed according to \( P_{X_S, X_U, Y|E} \). We can now formalize the three key assumptions underlying our approach:

**Definition 3.1 (Stable and Unstable Predictors).** \( X_S \) is a stable predictor of \( Y \) if \( P_{Y|X_S} \) does not depend on \( E \); equivalently, if \( Y \perp \!\!\!\!\perp E | X_S \). Conversely, \( X_U \) is an unstable predictor of \( Y \) if \( P_{Y|X_U} \) depends on \( E \); equivalently, if \( Y \not\!\!\!\!\perp \!\!\!\!\perp E | X_U \).

**Definition 3.2 (Complementary Features).** \( X_S \) and \( X_U \) are complementary predictors of \( Y \) if \( X_S \perp \!\!\!\!\perp X_U \mid (Y, E) \); i.e., redundant information in \( X_S \) and \( X_U \) comes only from \( (Y, E) \).

**Definition 3.3 (Informative Stable Predictor).** \( X_S \) is said to be informative of \( Y \) in environment \( E \) if \( X \not\!\!\!\!\perp \!\!\!\!\perp Y \mid E \) (i.e., \( X_S \) is predictive of \( Y \) within the environment \( E \)).

We discuss the roles of these assumptions after stating our main result (Thm. 3.4) that uses them. To keep our results general, we avoid assumptions on the underlying causal generative model. However, our conditional (in)dependence assumptions can be interpreted as constraints on such a causal model. Appendix D.1 characterizes the models that are consistent with our assumptions, and shows how they generalize those of prior works (Rojas-Carulla et al., 2018; von Kügelgen et al., 2019; Jiang and Veitch, 2022).

**Simplified notation.** By Defn. 3.1, we have the same stable relationship \( P_{Y|X_S, E} = P_{Y|X_S} \) in training and test domains. Now, suppose we have used the training data to learn this stable relationship and thus know \( P_{Y|X_S} \). Also suppose that we have enough unlabeled data from test domain \( E \) to learn \( P_{X_S, X_U|E} \), and recall that our goal is to predict \( Y \) from \( (X_S, X_U) \) in test domain \( E \). Since the rest of our discussion is conditioned on \( E \) being the test domain, we omit \( E \) from the notation. We further simplify notation by assuming a binary label \( Y \), deferring the multi-class case to Appendix C.

**Main Result.** We now present our main result which shows how to reconstruct \( P_{Y|X_S, X_U} \) from \( P_{Y|X_S} \) and \( P_{X_S, X_U} \) when \( X_S \) and \( X_U \) are complementary and \( X_S \) is informative.

**Theorem 3.4 (Solution to the marginal problem with binary labels and complementary features).** Consider three random variables \( X_S, X_U, \) and \( Y \), where (i) \( Y \) is binary \( \{0,1\}-valued \), (ii) \( X_S \) and \( X_U \) are complementary features (i.e., \( X_S \perp \!\!\!\!\perp X_U \mid Y \)), and (iii) \( X_S \) is informative of \( Y \) (\( X_S \not\!\!\!\!\perp \!\!\!\!\perp Y \)). Suppose \( \hat{Y} \mid X_S \sim \text{Bernoulli} (\Pr [Y = 1 | X_S] ) \) is a pseudo-label, and \( e_0 := \Pr [Y = 0 | Y = 0] \) and are the conditional probabilities that \( \hat{Y} \) and \( Y \) agree, given \( Y = 0 \) and \( Y = 1 \), respectively. Then, we have \( e_0 + e_1 > 1 \).

\[
\Pr [Y = 1 | X_U] = \frac{\Pr [\hat{Y} = 1 | X_U] + e_0 - 1}{e_0 + e_1 - 1}, \tag{3.1}
\]

\[
\Pr [Y = 1 | X_S, X_U] = \sigma (\logit (\Pr [Y = 1 | X_S]) + \logit (\Pr [Y = 1 | X_U]) - \logit (\Pr [Y = 1])) . \tag{3.2}
\]

Intuitively, suppose we train a model to predict a pseudo-label \( \hat{Y} \) (based on feature \( X_S \)) from feature \( X_U \). Assuming \( X_S \) and \( X_U \) are complementary, Eq. (3.1) shows how to transform this into a prediction of the true label \( Y \), correcting for biases caused by possible disagreement between \( \hat{Y} \) and \( Y \). Meanwhile, Eq. (3.2) integrates predictions based on \( X_S \) and \( X_U \), accounting for redundancy in the two predictions.

**Complementarity.** The assumption \( X_S \perp \!\!\!\!\perp X_U \mid Y \) plays two separate but equally crucial roles in Thm. 3.4. First, it ensures that \( X_S \) and \( X_U \) only share information about \( Y \), or, graphically, that the only unblocked path between \( X_S \) and \( X_U \) goes through \( Y \). Thus, when we train a model to predict \( \hat{Y} \) (a function of \( X_S \) only) from \( X_U \), the model must use information about \( Y \)—since there are no other relationships between \( X_S \) and \( X_U \). This insight is key to justifying the bias-correction formula of Eq. (3.1). Second, by ensuring that the only interaction between \( X_S \) and \( X_U \) is due to \( Y \) itself, complementarity implies that \( P_{Y|X_S, X_U} \) decomposes into separately estimatable \( P_{Y|X_S} \) and \( P_{Y|X_U} \). Specifically, as shown in Eq. (3.2), one can simply add estimates of \( P_{Y|X_S} \) and \( P_{Y|X_U} \) (in logit-space) while subtracting a correction-term based on the marginal distribution of \( Y \).

**Informativeness.** It is intuitive that \( X_S \not\!\!\!\!\perp \!\!\!\!\perp Y \) is necessary for pseudo-labels to be useful. More surprising is that \( X_S \not\!\!\!\!\perp \!\!\!\!\perp Y \) is sufficient for Thm. 3.4: any dependence between \( X_S \) and \( Y \) allows us to fully learn the relationship between \( X_U \) and \( Y \), affirmatively answering our question from § 2: Can the student outperform the teacher? A strong relationship between \( X_S \) and \( Y \) is still helpful in terms of the (unlabeled) sample complexity of learning \( P_{Y|X_U} \), but it is not required for consistency (Thm. 3.5, below).

**Provably consistent adaptation.** Thm. 3.4 implies that, given \( P_{Y|X_S} \) from the training domains, we can learn...
Theorem 3.5 (Consistency Guarantee, Informal). Assume (i) \(X_S\) is stable, (ii) \(X_S\) and \(X_U\) are complementary, and (iii) \(X_S\) is informative of \(Y\) in the test domain. If \(\hat{y}_{U,n} \to \Pr[Y = 1|X_U]\) as \(n \to \infty\), then \(\hat{y}_n \to \Pr[Y = 1|X_S, X_U]\).

In words, as the amount of unlabeled test-domain data increases, if the regressor on line 3 of Alg. 1 learns to predict the pseudo-label \(\hat{Y}\), then the test-domain classifier output by Alg. 1 learns to predict the true label \(Y\) in the test domain.

4. Algorithm: Stable Feature Boosting

We now use our theoretical insights from §3 to propose Stable Feature Boosting (SFB): an algorithm for reliably harnessing unstable features without labels.

Learning goals. §3 showed that, if we can indeed learn informative stable features \(X_S\) and complementary features \(X_C\), then we can employ the bias-corrected adaptation algorithm of Alg. 1 (or Alg. 2 for multi-class) to update \(h_{U}^{*}\) in the test domain. Thus, our training-domain goal is to extract \(X_S\) and \(X_C\) from the observed \(X\) such that we can reliably harness \(X_C\) in the test domain. More precisely, using the notation of Eq. (2.5), we have the following learning goals:

1. \(f_S\): stable, well-calibrated, good performance.
2. \(f_{U}\): boosts the performance of \(f_S\) in domain \(e\) using complementary features.

Objective function. To achieve the above learning goals on the training domains, we propose the following objective:

\[
\text{min}_{\phi,h_{U},\epsilon\in\mathcal{C}} \left\{ R'(\sigma \circ h_{S} \circ \Phi_{S}) + R'(\sigma \circ ((h_{S} \circ \Phi_{S}) + (h_{U}^{*} \circ \Phi_{U}))) + \lambda_{S} \cdot P_{\text{Stability}}(h_{S}, \Phi_{S}, R') + 
\lambda_{C} \cdot P_{\text{Complem.}}(\Phi_{S}(X^{e}), \Phi_{U}(X^{e})) \right\}
\]  

Here, \(P_{\text{Stability}}\) is a penalty encouraging stability while \(P_{\text{Complem.}}\) is a penalty encouraging complementarity, i.e., \(\Phi_{S}(X) \perp \Phi_{S}(X)|Y\). Several approaches have been proposed for enforcing stability, e.g., IRM (Arjovsky et al., 2020), while complementarity can be enforced by any conditional-dependence penalty (e.g., the conditional Hilbert-Schmidt Independence Criterion (Gretton et al., 2005, HSIC) or cheaper approximations like (Jiang and Veitch, 2022, §3.1)). Both \(\lambda_{S} \in [0, \infty)\) and \(\lambda_{C} \in [0, \infty)\) are regularization hyperparameters. While another hyperparameter \(\gamma \in [0, 1]\) could control the relative weighting of stable and joint risks, i.e., \(\gamma R'(h_{S} \circ \Phi_{S}) \) and \((1-\gamma) R'((h_{S} \circ \Phi_{S}) + (h_{U}^{*} \circ \Phi_{U}))\), we found this to be unnecessary in practice.

Post-hoc calibration. As discussed in §3, correctly combining the stable and unstable predictions requires them to be properly calibrated. Thus, after optimizing Eq. (4.1), we apply a standard post-processing step to improve the stable predictor’s calibration, e.g., simple temperature scaling (Guo et al., 2017).

Adapting without labels. Armed with a stable, well-calibrated \(f_S\) and complementary \(\Phi_U(X)\), we apply Alg. 1 (or Alg. 2 for the multi-class case) to arrive at an adapted joint classifier \(\hat{f}^{JT}\) (the logit of \(\hat{y}_n\) in Line 7 of Alg. 1).

5. Experiments

Implementation and dataset details are in Apps. G and E, respectively. Further results are in App. F, including synthetic (F.1) and real-world (F.3) datasets, as well as ablations (F.2).

CMNIST. Fig. 2 shows that: (i) both bias-correction (BC) and post-hoc calibration (CA) improve SFB’s adaptation performance; and (ii) without labels, SFB harnesses color neural-optimally in test domains of varying color-label correlation—
the original goal we set out to achieve (see Fig. 1a). In addition, Table 4 of App. F.2 shows that: (i) SFB learns a stable predictor with performance comparable to other invariant-prediction methods; and (ii) only SFB is capable of harnessing the spurious color feature in the test domain without labels, leading to a near-optimal boost in performance. Further results and ablations are provided in App. F.2.

PACS. Table 2 shows that SFB’s stable (i.e., no adaptation) performance is comparable to I RM and ACTIR. One exception is the severe shift of domain S (sketch), where our stable predictor performs best. Another lies with domains A and C, where ACTIR performs better. Most notable, however, is: (i) the consistent boost that SFB gets from adaptation; and (ii) SFB performing best or joint-best on 3 of 4 domains.

References


---

**Table 2. PACS test-domain accuracies over 5 seeds.**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P</th>
<th>A</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>93.0 ± 0.7</td>
<td>79.3 ± 0.5</td>
<td>74.3 ± 0.7</td>
<td>65.4 ± 1.5</td>
</tr>
<tr>
<td>IRM</td>
<td>93.3 ± 0.3</td>
<td>78.7 ± 0.7</td>
<td>75.4 ± 1.5</td>
<td>65.6 ± 2.5</td>
</tr>
<tr>
<td>ACTIR</td>
<td>94.8 ± 0.1</td>
<td>82.5 ± 0.4</td>
<td>76.6 ± 0.6</td>
<td>62.1 ± 1.3</td>
</tr>
<tr>
<td>SFB w/o adapt</td>
<td>93.7 ± 0.6</td>
<td>78.1 ± 1.1</td>
<td>73.7 ± 0.6</td>
<td>69.7 ± 2.3</td>
</tr>
<tr>
<td>SFB w. adapt</td>
<td>95.8 ± 0.6</td>
<td>80.4 ± 1.3</td>
<td>76.6 ± 0.6</td>
<td>71.8 ± 2.0</td>
</tr>
</tbody>
</table>


# Appendices

## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Proof and further discussion of Theorem 3.4</td>
<td>18</td>
</tr>
<tr>
<td>A.1</td>
<td>Proof of Theorem 3.4</td>
<td>18</td>
</tr>
<tr>
<td>A.2</td>
<td>Further discussion of Theorem 3.4</td>
<td>20</td>
</tr>
<tr>
<td>B</td>
<td>Proof of Theorem 3.5</td>
<td>21</td>
</tr>
<tr>
<td>C</td>
<td>Multiclass Case</td>
<td>24</td>
</tr>
<tr>
<td>D</td>
<td>Supplementary Results</td>
<td>26</td>
</tr>
<tr>
<td>D.1</td>
<td>Causal Perspectives</td>
<td>26</td>
</tr>
<tr>
<td>E</td>
<td>Datasets</td>
<td>28</td>
</tr>
<tr>
<td>F</td>
<td>Further Experiments</td>
<td>29</td>
</tr>
<tr>
<td>F.1</td>
<td>ColorMNIST</td>
<td>29</td>
</tr>
<tr>
<td>F.2</td>
<td>Camelyon17</td>
<td>31</td>
</tr>
<tr>
<td>G</td>
<td>Implementation Details</td>
<td>31</td>
</tr>
<tr>
<td>G.1</td>
<td>Synthetic</td>
<td>31</td>
</tr>
<tr>
<td>G.2</td>
<td>ColorMNIST</td>
<td>32</td>
</tr>
<tr>
<td>G.3</td>
<td>PACS</td>
<td>32</td>
</tr>
<tr>
<td>G.4</td>
<td>Camelyon17</td>
<td>33</td>
</tr>
<tr>
<td>H</td>
<td>Further Related Work</td>
<td>33</td>
</tr>
</tbody>
</table>
A. Proof and further discussion of Theorem 3.4

A.1. Proof of Theorem 3.4

In this section, we prove our main results regarding the marginal generalization problem presented in Section 3, namely Theorem 3.4. For the reader’s convenience, we restate Theorem 3.4 here:

**Theorem 3.4** (Marginal generalization with for binary labels and complementary features). Consider three random variables $X_S$, $X_U$, and $Y$, where

1. $Y$ is binary ($\{0, 1\}$-valued),
2. $X_S$ and $X_U$ are complementary features for $Y$ (i.e., $X_S \perp \!\!\!\!\!\perp X_U|Y$), and
3. $X_S$ is informative of $Y$ ($X_S \not\perp \!\!\!\!\!\perp Y$).

Then, the joint distribution of $(X_S, X_U, Y)$ can be written in terms of the joint distributions of $(X_S, Y)$ and $(X_S, X_U)$. Specifically, if $\hat{Y}|X_S \sim \text{Bernoulli}(\Pr[Y = 1|X_S])$ is pseudo-label and

$$e_0 := \Pr[\hat{Y} = 0|Y = 0] \quad \text{and} \quad e_1 := \Pr[\hat{Y} = 1|Y = 1]$$

are the conditional probabilities that $\hat{Y}$ and $Y$ agree, given $Y = 0$ and $Y = 1$, respectively, then,

1. $e_0 + e_1 > 1$,
2. $\Pr[Y = 1|X_U] = \frac{\Pr[\hat{Y} = 1|X_U] + e_0 - 1}{e_0 + e_1 - 1}$, and
3. $\Pr[Y = 1|X_S, X_U] = \sigma(\logit(\Pr[Y = 1|X_S]) + \logit(\Pr[Y = 1|X_U]) - \logit(\Pr[Y = 1]))$.

Before proving Theorem 3.4, we provide some examples demonstrating that the complements and informativeness assumptions in Theorem 3.4 cannot be dropped.

**Example A.1.** Suppose $X_S$ and $X_U$ have independent Bernoulli($1/2$) distributions. Then, $X_S$ is informative of both of the binary variables $Y_1 = X_S X_U$ and $Y_2 = X_S (1 - X_U)$ and both have identical conditional distributions given $X_S$, but $Y_1$ and $Y_2$ have different conditional distributions given $X_U$:

$$\Pr[Y_1 = 1|X_U = 0] = 0 \neq 1/2 = \Pr[Y_2 = 1|X_U = 0].$$

Thus, the complementarity condition cannot be omitted.

On the other hand, $X_S$ and $X_U$ are complementary for both $Y_3 = X_U$ and an independent $Y_4 \sim \text{Bernoulli}(1/2)$ and both $Y_3$ and $Y_4$ both have identical conditional distributions given $X_S$, but $Y_1$ and $Y_2$ have different conditional distributions given $X_U$:

$$\Pr[Y_3 = 1|X_U = 1] = 1/2 \neq 1 = \Pr[Y_4 = 1|X_U = 1].$$

Thus, the informativeness condition cannot be omitted.

Before proving Theorem 3.4, we prove Lemma A.2, which allows us to safely divide by the quantity $e_0 + e_1 - 1$ in the formula for $\Pr[Y = 1|X_U]$, under the condition that $X_S$ is informative of $Y$.

**Lemma A.2.** In the setting of Theorem 3.4, let $e_0$ and $e_1$ be the class-wise pseudo-label accuracies defined in as in Eq. (A.1). Then, $e_0 + e_1 = 1$ if and only if $X_S$ and $Y$ are independent.

Note that the entire result also holds, with almost identical proof, in the multi-environment setting of Sections 2 and 4, conditioned on a particular environment $E$.

**Proof.** We first prove the forwards implication. Suppose $e_0 + e_1 = 1$. If $\Pr[Y = 1] \in \{0, 1\}$, then $X_S$ and $Y$ are trivially independent, so we may assume $\Pr[Y = 1] \in (0, 1)$. Then,

$$\mathbb{E}[\hat{Y}] = e_1 \Pr[Y = 1] + (1 - e_0)(1 - \Pr[Y = 1]) \quad \text{(Law of Total Expectation)}$$

$$= (e_0 + e_1 - 1) \Pr[Y = 1] + 1 - e_0$$

$$= 1 - e_0$$

$$= \mathbb{E}[\hat{Y}|Y = 0].$$

**(e_0 + e_1 = 1)**

**(Definition of e_0)**
We now use Lemma A.2 to prove Theorem 3.4:

\[
0 = \mathbb{E}[(Y - \mathbb{E}[Y])(\hat{Y} - \mathbb{E}[\hat{Y}])]
= \mathbb{E}[\mathbb{E}[(Y - \mathbb{E}[Y])(\hat{Y} - \mathbb{E}[\hat{Y}])|X_S]]
= \mathbb{E}[\mathbb{E}[Y - \mathbb{E}[Y]|X_S]\mathbb{E}[\hat{Y} - \mathbb{E}[\hat{Y}]|X_S]]
= \mathbb{E}[\mathbb{E}[Y - \mathbb{E}[Y]|X_S]^2],
\]

where the final equality holds because \( \hat{Y} \) and \( Y \) have identical conditional distributions given \( X_S \). Since the \( L_2 \) norm of a random variable is 0 if and only if the variable is 0 almost surely, it follows that, \( P_{X_S} \)-almost surely,

\[
0 = \mathbb{E}[Y - \mathbb{E}[Y]|X_S] = \mathbb{E}[Y|X_S] - \mathbb{E}[Y],
\]

so that \( \mathbb{E}[Y|X_S] \perp \perp X_S \). Since \( Y \) is binary, its distribution is specified entirely by its mean, and so \( Y \perp \perp X_S \), proving the forwards implication.

To prove the reverse implication, suppose \( X_S \) and \( Y \) are independent. Then \( \hat{Y} \) and \( Y \) are also independent. Hence,

\[
e_1 = \mathbb{E}[\hat{Y}|Y = 1] = \mathbb{E}[\hat{Y}|Y = 0] = 1 - e_0,
\]

so that \( e_0 + e_1 = 1 \).

We now use Lemma A.2 to prove Theorem 3.4:

**Proof.** To begin, note that \( \hat{Y} \) has the same conditional distribution given \( X_S \) as \( Y \) (i.e., \( P_{\hat{Y}|X_S} = P_{Y|X_S} \) and that \( \hat{Y} \) is conditionally independent of \( Y \) given \( X_S \) (\( \hat{Y} \perp \perp Y|X_S \)). Then, since

\[
\text{Pr}[\hat{Y} = 1] = \mathbb{E}[\text{Pr}[Y = 1|X_S]] = \text{Pr}[Y = 1],
\]

we have

\[
e_1 = \text{Pr}[\hat{Y} = 1|Y = 1] = \frac{\text{Pr}[Y = 1, \hat{Y} = 1]}{\text{Pr}[Y = 1]} \quad \text{(Definition of } e_1) \]

\[
= \frac{\text{Pr}[Y = 1]}{\text{Pr}[\hat{Y} = 1]} \quad \text{(Eq. (A.2))}
\]

\[
= \frac{\mathbb{E}_{X_S}[\text{Pr}[Y = 1, \hat{Y} = 1|X_S]]}{\mathbb{E}_{X_S}[\text{Pr}[\hat{Y} = 1|X_S]]} \quad \text{(Law of Total Expectation)}
\]

\[
= \frac{\mathbb{E}_{X_S}[\text{Pr}[Y = 1|X_S]\text{Pr}[\hat{Y} = 1|X_S]]}{\mathbb{E}_{X_S}[\text{Pr}[\hat{Y} = 1|X_S]]} \quad \text{ (} \hat{Y} \perp \perp Y|X_S \text{)}
\]

\[
= \frac{\mathbb{E}_{X_S}[(\text{Pr}[Y = 1|X_S])]^2}{\mathbb{E}_{X_S}[\text{Pr}[Y = 1|X_S]]} \quad \text{(} P_{\hat{Y}|X_S} = P_{Y|X_S} \text{)}
\]

entirely in terms of the conditional distribution \( P_{Y|X_S} \) and the marginal distribution \( P_{X_S} \). Similarly, \( e_0 \) can be written as \( e_0 = \frac{\mathbb{E}_{X_S}[(\text{Pr}[Y = 0|X_S])]^2}{\mathbb{E}_{X_S}[\text{Pr}[Y = 0|X_S]]} \). Meanwhile, by the law of total expectation, and the assumption that \( X_S \) (and hence \( \hat{Y} \)) is conditionally independent of \( X_U \) given \( Y \), the conditional distribution \( P_{\hat{Y}|X_U} \) of \( \hat{Y} \) given \( X_U \) can be written as

\[
\text{Pr}[\hat{Y} = 1|X_U] = \text{Pr}[\hat{Y} = 1|Y = 0, X_U]\text{Pr}[Y = 0|X_U] + \text{Pr}[\hat{Y} = 1|Y = 1, X_U]\text{Pr}[Y = 1|X_U]
= \text{Pr}[\hat{Y} = 1|Y = 0]\text{Pr}[Y = 0|X_U] + \text{Pr}[\hat{Y} = 1|Y = 1]\text{Pr}[Y = 1|X_U]
= (1 - e_0)(1 - \text{Pr}[Y = 1|X_U]) + e_1\text{Pr}[Y = 1|X_U] = e_0 + e_1 - 1 \text{Pr}[Y = 1|X_U] + 1 - e_0.
\]
By Lemma A.2, the assumption \( X_S \not\perp Y \) implies \( \epsilon_0 + \epsilon_1 \neq 1 \). Hence, re-arranging the above equality gives us the conditional distribution \( P_{Y|X_U} \) of \( Y \) given \( X_U \) purely in terms of the conditional \( P_{Y|X_S} \) and \( P_{X_S,X_U} \):

\[
\Pr[Y = 1|X_U = X_U] = \frac{\Pr[\hat{Y} = 1|X_U = X_U] + \epsilon_0 - 1}{\epsilon_0 + \epsilon_1 - 1}.
\]

It remains now to write the conditional distribution \( P_{Y|X_S,X_U} \) in terms of the conditional distributions \( P_{Y|X_S} \) and \( P_{Y|X_U} \) and the marginal \( P_Y \). Note that

\[
\begin{align*}
\Pr[Y = 1|X_S, X_U] &= \frac{\Pr[X_S, X_U|Y = 1] \Pr[Y = 1]}{\Pr[X_S, X_U|Y = 0] \Pr[Y = 0]} \quad \text{(Bayes' Rule)} \\
&= \frac{\Pr[X_S|Y = 1] \Pr[X_U|Y = 1] \Pr[Y = 1]}{\Pr[X_S|Y = 0] \Pr[X_U|Y = 0] \Pr[Y = 0]} \quad \text{(Complementarity)} \\
&= \frac{\Pr[Y = 1|X_S] \Pr[Y = 1|X_U] \Pr[Y = 0]}{\Pr[Y = 0|X_S] \Pr[Y = 0|X_U] \Pr[Y = 1]} \quad \text{(Bayes' Rule)}.
\end{align*}
\]

It follows that the logit of \( \Pr[Y = 1|X_S, X_U] \) can be written as the sum of a term depending only on \( X_S \), a term depending only on \( X_U \), and a constant term:

\[
\begin{align*}
\logit (\Pr[Y = 1|X_S, X_U]) &= \log \frac{\Pr[Y = 1|X_S, X_U]}{1 - \Pr[Y = 1|X_S, X_U]} \\
&= \log \frac{\Pr[Y = 1|X_S, X_U]}{\Pr[Y = 0|X_S, X_U]} \\
&= \log \Pr[Y = 1|X_S] + \log \Pr[Y = 1|X_U] - \log \Pr[Y = 0] \\
&= \logit (\Pr[Y = 1|X_S]) + \logit (\Pr[Y = 1|X_U]) - \logit (\Pr[Y = 1]).
\end{align*}
\]

Since the sigmoid \( \sigma \) is the inverse of logit,

\[
\Pr[Y = 1|X_S, X_U] = \sigma(\logit (\Pr[Y = 1|X_S]) + \logit (\Pr[Y = 1|X_U]) - \logit (\Pr[Y = 1])),
\]

which, by Eq. (3.1), can be written in terms of the conditional distribution \( P_{Y|X_S} \) and the joint distribution \( P_{X_S,X_U} \). \( \square \)

### A.2. Further discussion of Theorem 3.4

**Connections to learning from noisy labels.** Theorem 3.4 leverages two theoretical insights about the special structure of pseudo-labels that complement results in the literature on learning from noisy labels. First, Blanchard et al. (2016) showed that learning from noisy labels is possible if and only if the total noise level is below the critical threshold \( \epsilon_0 + \epsilon_1 > 1 \); in the case of learning from pseudo-labels, we show (see Lemma A.2 in Appendix A.1) that this is satisfied if and only if \( X_S \) is informative of \( Y \) (i.e., \( Y \not\perp X_S \)). Second, methods for learning under label noise commonly assume knowledge of \( \epsilon_0 \) and \( \epsilon_1 \) (Natarajan et al., 2013), which is unrealistic in many applications; however, for pseudo-labels sampled from a known conditional probability distribution \( P_{Y|X_S} \), one can express these noise levels we show (as part of Theorem 3.4) that the class-conditional noise levels can be easily estimated.

**Possible applications of Theorem 3.4 beyond domain adaptation** The reason we wrote Theorem 3.4 in the more general setting of the marginal problem rather than in the specific context of domain adaptation is that we envision possible applications to a number of problems besides domain adaptation. For example, suppose that, after learning a calibrated machine learning model \( M_1 \) using a feature \( X_S \), we observe an additional feature \( X_U \). In the case that \( X_S \) and \( X_U \) are complementary, Theorem 3.4 justifies using the student-teacher paradigm (Buciluă et al., 2006; Ba and Caruana, 2014; Hinton et al., 2015) to train a model for predicting \( Y \) from \( X_U \) (or from \( (X_S, X_U) \) jointly) based on predictions from \( M_1 \). This could be useful if we don’t have access to labeled pairs \((X_U, Y)\), or if retraining a model using \( X_S \) would require substantial computational resources or access to sensitive or private data. Exploring such approaches could be a fruitful direction for future work.
B. Proof of Theorem 3.5

This appendix provides a proof of Theorem 3.5, which provides conditions under which our proposed domain adaptation procedure (Alg. 1) is consistent.

We first provide a formal version of Theorem 3.5:

**Theorem 3.5 (Consistency of the bias-corrected classifier).** Assume

1. $X_S$ is stable,
2. $X_S$ and $X_U$ are complementary, and
3. $X_S$ is informative of $Y$ (i.e., $X_S \perp \perp Y$).

Let $\hat{h}_n : X_S \times X_U \to [0,1]$ given by

$$
\hat{h}_n(x_S, x_U) = \sigma \left( f_S(x_S) + \logit \left( \frac{\hat{\eta}_{U,n}(x_U) + \hat{\epsilon}_{0,n} - 1}{\hat{\epsilon}_{0,n} + \hat{\epsilon}_{1,n} - 1} \right) \right), \text{ for all } (x_S, x_U) \in X_S \times X_U,
$$

denote the bias-corrected regression function estimate proposed in Alg. 1, and let $\hat{h}_n : X_S \times X_U \to \{0,1\}$ given by

$$
\hat{h}_n(x_S, x_U) = 1\{\eta(x_S, x_U) > 1/2\}, \text{ for all } (x_S, x_U) \in X_S \times X_U,
$$

denote the corresponding hard classifier. Let $\eta_U : X_U \to [0,1]$, given by $\eta_U(x_U) = \Pr[Y = 1|X_U = x_U, E = 1]$ for all $x_U \in X_U$, denote the true regression function over $X_U$, and let $\hat{\eta}_{U,n}$ denote its estimate as assumed in Line 3 of Alg. 1. Then, as $n \to \infty$,

(a) if, for $P_{X_U}$-almost all $x_U \in X_U$, $\hat{\eta}_{U,n}(x_U) \to \eta_U(x_U)$ in probability, then $\hat{h}_n$ and $\hat{h}_n$ are weakly consistent (i.e., $\hat{h}_n(x_S, x_U) \to \eta(x_S, x_U)$ $P_{X_S,X_U}$-almost surely and $R(\hat{h}_n) \to R(h^*)$ in probability).

(b) if, for $P_{X_U}$-almost all $x_U \in X_U$, $\hat{\eta}_{U,n}(x_U) \to \eta_U(x_U)$ almost surely, then $\hat{h}_n$ and $\hat{h}_n$ are strongly consistent (i.e., $\hat{h}_n(x_S, x_U) \to \eta(x_S, x_U)$ $P_{X_S,X_U}$-almost surely and $R(\hat{h}_n) \to R(h^*)$ a.s.).

Before proving Theorem 3.5, we provide a few technical lemmas. The first shows that almost-everywhere convergence of regression functions implies convergence of the corresponding classifiers in classification risk:

**Lemma B.1.** Consider a sequence of regression functions $\eta, \eta_1, \eta_2, \ldots : X \to [0,1]$. Let $h, h_1, h_2, \ldots : X \to \{0,1\}$ denote the corresponding classifiers

$$
h(x) = 1\{\eta(x) > 1/2\} \quad \text{and} \quad h_i(x) = 1\{\eta_i(x) > 1/2\}, \text{ for all } i \in \mathbb{N}, x \in X.
$$

(a) If $\eta_n(x) \to \eta(x)$ for $P_X$-almost all $x \in X$ in probability, then $R(h_n) \to R(h^*)$ in probability.

(b) If $\eta_n(x) \to \eta(x)$ for $P_X$-almost all $x \in X$ almost surely as $n \to \infty$, then $R(h_n) \to R(h)$ almost surely.

**Proof.** Note that, since $h_n(x) \neq h(x)$ implies $|\eta_n(x) - \eta(x)| \geq |\eta(x) - 1/2|$, 

$$1\{h_n(x) \neq h(x)\} \leq 1\{|\eta_n(x) - \eta(x)| \geq |\eta(x) - 1/2|\}. \quad \text{(B.1)}
$$

We utilize this observation to prove both (a) and (b).

**Proof of (a)** Let $\delta > 0$. By Inequality (B.1) and partitioning $X$ based on whether $|2\eta(X) - 1| \leq \delta/2$,

$$
\begin{align*}
\mathbb{E}_X [2\eta(X) - 1|1\{h_n(X) \neq h(X)\}] &\leq \mathbb{E}_X [2\eta(X) - 1|1\{|\eta_n(X) - \eta(X)| \geq |\eta(x) - 1/2|\}] \\
&= \mathbb{E}_X [2\eta(X) - 1|1\{|\eta_n(X) - \eta(X)| \geq |\eta(x) - 1/2|\}1\{|2\eta(X) - 1| \leq \delta/2\}] + \mathbb{E}_X [2\eta(X) - 1|1\{|\eta_n(X) - \eta(X)| \geq |\eta(x) - 1/2|\}1\{|2\eta(X) - 1| > \delta/2\}] \\
&\leq \mathbb{E}_X [1\{|\eta_n(X) - \eta(X)| \geq |\eta(x) - 1/2|\}1\{|2\eta(X) - 1| \leq \delta/2\}] + \delta/2.
\end{align*}
$$

**Proof of (b)** Let $\delta > 0$. By Inequality (B.1) and partitioning $X$ based on whether $|2\eta(X) - 1| \leq \delta/2$, 

$$
\begin{align*}
\mathbb{E}_X [2\eta(X) - 1|1\{h_n(X) \neq h(X)\}] &\leq \mathbb{E}_X [2\eta(X) - 1|1\{|\eta_n(X) - \eta(X)| \geq |\eta(x) - 1/2|\}] \\
&= \mathbb{E}_X [2\eta(X) - 1|1\{|\eta_n(X) - \eta(X)| \geq |\eta(x) - 1/2|\}1\{|2\eta(X) - 1| \leq \delta/2\}] + \mathbb{E}_X [2\eta(X) - 1|1\{|\eta_n(X) - \eta(X)| \geq |\eta(x) - 1/2|\}1\{|2\eta(X) - 1| > \delta/2\}] \\
&\leq \mathbb{E}_X [1\{|\eta_n(X) - \eta(X)| \geq |\eta(x) - 1/2|\}1\{|2\eta(X) - 1| \leq \delta/2\}] + \delta/2.
\end{align*}
$$
Hence,

\[
\lim_{n \to \infty} \Pr_{\eta_n} \left[ |2\eta(X) - 1|1\{h_n(X) \neq h(X)\} > \delta \right] \\
\leq \lim_{n \to \infty} \Pr_{\eta_n} \left[ |\eta_n(X) - \eta(X)| > \delta/2 \right] > \delta/2 \\
= \frac{2}{\delta} \mathbb{E}_{\eta_n} \left[ \lim_{n \to \infty} \Pr_{\eta_n} (|\eta_n(X) - \eta(X)| > \delta/2) \right] \\
= \frac{2}{\delta} \mathbb{E}_{\eta} \left[ \lim_{n \to \infty} \Pr_{\eta_n} (|\eta_n(X) - \eta(X)| > \delta/2) \right].
\]

(Markov’s Inequality)

Proof of (b) For any \( x \in \mathcal{X} \) with \( \eta(x) \neq 1/2 \), if \( \eta_n(x) \to \eta(x) \) then \( 1\{ |\eta_n(x) - \eta(x)| \geq |\eta(x) - 1/2| \} \to 0 \). Hence, by Inequality (B.1), the dominated convergence theorem with \( |2\eta(x) - 1|1\{ |\eta_n(x) - \eta(x)| \geq |\eta(x) - 1/2| \} \leq 1 \), and the assumption that \( \eta_n(x) \to \eta(x) \) for \( P_X \)-almost all \( x \in \mathcal{X} \) almost surely,

\[
\lim_{n \to \infty} \mathbb{E}_X |2\eta(X) - 1|1\{h_n(X) \neq h(X)\} \\
\leq \lim_{n \to \infty} \mathbb{E}_X |2\eta(X) - 1|1\{ |\eta_n(X) - \eta(X)| \geq |\eta(x) - 1/2| \} \\
= \mathbb{E}_X \left[ \lim_{n \to \infty} |2\eta(X) - 1|1\{ |\eta_n(X) - \eta(x)| \geq |\eta(x) - 1/2| \} \right] \\
= 0, \quad \text{almost surely.}
\]

Our next lemma concerns an edge case in which the features \( X_S \) and \( X_U \) provide perfect but contradictory information about \( Y \), leading to Equation (3.2) being ill defined. We show that this can happen only with probability 0 over \( (X_S, X_U) \sim P_{X_S, X_U} \) can thus be safely ignored:

Lemma B.2. Consider two predictors \( X_S \) and \( X_U \) of a binary label \( Y \). Then,

\[
\Pr_{X_S, X_U} [\mathbb{E}[Y|X_S] = 1 \text{ and } \mathbb{E}[Y|X_U] = 0] = \Pr_{X_S, X_U} [\mathbb{E}[Y|X_S] = 0 \text{ and } \mathbb{E}[Y|X_U] = 1] = 0.
\]

Proof. Suppose, for sake of contradiction, that the event

\[
A := \{(x_S, x_U) : \mathbb{E}[Y|X_S = x_S] = 1 \text{ and } \mathbb{E}[Y|X_U = x_U] = 0\}
\]

has positive probability. Then, the conditional expectation \( \mathbb{E}[Y|A] \) is well-defined, giving the contradiction

\[
1 = \mathbb{E}_{X_S} [\mathbb{E}[Y|E, X_S]] = \mathbb{E}[Y|A] = \mathbb{E}_{X_U} [\mathbb{E}[Y|E, X_U]] = 0.
\]

The case \( \mathbb{E}[Y|X_S] = 0 \) and \( \mathbb{E}[Y|X_U] = 1 \) is similar. \( \square \)

We now utilize Lemmas B.1 and B.2 to prove Theorem 3.5.

Proof. By Lemma B.1, it suffices to prove that \( \hat{\eta}(x_S, x_U) \to \eta(x_S, x_U) \), for \( P_{X_S, X_U} \)-almost all \( (x_S, x_U) \in \mathcal{X}_S \times \mathcal{X}_U \), in probability (to prove (a)) and almost surely (to prove (b)).
We now prove each of these in turn.

**Proof of (ii)**

Recall that \( \bar{\eta}(x_u) \rightarrow \eta(x_u) \), to prove both of these, it suffices to show:

(i) \( \hat{e}_0 \rightarrow e_0 \) and \( \hat{e}_1 \rightarrow e_1 \) almost surely as \( n \rightarrow \infty \).

(ii) \( \hat{\beta}_{1,n} \rightarrow \beta_1 \in (-\infty, \infty) \) almost surely as \( n \rightarrow \infty \).

(iii) The mapping \( (a, b, c) \mapsto \logit \left( \frac{a + b - 1}{b + c - 1} \right) \) is continuous at \((\hat{\eta}(x_u), e_0, e_1)\).

We now prove each of these in turn.

**Proof of (i)**

Since \( \hat{Y}_i \perp Y_i | X_S \) and \( 0 < \text{Pr}[\hat{Y} = 1] \), by the strong law of large numbers and the continuous mapping theorem,

\[
\hat{e}_1 = \frac{1}{n} \sum_{i=1}^{n} \hat{Y}_i \sigma(f_S(X_i)) = \frac{1}{n} \sum_{i=1}^{n} \hat{Y}_i \sigma(f_S(X_i)) \rightarrow \frac{\mathbb{E}[\sigma(f_S(X))1\{\hat{Y} = 1\}]}{\text{Pr}[\hat{Y} = 1]} = \frac{\text{Pr}[Y = 1]}{\text{Pr}[Y = 1]} = 1,
\]

almost surely as \( n \rightarrow \infty \). Similarly, since \( \text{Pr}[\hat{Y} = 0] = 1 - \text{Pr}[\hat{Y} = 1] > 0 \), \( \hat{e}_0 \rightarrow e_0 \) almost surely.

**Proof of (ii)**

Recall that

\[
\hat{\beta}_{1,n} = \logit \left( \frac{1}{n} \sum_{i=1}^{n} \hat{Y}_i \right).
\]

By the strong law of large numbers, \( \frac{1}{n} \sum_{i=1}^{n} \hat{Y}_i \rightarrow \text{Pr}[\hat{Y} = 1|E = 1] = \text{Pr}[Y = 1|E = 1] \). Since we assumed \( \text{Pr}[Y = 1|E = 1] \in (0, 1) \), it follows that the mapping \( a \mapsto \logit(a) \) is continuous at \( a = \text{Pr}[Y = 1|E = 1] \). Hence, by the continuous mapping theorem, \( \hat{\beta}_{1,n} \rightarrow \logit \left( \text{Pr}[Y = 1|E = 1] \right) = \beta_1 \) almost surely.

**Proof of (iii)**

Since the logit function is continuous on the open interval \((0, 1)\) and we assumed \( e_0 + e_1 > 1 \), it suffices to show that \( 0 < \eta^*(x_u) < 1 \). Since, according to Theorem 3.4,

\[
\eta(x_u) = (e_0 + e_1 - 1)\eta^*(x_u)) + 1 - e_0,
\]

this holds as long as \( 0 < \eta^*(x_u) < 1 \), as we assumed for \( P_{X_U} \)-almost all \( x_U \in \mathcal{X}_U \).

**Infinite case**

We now address the case where either \( \text{Pr}[Y|X_S = x_S] \in \{0, 1\} \) or \( \text{Pr}[Y|X_U = x_U] \in \{0, 1\} \). By Lemma B.2, only one of these can happen at once, \( P_{X_S, X_U} \)-almost surely. Hence, since \( \lim_{n \rightarrow \infty} \hat{\beta}_{1,n} \) is also finite almost surely, if \( \text{Pr}[Y|X_S = x_S] \in \{0, 1\} \), then \( \eta(x_S, x_U) = \sigma(\logit(\text{Pr}[Y|X_S = x_S])) = \eta(x_S, x_U) \), while, if \( \text{Pr}[Y|X_U = x_U] \in \{0, 1\} \), then \( \eta(x_S, x_U) \rightarrow \sigma(\logit(\text{Pr}[Y|X_U = x_U])) = \eta(x_S, x_U) \), in probability or almost surely, as appropriate.

\[\square\]

**C. Multiclass Case**

In the main paper, to simplify notation, we presented our unsupervised test-domain adaptation method in the case of binary labels \( Y \). However, in many cases, including several of our experiments in Section 5, the label \( Y \) can take more than 2 distinct values. Hence, in this section, we show how to generalize our method to the multiclass setting and then present the exact procedure (Alg. 2) used in our multiclass experiments in Section 5.
Suppose we have $K \geq 2$ classes. We “one-hot encode” these classes, so that $Y$ takes values in the set

$$
\mathcal{Y} = \{(1,0,...,0), (0,1,0,...,0), ..., (0,...,0,1)\} \subseteq \{0,1\}^K.
$$

Let $\epsilon \in [0,1]^{\mathcal{Y} \times \mathcal{Y}}$ with

$$
\epsilon_{y,y'} = \Pr[\hat{Y} = y | Y = y']
$$

denote the class-conditional confusion matrix of the pseudo-labels. Then, we have

$$
\mathbb{E}[\hat{Y}|X_U] = \sum_{y \in \mathcal{Y}} \mathbb{E}[\hat{Y}|Y = y, X_U] \Pr[Y = y|X_U] = \sum_{y \in \mathcal{Y}} \mathbb{E}[\hat{Y}|Y = y] \Pr[Y = y|X_U] = \epsilon \mathbb{E}[Y|X_U];
$$

(Law of Total Expectation)

in particular, when $\epsilon$ is invertible,

$$
\mathbb{E}[Y|X_U] = \epsilon^{-1} \mathbb{E}[\hat{Y}|X_U],
$$

giving a multiclass equivalent of Eq. (3.1) in Theorem 3.4. We also have

$$
\epsilon_{y,y'} = \Pr[\hat{Y} = y | Y = y'] = \frac{\Pr[\hat{Y} = y, Y = y']}{\Pr[Y = y']} = \frac{\mathbb{E}[\hat{Y} = y | X_S] \Pr[Y = y'|X_S]}{\mathbb{E}[Y = y'|X_S]} = \frac{\mathbb{E}[\eta_{1,y}(X_S)\eta_{1,y'}(X_S)]}{\mathbb{E}[\eta_{1,y'}(X_S)]},
$$

suggesting the estimate

$$
\hat{\epsilon}_{y,y'} = \frac{\sum_{i=1}^n \eta_{S,y}(X_{S,i})\eta_{S,y'}(X_{S,i})}{\sum_{i=1}^n \eta_{S,y'}(X_{S,i})} = \sum_{i=1}^n \frac{\hat{\eta}_{S,y}(X_{S,i})}{\sum_{i=1}^n \hat{\eta}_{S,y'}(X_{S,i})}
$$

of each $\epsilon_{y,y'}$, or, in matrix notation,

$$
\mathbf{\hat{\epsilon}} = \eta_S(X_S) \text{ Normalize}(\eta_S(X_S)),
$$

where Normalize($X$) scales each column of $X$ to sum to 1. This gives us an multiclass equivalent of Line 4 in Alg. 1.

The multiclass versions of Eq. (3.2) and Line 7 of Alg. 1 are slightly less straightforward. Specifically, whereas, in the binary case, we used the fact that $\Pr[X_S, X_U|Y \neq 1] = \Pr[X_S, X_U|Y = 0] = \Pr[X_S|Y = 0] \Pr[X_U|Y = 0] = \Pr[X_S|Y \neq 1] \Pr[X_U|Y \neq 1]$ (by complementarity), in the multiclass case, we do not have $\Pr[X_S, X_U|Y \neq 1] = \Pr[X_S|Y \neq 1] \Pr[X_U|Y \neq 1]$. However, following similar reasoning as in the proof of Theorem 3.4, we have

$$
\frac{\Pr[Y = y|X_S, X_U, E]}{\Pr[Y \neq y|X_S, X_U, E]} = \frac{\Pr[Y = y|X_S, X_U, E]}{\sum_{y' \neq y} \Pr[Y = y'|X_S, X_U, E]} = \frac{\Pr[X_S, X_U|Y = y, E] \Pr[Y = y|E]}{\sum_{y' \neq y} \Pr[Y \neq y'|X_S, X_U, E] \Pr[Y = y'|E]} = \frac{\Pr[X_S|Y = y, E] \Pr[X_U|Y = y, E] \Pr[Y = y|E]}{\sum_{y'} \Pr[X_S|Y = y', E] \Pr[X_U|Y = y', E] \Pr[Y = y'|E]} \cdot \frac{\Pr[Y = y|E]}{\Pr[Y = y'|E]} \cdot \frac{\Pr[Y = y'|E]}{\Pr[Y = y'|E]}. 
$$

(Bayes’ Rule)
Hence,
\[
\log(\Pr[Y = y|X_S, X_U, E]) = \log \left( \frac{\Pr[Y = y|X_S, E] \Pr[Y = y|X_U, E]}{\sum_{y' \neq y} \Pr[Y = y'|X_S, E] \Pr[Y = y'|X_U, E]} \cdot \frac{\Pr[Y = y|E]}{\Pr[Y = y']|E]} \right) = \log \left( \frac{C_y}{\sum_{y' \neq y} C_y'} \right) = \log \left( \frac{C_y}{\sum_{y' \neq y} C_y'} \right) = \log \left( \frac{C_y}{\|C\|_1} \right),
\]
for \(C \in \mathbb{R}^Y\) defined by
\[
C_y = \frac{\eta_S(y) \eta_U(y)}{\Pr[Y = y]} \quad \text{for each} \ y \in Y.
\]
In particular, applying the sigmoid function to each side, we have
\[
\Pr[Y|X_S, X_U] = \frac{C}{\|C\|_1}.
\]
We can estimate \(C_y\) by
\[
\hat{C}_y = \frac{\eta_S(y) \eta_U(y)}{\Pr[Y = y]} = \frac{1}{n} \sum_{i=1}^{n} \eta_S(y_i) \eta_U(y_i).
\]
In matrix notation, this is
\[
\hat{C} = \frac{\eta_S(X_S) \cdot \eta_U(X_U)}{\Pr[Y = y]} = \frac{1}{n} \sum_{i=1}^{n} \eta_S(X_{S,i}),
\]
where \(\circ\) denotes element-wise multiplication. Putting these derivations together gives us our multiclass version of Alg. 1, presented in Alg. 2, where \(\Delta^Y = \{z \in [0,1]^K : \sum_{y \in Y} z_y = 1\}\) denotes the standard probability simplex over \(Y\).

**Algorithm 2:** Multiclass bias-corrected unsupervised domain adaptation procedure.

**Input:** Regression function \(\eta_S : \mathcal{X} \rightarrow \Delta^Y\), subroutine \(\text{regressor}, n\) unlabeled samples \(\{(X_{S,i}, X_{U,i})\}_{i=1}^{n}\) from the test domain

**Output:** Estimate \(\hat{\eta}_n : \mathcal{X}_S \times \mathcal{X}_U \rightarrow \Delta^Y\) of regression function \(\eta_S(x_S, x_U) = \Pr[Y = y|X_S = x_S, X_U = x_U]\)

1. for \(i \in [n]\) do // generate pseudolabels
   2. Sample \(\hat{Y}_i \sim \text{Categorical}(\eta_S(X_{S,i}))\) // \(\hat{Y} \in \{0,1\}^{n \times K}\) is one-hot encoded
   3. \(\hat{\eta}_{U,i} \leftarrow \text{regressor}(\{(X_{U,i}, \hat{Y}_i)\}_{i=1}^{n})\) // regress pseudolabels over \(X_U\)
   4. \(\hat{c} \leftarrow \eta_S(X_S) \cdot \text{Normalize}(\eta_S(X_S))\)
   5. \(\hat{\eta}_{U,i} \leftarrow \{x_U \mapsto \max\{0, \min\{1, e^{-1} \hat{\eta}_{U,i}(x_U)\}\} \} \quad \text{// Unstable predictor}\)
6. for \(y \in [K]\) do
   7. \(C_y \leftarrow (x_S, x_U) \mapsto \frac{\eta_S(y) \circ \hat{\eta}_{U,i}(x_U)}{\frac{1}{n} \sum_{i=1}^{n} \eta_S(y_i)}\)
   8. \(\hat{\eta}_{S,U,i} \leftarrow (x_S, x_U) \mapsto \frac{C_y(x_S, x_U)}{\|C_y(x_S, x_U)\|_1} \quad \text{// Joint predictor}\)
9. return \((\hat{\eta}_{U,i}, \hat{\eta}_{S,U,i})\)

**D. Supplementary Results**

**Proposition D.1.** Suppose \(\hat{Y}|f_S(X) \sim \text{Bernoulli}(\sigma(f_S(X)))\), such that \(\hat{Y} \perp f_U(X)|f_S(X)\). Then,
\[
0 \in \arg\min_{\hat{Y} : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{E}_{f_U : \mathcal{X} \rightarrow \mathbb{R}}[\ell(\hat{Y}, \sigma(f_S(X) + f_U(X)))],
\]
where \(\ell(x, y) = -x \log y - (1 - x) \log(1 - y)\) denotes the cross-entropy loss.
Then, as illustrated in Figure 3, three types of stable features are possible:

1. Causal ancestors $X_{S,C}$ of $Y$,
2. Causal descendants $X_{S,E}$ of $Y$ that are not also descendants of $E$,  
3. Causal spouses $X_{S,S}$ of $Y$ (i.e., causal ancestors of $X_{S,E}$), and

while the only unstable features possible are descendants of $Y$.

Notable special cases of the DAG in Figure 3 include:

1. the “cause-effect” settings, studied by Rojas-Carulla et al. (2018); von Kügelgen et al. (2019), where $X_S$ is a cause of $Y$, $X_U$ is an effect of $Y$, and $E$ affects both $X_S$ and $X_U$ but affects $Y$ only through $X_S$. Note that this generalizes the commonly used “covariate shift” assumption, as not only the covariate distribution $P_{X_S,X_U}$ but also the conditional distribution $P_{Y|X_U}$ can change between environments.

2. the “anti-causal” setting, studied by Jiang and Veitch (2022), where $X_S$ and $X_U$ are both effects of $Y$, but $X_S$ is unaffected by $E$.

3. the widely studied “covariate shift” setting (Sugiyama et al., 2007; Gretton et al., 2009; Bickel et al., 2009; Sugiyama and Kawanabe, 2012), which corresponds (see Sections 3 and 5 of Schölkopf (2022)) to a causal factorization $P(X,Y) = P(X)P(Y|X)$ (i.e., in which the only stable components $X_S$ are causes $X_{S,E}$) of $Y$ or unconditionally independent (e.g., causal spouses $X_{S,S}$) of $Y$.

However, this model is more general than these special cases. Also, for sake of simplicity, we assumed causal sufficiency here; however, in the presence of unobserved confounders, other types of stable features are also possible; for example, if we consider the possibility of unobserved confounders $U$ influencing $Y$ that are independent of $E$ (i.e., invariant across domains), then our method can also utilize stable features that are descendants of $U$ (i.e., “siblings” of $Y$).

E. Datasets

**Synthetic: Anti-causal.** We consider an anti-causal synthetic dataset based on that of Jiang and Veitch (2022, §6.1) where data is generated according to the following structural equations (illustrated graphically in Fig. 4a):

$$
Y \leftarrow \text{Rad}(0.5); \\
X_S \leftarrow Y \cdot \text{Rad}(0.75); \\
X_U \leftarrow Y \cdot \text{Rad}(\beta^e),
$$

where the input $X = (X_S, X_U)$ and $\text{Rad}(\beta)$ means that a random variable is $-1$ with probability $1 - \beta$ and $+1$ with probability $\beta$. Following Jiang and Veitch (2022, §6.1), we create two training domains with $\beta_e \in \{0.95, 0.7\}$, one validation domain with $\beta_e = 0.6$ and one test domain with $\beta_e = 0.1$. The idea here is that, during training, prediction based on the stable $X_S$ results in lower accuracy (75%) than prediction based on the unstable $X_U$ (82.5%). Thus, models optimizing for prediction accuracy only—and not stability—will use $X_U$ and ultimately end up with only 10% in the test domain. Importantly, while the stable predictor achieves 75% accuracy in the test domain, performance can be improved to 90% if $X_U$ can be used correctly.

**Synthetic: Cause-effect with direct dependence.** We also consider a synthetic cause-effect dataset in which there is a direct dependence between $X_S$ and $X_U$. In particular, similar to Jiang and Veitch (2022, App. B), we generate synthetic data according to the following structural equations (illustrated graphically in Fig. 4b):

$$
X_S \leftarrow N_S, \text{ with } N_S \leftarrow \text{Bern}(0.5); \\
Y \leftarrow \text{XOR}(X_S, N_Y), \text{ with } N_Y \leftarrow \text{Bern}(0.75); \\
X_U \leftarrow \text{XOR}(\text{XOR}(Y, N_U), X_S), \text{ with } N_U \leftarrow \text{Bern}(\beta_e).
$$

Here, the input $X = (X_S, X_U)$ and $\text{Bern}(\beta)$ means that a random variable is $1$ with probability $\beta$ and $0$ with probability $1 - \beta$. Following Jiang and Veitch (2022, Appendix B), we create two training domains with $\beta_e \in \{0.95, 0.8\}$, one validation domain with $\beta_e = 0.2$, and one test domain with $\beta_e = 0.1$. Like the anti-causal synthetic dataset, the idea is that
prediction based on the stable $X_S$ results in lower accuracy (75%) than prediction based on the unstable $X_U$. Thus, models optimizing for prediction accuracy only—and not stability—will use $X_U$ and ultimately end up with only 10% accuracy in the test domain. In addition, while the stable predictor achieves 75% accuracy in the test domain, performance can be improved to 90% if $X_U$ can be used correctly. However, unlike the anti-causal synthetic dataset, the stable $X_S$ and unstable $X_U$ features are not conditionally independent, i.e., $X_U \not\perp\!
mid X_S|Y$, since $X_S$ directly influences $X_U$.

![Figure 4. Causal DAGs behind the synthetic datasets. Dashed circles indicate latent/unobserved variables, solid indicate observed.](image)

**ColorMNIST.** We consider the ColorMNIST or CMNIST dataset (Arjovsky et al., 2020). This takes the original MNIST dataset and first turns it into a binary classification task (digit in 0–4 or 5–9) and then colorizes it such that digit color (red or green) is a highly-informative but spurious feature. In particular, one first adds label noise such that, across all 3 domains, digit shape correctly determines the label with probability 0.75. Then, as depicted in Fig. 5, one colorizes the digits such that green digits generally belong to class 0 in the two training domains and generally belong to class 1 in the test domain.

**PACS.** We consider the PACS dataset (Li et al., 2017a)—a 7-class image-classification dataset consisting of 4 domains: photos (P), art (A), cartoons (C) and sketches (S), with examples shown in Fig. 5. Model performances are reported for each domain after training on the other three domains.

**Camelyon17.** We consider the Camelyon17 (Bandi et al., 2018) dataset from the WILDS benchmark (Koh et al., 2021), a medical dataset with histopathology images from 5 hospitals which use different staining and imaging techniques (see Fig. 5). The goal is to determine whether or not a given image contains tumour tissue, making it a binary classification task.

### F. Further Experiments

This appendix provides further experiments which supplement those in the main text. In particular, it provides: (i) experiments on synthetic datasets (F.1); (ii) ablations on the ColorMNIST dataset showing the effects of bias correction and post-hoc calibration (F.2); and (iii) experiments on a real-world medical dataset, namely, Camelyon17 (F.3).

#### F.1. Synthetic datasets

**F.1.1. Anti-causal**

We first consider a simple anti-causal synthetic dataset based on that of Jiang and Veitch (2022, §6.1), where our conditional independence assumption holds, i.e., $X_U \perp\!
mid X_S|Y$. The main idea is that: (i) models optimizing for accuracy only (e.g., ERM) use the unstable $X_U$ in a fixed manner and end up with only 10% in the test domain; (ii) models also optimizing for stability (e.g., IRM) use the stable $X_S$ and end up with 75% accuracy; and (iii) accuracy can be improved to 90% if $X_U$ is used correctly in the test domain. See Appendix E for details on the data-generation procedure and Appendix G for details on the experimental setup.

Table 3 shows that ERM performs poorly as it uses the unstable feature $X_U$, while IRM (Arjovsky et al., 2020), ACTIR (Jiang and Veitch, 2022) and our SFB algorithm all do well by using only the stable feature $X_S$. Critically, only SFB is capable of harnessing $X_U$ in the test domain without labels, leading to a near-optimal boost in performance.
F.1.2. CAUSE-EFFECT WITH DIRECT $X_S$-$X_U$ DEPENDENCE

Our SFB approach assumes that the harnessed unstable features $X_C \subseteq X_U$ are conditionally independent of the stable features $X_S$. If this assumption is violated, then adaptation can fail as SFB is not guaranteed to learn an asymptotically-optimal predictor in the test domain. To investigate the adaptation performance of SFB when this assumption is violated, we also consider a synthetic cause-effect dataset in which there is a direct dependence between $X_S$ and $X_U$. See Appendix E for details on the data-generation procedure and Appendix G.4 for details of the experimental setup (the same as for the anti-causal synthetic dataset).

Looking at Table 3 we see that: (i) ACTIR has poor stable/invariant performance as its notion of stability relies on the now-violated conditional-independence assumption; (ii) IRM has good stable/invariant performance as its notion of stability does not rely on conditional independence; (iii) SFB has good stable/invariant performance as its notion of stability does not rely on conditional independence (IRM’s stability penalty is used); and (iv) surprisingly, SFB has near-optimal adapted performance despite the conditional-independence assumption being violated. One explanation for (iv) is that the conditional-independence assumption is only weakly violated in the test domain. Another is that conditional independence isn’t necessary for SFB and some weaker, yet-to-be-determined condition suffices.

| Algorithm              | Anti-Causal (with $X_U \perp \perp X_S | Y$) | Cause-Effect (with $X_U \not\perp \perp X_S | Y$) |
|------------------------|-------------------------------------------|-----------------------------------------------|
| ERM                    | 9.9 ± 0.1                                 | 11.6 ± 0.7                                    |
| IRM                    | 74.9 ± 0.1                                | 69.6 ± 1.3                                    |
| ACTIR                  | 74.8 ± 0.4                                | 43.5 ± 2.6                                    |
| SFB (Ours) w/o adapt   | 74.7 ± 1.2                                | 74.9 ± 3.6                                    |
| SFB (Ours) w. adapt    | 89.2 ± 2.9                                | 88.6 ± 1.4                                    |
F.2. ColorMNIST

F.2.1. Comparison to Baselines

We now provide results on the “standard” ColorMNIST test domain, which has a color-label correlation of -0.9 (see Fig. 5 and red dot of Fig. 1a), in order to compare to the relevant baselines. As shown in Table 4: (i) SFB learns a stable predictor with performance comparable to other invariant-prediction methods; and (ii) only SFB is capable of harnessing the spurious color feature in the test domain without labels, leading to a near-optimal boost in performance. Note that “Oracle w/o adapt.” refers to an ERM model trained on grayscale images, while “Oracle w. adapt” refers to an ERM model trained on labelled test-domain data.

Table 4. ColorMNIST test accuracies.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Test Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM (Vapnik, 1998)</td>
<td>27.9 ± 1.5</td>
</tr>
<tr>
<td>GroupDRO (Sagawa et al., 2019)</td>
<td>29.0 ± 1.1</td>
</tr>
<tr>
<td>IRM (Arjovsky et al., 2020)</td>
<td>69.7 ± 0.9</td>
</tr>
<tr>
<td>V-REx (Krueger et al., 2021)</td>
<td>71.6 ± 0.5</td>
</tr>
<tr>
<td>EQRM (Eastwood et al., 2022)</td>
<td>71.4 ± 0.4</td>
</tr>
<tr>
<td>SFB (Ours) w/o adapt.</td>
<td>70.6 ± 1.8</td>
</tr>
<tr>
<td>SFB (Ours) w. adapt.</td>
<td>88.1 ± 1.8</td>
</tr>
<tr>
<td>Oracle w/o adapt.</td>
<td>72.1 ± 0.7</td>
</tr>
<tr>
<td>Oracle w. adapt.</td>
<td>89.9 ± 0.1</td>
</tr>
</tbody>
</table>

F.2.2. Ablations

We now provide ablations on the ColorMNIST dataset to illustrate the effectiveness of the different components of SFB. In particular, we focus on bias correction and calibration, while also showing how multiple rounds of pseudo-labelling can improve performance in practice.

Bias correction. To adapt the unstable classifier in the test domain, SFB employs the bias-corrected adaptation algorithm of Alg. 1 (or Alg. 2 for the multi-class case) which corrects for biases caused by possible disagreements between the stable-predictor pseudo-labels $\hat{Y}$ and the true label $Y$. In this (sub)section, we investigate the performance of SFB with and without bias correction (BC).

Calibration. As discussed in § 3, correctly combining the stable and unstable predictions post-adaptation requires them to be properly calibrated. In particular, it requires the stable predictor $f_S$ to be calibrated with respect to the true labels $Y$ and the unstable predictor $f_U$ to be calibrated with respect to the pseudo-labels $\hat{Y}$. In this (sub)section, we investigate the performance of SFB with and without post-hoc calibration (in particular, simple temperature scaling (Guo et al., 2017)). More specifically, we investigate the effect of calibrating the stable predictor (CS) and calibrating the unstable predictor (CU).

Multiple rounds of pseudo-labelling. While SFB learns the optimal unstable classifier $h_U^*$ in the test domain given enough unlabelled data, § 3 showed how more accurate pseudo-labels $\hat{Y}$ improve the sample efficiency of SFB. In particular, in a restricted-sample setting, more accurate pseudo-labels result in an unstable classifier $h_U^*$ which better harnesses $X_U$ in the test domain. With this in mind, note that, after adapting, we expect the joint predictions of SFB to be more accurate than its stable-only predictions. This raises the question: can we use these improved predictions to form more accurate pseudo-labels, and, in turn, an unstable classifier $h_U^*$ that leads to even better performance? Furthermore, can we repeat this process, using multiple rounds of pseudo-labelling to refine our pseudo-labels and ultimately $h_U^*$? While this multi-round approach loses the asymptotic guarantees of Thm. 3.5, we found it to work quite well in practice. In this (sub)section, we thus investigate the performance of SFB with and without multiple rounds of pseudo-labelling (PL rounds).

Results. Table 5 reports the ablations of SFB on ColorMNIST. Here we see that: (i) bias correction significantly boosts performance (+BC); (ii) calibrating the stable predictor also boosts performance without (+CS) and with (+BC+CS) bias correction, with the latter leading to the best performance; (iii) calibrating the unstable predictor (with respect to the
Table 5. SFB ablations on ColorMNIST. Means and standard errors are over 3 random seeds. BC: bias correction. CS: post-hoc calibration of the stable classifier. CU: post-hoc calibration of the unstable classifier. PL Rounds: Number of pseudo-labelling rounds used. GT adapt: adapting using true labels in the test domain.

<table>
<thead>
<tr>
<th>Model</th>
<th>Bias Correction</th>
<th>Calibration Stable</th>
<th>PL Rounds</th>
<th>Test Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFB w/o adapt</td>
<td></td>
<td></td>
<td>1</td>
<td>70.6 ± 1.8</td>
</tr>
<tr>
<td>SFB with adapt +BC</td>
<td></td>
<td></td>
<td>1</td>
<td>78.0 ± 2.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>83.4 ± 2.8</td>
</tr>
<tr>
<td>SFB with adapt +CS</td>
<td>✔</td>
<td></td>
<td>1</td>
<td>80.6 ± 3.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>76.6 ± 2.4</td>
</tr>
<tr>
<td>SFB with adapt +CU</td>
<td>✔</td>
<td>✔</td>
<td>1</td>
<td>84.4 ± 2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>84.9 ± 2.6</td>
</tr>
<tr>
<td>SFB with adapt +BC+CS+CU</td>
<td>✔</td>
<td>✔</td>
<td>2</td>
<td>87.4 ± 1.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>88.1 ± 1.8</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>✔</td>
<td>4</td>
<td>88.6 ± 1.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>88.7 ± 1.3</td>
</tr>
<tr>
<td>SFB with GT adapt +BC+CS</td>
<td>✔</td>
<td>✔</td>
<td>1</td>
<td>89.0 ± 0.3</td>
</tr>
</tbody>
</table>

 pseudo-labels) slightly hurts performance without (+CU) and with (+BC+CS+CU) bias correction and stable-predictor calibration; (iv) multiple rounds of pseudo-labelling boosts performance, while also reducing the performance variation across random seeds; (v) using bias correction, stable-predictor calibration and 5 rounds of pseudo-labelling results in near-optimal adaptation performance, as indicated by the similar performance of SFB when using true labels $Y$ to adapt $h_{U}^{*}$ (denoted “SFB with GT adapt” in Table 5).

F.3. Camelyon17

Table 2 shows mixed results for Camelyon17 (Bandi et al., 2018). On the one hand, adapting gives SFB a small performance boost and reduces the variance across random seeds. On the other hand, the adapted performance is on par with both IRM and ERM. In line with (Gulrajani and Lopez-Paz, 2020), we found that a properly-tuned ERM model can be difficult to beat on real-world datasets, particularly when they don’t contain severe distribution shift. While we conducted this proper tuning for ERM, IRM and SFB (see Appendix G.3), doing so for ACTIR was non-trivial. We thus report the result from their paper (Jiang and Veitch, 2022, Tab. 1), which is likely lower due to sub-optimal hyperparameters (they report $\approx 70\%$ for ERM and IRM).

Table 6. Camelyon17 test-domain accuracies. Mean and standard errors are over 5 random seeds.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>90.2 ± 1.1</td>
</tr>
<tr>
<td>IRM</td>
<td>90.2 ± 1.1</td>
</tr>
<tr>
<td>ACTIR</td>
<td>77.7 ± 1.7†</td>
</tr>
<tr>
<td>SFB w/o adapt</td>
<td>89.8 ± 1.2</td>
</tr>
<tr>
<td>SFB w. adapt</td>
<td>90.3 ± 0.7</td>
</tr>
</tbody>
</table>

G. Implementation Details

Below we provide further implementation details for each of the experiments/datasets considered in this work. Code for reproducing all experimental results will be made available upon acceptance.
G.1. ColorMNIST

Training details. We follow the setup of Eastwood et al. (2022, §6.1) and build on their open-source code. In particular, we use the original MNIST training set to create training and validation sets for each domain, and the original MNIST test set for the test sets of each domain. For all methods, we use a 2-hidden-layer MLP with 390 hidden units, the Adam optimizer, a learning rate of 0.0001 with cosine scheduling, and dropout with $p = 0.2$. In addition, we use full batches (size 25000), 400 steps for ERM pertaining (which directly corresponds to the delicate penalty “annealing” or warm-up periods used by penalty-based methods on ColorMNIST [Arjovsky et al., 2020; Krueger et al., 2021; Eastwood et al., 2022]), and 600 total steps. We sweep over stability-penalty weights in $\{50, 100, 500, 1000, 5000\}$ for IRM, VREx and SFB and $\alpha$’s in $1 - \{e^{-100}, e^{-250}, e^{-500}, e^{-750}, e^{-1000}\}$ for EQRM. As the stable (shape) and unstable (color) features are conditionally independent given the label, we fix SFB’s conditional-independence penalty weight $\lambda_C = 0$. As is the standard for ColorMNIST, we use a test-domain validation set to select the best settings (after the total number of steps), and then report the mean and standard error over 10 random seeds on a test-domain test set. As in previous works, the hyperparameter ranges of all methods are selected by peeking at test-domain performance. While far from ideal, this is quite difficult to avoid with ColorMNIST and highlights a core problem with hyperparameter selection in DG—as discussed by many previous works (Arjovsky et al., 2020; Krueger et al., 2021; Gulrajani and Lopez-Paz, 2020; Zhang et al., 2022; Eastwood et al., 2022).

Adaptation details. For SFB’s unsupervised adaptation in the test domain, we use a batch size of 2048 and employ the bias correction of Alg. 1. In addition, we calibrate the stable predictor using post-hoc temperature scaling, choosing the temperature to minimize the expected calibration error (ECE, (Guo et al., 2017)) across the two training domains. Again using the two training domains for hyperparameter selection, we sweep over adaptation learning rates in $\{0.1, 0.01\}$, choose the best adaptation step in $[5, 20]$ (via early stopping), and sweep over the number of pseudo-labelling rounds in $[1, 3]$. Finally, we report the mean and standard error over 3 random seeds for adaptation.

G.2. PACS

We follow the experimental setup of Jiang and Veitch (2022, Section 6.4) and build on their open-source implementation. This means using an ImageNet-pretrained ResNet-18, the Adam optimizer with a learning rate of $10^{-4}$, and, following (Gulrajani and Lopez-Paz, 2020), choosing hyperparameters using leave-one-domain-out cross-validation. This is akin to K-fold cross-validation except with domains, meaning that we train 3 models—each time leaving out 1 of the 3 training domains for validation—and then select hyperparameters based on the best average performance across the held-out validation domains. Finally, we use the selected hyperparameters to retrain the model using all 3 training domains.

For SFB, we sweep over $\lambda_S$ in $\{0.01, 0.1, 1, 5, 10, 20\}$, $\lambda_C$ in $\{0.01, 0.1, 1\}$, and learning rates in $\{10^{-4}, 50^{-4}\}$. For SFB’s unsupervised adaptation, we employ the multi-class bias correction of Alg. 2 and calibrate the stable predictor using post-hoc temperature scaling, choosing the temperature to minimize the expected calibration error (ECE, (Guo et al., 2017)) across the three training domains. In addition, we use the Adam optimizer with an adaptation learning rate of 0.01, choosing the number of adaptation steps in $[1, 20]$ (via early stopping) using the training domains. Finally, we report the mean and standard error over 3 random seeds.

G.3. Camelyon17

We follow the experimental setup of Jiang and Veitch (2022, Section 6.3) and build on their open-source implementation. This means using an ImageNet-pretrained ResNet-18, the Adam optimizer, and, following (Koh et al., 2021), choosing hyperparameters using the validation domain (hospital 4). In contrast to (Jiang and Veitch, 2022), we use a learning rate of $10^{-5}$ for all methods, rather than $10^{-4}$, and employ early stopping using the validation domain. We found this to significantly improve all methods. E.g., the baselines of ERM and IRM improve by approximately 20 percentage points, jumping from $\approx 70\%$ to $\approx 90\%$.

For SFB, we sweep over $\lambda_S$ in $\{0.01, 0.1, 1, 5, 10, 20\}$ and $\lambda_C$ in $\{0.01, 0.1, 1\}$. For SFB’s unsupervised adaptation, we employ the bias correction of Alg. 1 and calibrate the stable predictor using post-hoc temperature scaling, choosing the

---

2https://github.com/cianeastwood/qrm/tree/main/CMNIST
3https://github.com/ybijaang/ACTIR.
4See Footnote 3.
temperature to minimize the expected calibration error (ECE, (Guo et al., 2017)) on the validation domain. In addition, we use the Adam optimizer with an adaptation learning rate of 0.01, choosing the number of adaptation steps in $[1, 20]$ (via early stopping) using the validation domain. Finally, we report the mean and standard error over 100 random seeds.

G.4. Synthetic

Following Jiang and Veitch (2022), we use a simple three-layer network with 8 units in each hidden layer and the Adam optimizer, choosing hyperparameters using the validation domain.

For SFB, we sweep over $\lambda_S$ in $\{0.01, 0.1, 1, 5, 10, 20\}$ and $\lambda_C$ in $\{0.01, 0.1, 1\}$. For SFB’s unsupervised adaptation, we employ the bias correction of Alg. 1 and calibrate the stable predictor using post-hoc temperature scaling, choosing the temperature to minimize the expected calibration error (ECE, (Guo et al., 2017)) on the validation domain. In addition, we use the Adam optimizer with an adaptation learning rate of 0.01, choosing the number of adaptation steps in $[1, 20]$ (via early stopping) using the validation domain. Finally, we report the mean and standard error over 100 random seeds.

H. Further Related Work

**Domain generalization.** A fundamental starting point for work in domain generalization and robustness is the observation that certain “stable” features, often direct causes of the label, may have an invariant relationship with the label across domains (Peters et al., 2016; Arjovsky et al., 2020; Veitch et al., 2021; Schölkopf, 2022; Makar et al., 2022; Zheng and Makar, 2022). However, such stable or causal predictors often discard highly-informative but unstable information about the label. Rothenhäusler et al. (2021) show that we may need to trade-off stability and predictiveness, with the causal predictor often too conservative. Eastwood et al. (2022) seek such a trade-off via an interpretable probability-of-generalization parameter. The current work is motivated by the idea that one might avoid such a trade-off by changing how spurious features are used at test time, rather than discarding them at training time.

**Test-domain adaptation with labels.** Fine-tuning part of a model using a small number of labelled test-domain examples is a common way to deal with distribution shift (Fei-Fei et al., 2006; Finn et al., 2017; Eastwood et al., 2021). More recently, it has been shown that simply retraining the last layer of an ERM-trained model outperforms more robust feature-learning methods on spurious correlation benchmarks (Rosenfeld et al., 2022; Kirichenko et al., 2022). In particular, Jiang and Veitch (2022) do so when using a conditional-independence assumption similar to ours. All of these works require labels in the test domain, while we seek to adapt without labels.

**Learning with noisy labels.** An intermediate goal in our work, namely learning a model to predict $Y$ from $X_U$, is an instance of learning with noisy labels, a widely studied problem (Scott et al., 2013; Natarajan et al., 2013; Blanchard et al., 2016; Song et al., 2022; Li et al., 2017b; Tanaka et al., 2018). Specifically, under the complementarity assumption ($X_S \perp \!
\!\!\perp X_U \mid Y$), the accuracy of the pseudo-labels on each class is independent of $X_U$, placing us in the so-called class-conditional random noise model (Scott et al., 2013; Natarajan et al., 2013; Blanchard et al., 2016). As we discuss in Section 3, our theoretical insights about the special structure of pseudo-labels complement existing results on learning under this model. Our bias-correction (Eq. (3.1)) for $P_{Y \mid X_U}$ is also closely related to the “method of unbiased estimators” (Natarajan et al., 2013). However, rather than correcting the loss used in ERM, our post-hoc bias correction applies to any calibrated classifier. Moreover, our ultimate goal, learning a predictor of $Y$ jointly using $X_S$ and $X_U$, is not captured by learning with noisy labels.

**Co-training.** Our use of stable-feature pseudo-labels to train a classifier based on a disjoint subset of (unstable) features is reminiscent of co-training (Blum and Mitchell, 1998). Both methods benefit from conditional independence of the two feature subsets given the label to ensure that they provide complementary information. The key difference is that while co-training requires (a small number of) labeled samples from the same distribution as the test data, our method instead uses labeled data from a different distribution (training domains), along with the assumption of a stable feature.

**Using spurious or unstable features without labels.** Bui et al. (2021) exploit-domain specific or unstable features with a meta-learning approach. However, they use the unstable features in the same way in the test domain, which, by their very definition, can lead to degraded performance. In contrast, we seek a robust approach to safely harness the unstable features

\footnote{See Krogel and Scheffer (2004) and Theorem 1 of Blum and Mitchell (1998) for discussion of this assumption.}
in the test domain, as summarised in Table 1. Sun et al. (2022) share the goal of exploiting spurious or unstable features to go “beyond invariance”. However, their approach requires labels for the spurious features at training time and only applies to label shifts. In contrast, we do not require labels for the spurious features and are not restricted to label shifts.

Self-learning via pseudo-labelling. In the source-free and test-time domain adaptation literature, adapting to the test domain using a model’s own pseudo-labels is a common approach (Lee et al., 2013; Liang et al., 2020; Wang et al., 2021; Iwasawa and Matsuo, 2021)—see Rusak et al. (2022) for a recent review. In contrast to these approaches, we use one model to provide the pseudo-labels (the stable model) and the other to use/adapt to the pseudo-labels (the unstable model). In addition, while the majority of this pseudo-labelling work is purely empirical, we provide theoretical justification and guarantees for our SFB approach.

I. Limitations

In our view, the most significant limitation of this work is the assumption of complementarity (i.e., that the spurious features are conditionally independent of the stable features, given the label). Complementarity is implicit in the causal generative models assumed by existing related work (Rojas-Carulla et al., 2018; von Kügelgen et al., 2021; Jiang and Veitch, 2022), and, as Example A.1 in Appendix A.1 demonstrates, is cannot simply be dropped from our theoretical motivation. In the related context of co-training, this condition was initially assumed and then weakened in subsequent work (Blum and Mitchell, 1998; Balcan et al., 2004; Abney, 2002; Wang and Zhou, 2010); similarly, we hope future work will identify weaker conditions that are sufficient for SFB to succeed. On the other hand, our experimental results on the synthetic dataset of Appendix F.1, as well as the real datasets of PACS and Camelyon17, suggest that SFB may be robust to violations of complementarity—perhaps mirroring the surprisingly good practical performance of methods such as naive Bayes classification which are justified under similar assumptions (Rish et al., 2001).

J. Discussion

This work demonstrated, both theoretically and practically, how to adapt spurious but informative features to new test domains using only a stable, complementary training signal. Our proposed Stable Feature Boosting algorithm can provide significant performance gains compared to only using stable features or using unadapted spurious features, without requiring any true labels in the test domain. In theory, the most significant limitation of SFB is its assumption of complementarity (i.e., conditional independence of spurious features and stable features, given the label). Importantly, our experimental results suggest that SFB may robust to violations of complementarity in practice; on real-world datasets such as PACS or Camelyon17, where there is no reason to believe complementarity holds, SFB performs at least as well or better than unadapted methods such as ERM and IRM.