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]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany; [2]University of Edinburgh, United Kingdom; [3]University of Cambridge, United Kingdom;

Summary

- We show when and how it is possible to **safely harness spurious or unstable features without test-domain labels**.
- We prove that predictions based on invariant or stable features provide sufficient guidance for doing so, provided that the **stable and unstable features are conditionally independent given the label**.
- We propose the **Stable Feature Boosting** (SFB) algorithm for optimally harnessing complementary spurious features without labels.

Motivation

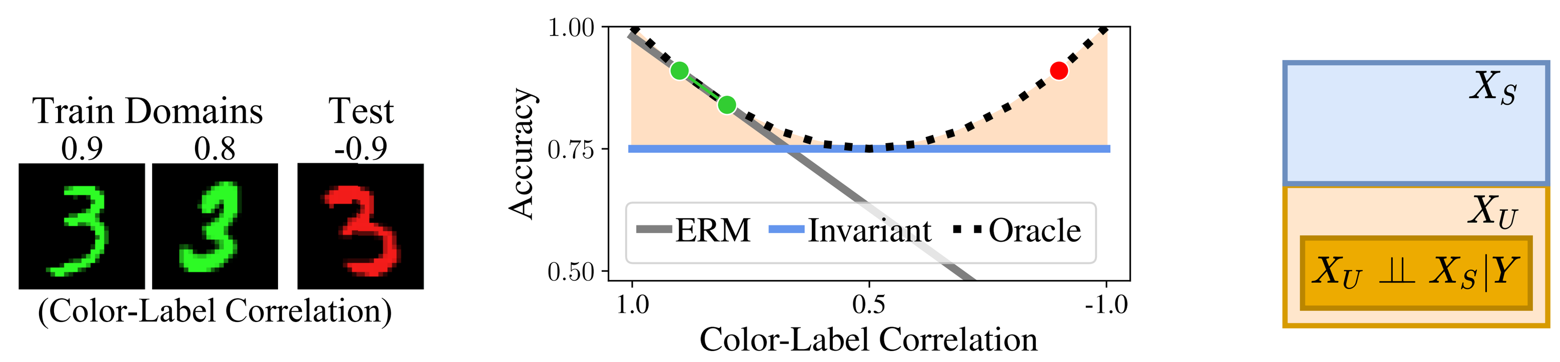

Figure 1. **Invariant (stable) and spurious (unstable) features.**

(a) Illustrative images from the `CMNIST` dataset.
(b) `CMNIST` accuracies (y-axis) over test domains of decreasing color-label correlation (x-axis). The 'Oracle' uses both invariant (shape) *and* spurious (color) features optimally in the test domain, boosting performance over an invariant model (orange region). **Our main contribution is to show how this can be done *without labels*.**
(c) Generally, invariant models use only the *stable* component X_S of X, discarding the spurious or *unstable* component X_U. We prove that predictions based on X_S can be used to safely harness a sub-component of X_U (dark-orange region), boosting test-domain performance.

Stable Features

Consider feature-label pairs (X, Y) drawn conditioned on domain E.

Definition: X is *stable* with respect to Y if $P_{Y|X}$ does not depend on the domain E; i.e., Y and E are conditionally independent given X.

Related Work

Method	Components of X Used: Stable	Complementary	All	Robust	No test-domain labels
ERM	✓	✓	✓	✗	✓
IRM [1]	✓	✗	✗	✓	✓
QRM [2]	✓	✓*	✓*	✓*	✓
DARE [4]	✓	✓	✓	✓	✗
ACTIR [3]	✓	✓	✗	✓	✗
SFB (**Ours**)	✓	✓	✗	✓	✓

Theorem: Test-Domain Adaptation

Consider three random variables X_S, X_U, and Y. Suppose

1. Y is binary ($\{0,1\}$-valued) (this can be relaxed)
2. X_S is **informative** of Y: $X_S \not\perp\!\!\!\perp Y$
3. X_S and X_U are **complementary** features for Y: $X_S \perp\!\!\!\perp X_U|Y$

Specifically, suppose $\hat{Y}|X_S \sim \text{Bernoulli}(\Pr[Y=1|X_S])$ is a pseudo-label,

$$\varepsilon_0 := \Pr[\hat{Y}=0|Y=0] \quad \text{and} \quad \varepsilon_1 := \Pr[\hat{Y}=1|Y=1]$$

are the class-wise accuracies of these pseudo-labels. Then,

1. $\varepsilon_0 + \varepsilon_1 > 1$,
2. $\Pr[Y=1|X_U] = \dfrac{\Pr[\hat{Y}=1|X_U] + \varepsilon_0 - 1}{\varepsilon_0 + \varepsilon_1 - 1}$, and
3. For $C(a,b,c) = \sigma\left(\text{logit}(a) + \text{logit}(b) - \text{logit}(c)\right)$, we have
$$\Pr[Y=1|X_S, X_U] = C\left(\Pr[Y=1|X_S], \Pr[Y=1|X_U], \Pr[Y=1]\right).$$

All of these quantities can be computed from joint distributions of (X_S, Y) **and** $(X_U, \hat{Y})$**.**
Idea: Learn (X_S, Y) from training data, learn $(X_U, \hat{Y})$ from unlabeled test data, and then adapt using above formulas.

Algorithm: Stable Feature Boosting (SFB)

Boosted joint predictor in domain e:

$$\begin{aligned} f^e(X) = C(f_S(X), f_U^e(X)) &= C(h_S(\Phi_S(X)), h_U^e(\Phi_U(X))) \\ &= C(h_S(X_S), h_U^e(X_U)). \end{aligned}$$

Learning goals:

1. f_S is a stable and calibrated predictor with good performance.
2. In a given domain e, f_U^e boosts the performance of f_S using complementary features $\Phi_U(X^e) \perp\!\!\!\perp \Phi_S(X^e)|Y^e$.

Objective function:

$$\min_{\Phi_S, \Phi_U, h_S, h_U^e} \sum_{e \in \mathcal{E}_{\text{tr}}} R^e(h_S \circ \Phi_S) + R^e(C(h_S \circ \Phi_S, h_U^e \circ \Phi_U))$$
$$+ \lambda_S \cdot P_{\text{Stability}}(\Phi_S, h_S, R^e) + \lambda_c \cdot P_{\text{CondIndep}}(\Phi_S(X^e), \Phi_U(X^e), Y^e)$$

Post-hoc calibration: Simple temperature scaling.

Test-domain adaptation: Apply previous Theorem to stable classifier h_S and unlabelled test-domain dataset $\{\Phi_S(x_i), \Phi_U(x_i)\}_{i=1}^{n_e}$.

Experiments

On `CMNIST` SFB can attain near-optimum performance across domains, with bias correction (BC) and calibration (CA) being essential.

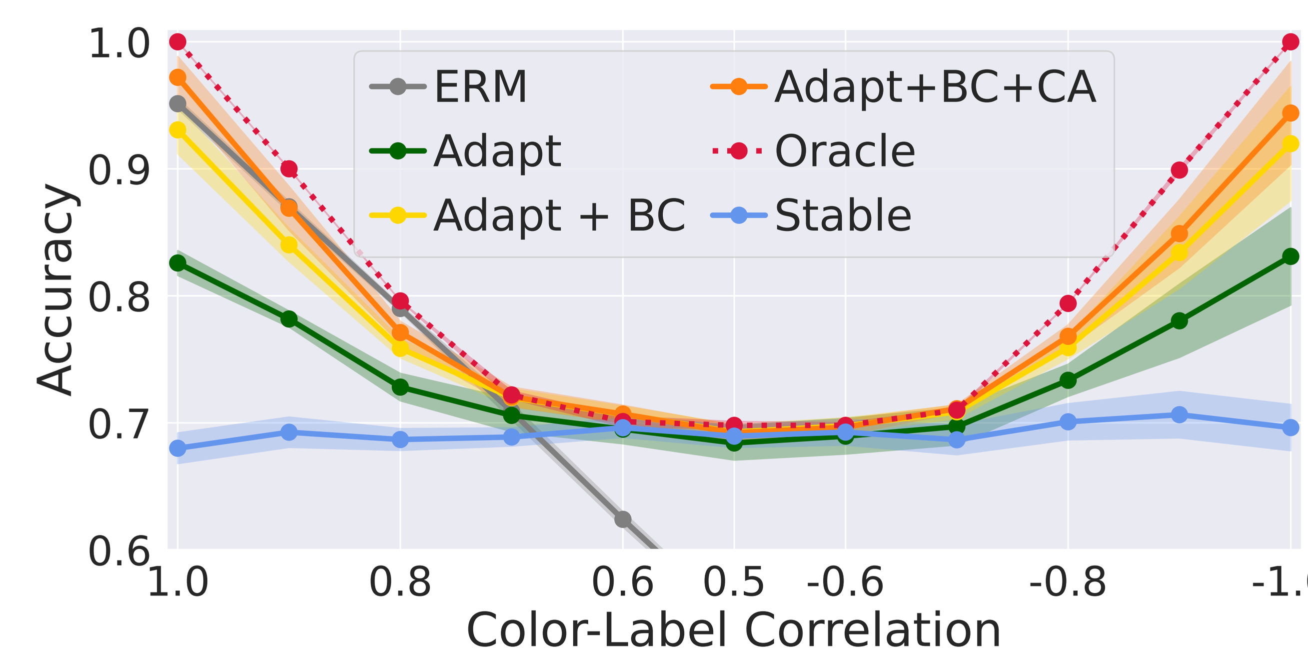

Figure 2. `CMNIST` accuracies over test domains of decreasing color-label correlation. Oracle: ERM with labelled test-domain data. All other curves (but ERM) refer to our algorithm. 'Stable': unadapted, 'BC': bias-corrected, and 'CA': calibrated.

We experiment on synthetic data with an anti-causal (AC) or cause-effect with a direct X_S-X_U dependence (CE-DD) structure. $X_S \perp\!\!\!\perp X_U|Y$ holds for the former, but not for the latter. We also evaluate real image datasets (PACS).

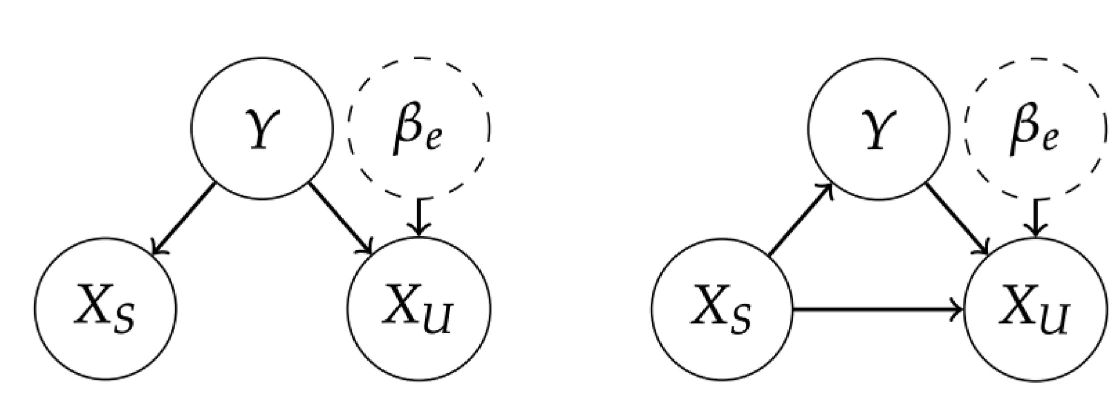

Figure 3. Synthetic-data DAGs. Dashed lines for unobserved variables.

Table 1. Test-domain accuracies over 100 (`Synthetic`) and 5 (`PACS`) seeds.

Algorithm	`Synthetic` AC	`Synthetic` CE-DD	`PACS` P	`PACS` A	`PACS` C	`PACS` S
ERM	9.9 ± 0.1	11.6 ± 0.7	93.0 ± 0.7	79.3 ± 0.5	74.3 ± 0.7	65.4 ± 1.5
IRM	74.9 ± 0.1	69.6 ± 1.3	93.3 ± 0.3	78.7 ± 0.7	75.4 ± 1.5	65.6 ± 2.5
ACTIR	74.8 ± 0.4	43.5 ± 2.6	94.8 ± 0.1	$\mathbf{82.5 \pm 0.4}$	$\mathbf{76.6 \pm 0.6}$	62.1 ± 1.3
SFB w/o adapt	74.7 ± 1.2	74.9 ± 3.6	93.7 ± 0.6	78.1 ± 1.1	73.7 ± 0.6	69.7 ± 2.3
SFB w. adapt	$\mathbf{89.2 \pm 2.9}$	$\mathbf{88.6 \pm 1.4}$	$\mathbf{95.8 \pm 0.6}$	80.4 ± 1.3	$\mathbf{76.6 \pm 0.6}$	$\mathbf{71.8 \pm 2.0}$

Discussion

- Exploiting *newly-available* test-domain features without labels
- Weakening the complementarity condition

References

[1] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization, 2020. arXiv:1907.02893.
[2] C. Eastwood, A. Robey, S. Singh, J. von Kügelgen, H. Hassani, G. J. Pappas, and B. Schölkopf. Probable domain generalization via quantile risk minimization. In *NeurIPS*, 2022.
[3] Y. Jiang and V. Veitch. Invariant and transportable representations for anti-causal domain shifts. In *NeurIPS*, 2022.
[4] E. Rosenfeld, P. Ravikumar, and A. Risteski. Domain-adjusted regression or: ERM may already learn features sufficient for out-of-distribution generalization. *arXiv preprint arXiv:2202.06856*, 2022.