

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

12-2023

Make the U in UDA matter: Invariant consistency learning for unsupervised domain adaptation

Zhongqi YUE

Qianru SUN

Singapore Management University, qianrusun@smu.edu.sg

Hanwang ZHANG

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Databases and Information Systems Commons](#)

Citation

YUE, Zhongqi; SUN, Qianru; and ZHANG, Hanwang. Make the U in UDA matter: Invariant consistency learning for unsupervised domain adaptation. (2023). *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023), New Orleans, December 10-16*. 1-20.

Available at: https://ink.library.smu.edu.sg/sis_research/8474

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylids@smu.edu.sg.

Make the U in UDA Matter: Invariant Consistency Learning for Unsupervised Domain Adaptation

Zhongqi Yue¹, Hanwang Zhang¹, Qianru Sun²

¹Nanyang Technological University, ²Singapore Management University
yuez0003@ntu.edu.sg, hanwangzhang@ntu.edu.sg, qianrusun@smu.edu.sg

Abstract

Domain Adaptation (DA) is always challenged by the spurious correlation between domain-invariant features (*e.g.*, class identity) and domain-specific features (*e.g.*, environment) that does not generalize to the target domain. Unfortunately, even enriched with additional unsupervised target domains, existing Unsupervised DA (UDA) methods still suffer from it. This is because the source domain supervision only considers the target domain samples as auxiliary data (*e.g.*, by pseudo-labeling), yet the inherent distribution in the target domain—where the valuable de-correlation clues hide—is disregarded. We propose to make the U in UDA matter by giving equal status to the two domains. Specifically, we learn an invariant classifier whose prediction is simultaneously consistent with the labels in the source domain and clusters in the target domain, hence the spurious correlation inconsistent in the target domain is removed. We dub our approach “Invariant Consistency learning” (ICON). Extensive experiments show that ICON achieves the state-of-the-art performance on the classic UDA benchmarks: OFFICE-HOME and VISDA-2017, and outperforms all the conventional methods on the challenging WILDS 2.0 benchmark. Codes are in <https://github.com/yue-zhongqi/ICON>.

1 Introduction

Domain Adaptation (DA) is all about training a model in a labelled source domain S (*e.g.*, an autopilot trained in daytime), and the model is expected to generalize in a target domain T (*e.g.*, the autopilot deployed in nighttime), where T and S has a significant domain shift (*e.g.*, day vs. night) [43]. To illustrate the shift and generalization, we introduce the classic notion of causal representation learning [22, 51]: any sample is generated by $\mathbf{x} = \Phi(\mathbf{c}, \mathbf{e})$, where Φ is the generator, \mathbf{c} is the causal feature determining the domain-invariant class identity (*e.g.*, road lane), and \mathbf{e} is the environmental feature (*e.g.*, lighting conditions). As the environment is domain-specific, the domain shift $\mathbf{e}_S \neq \mathbf{e}_T$ results in $\mathbf{x}_S \neq \mathbf{x}_T$, challenging the pursuit of a domain-invariant class model. However, if a model disentangles the causal feature by transforming each $\mathbf{x} = \mathbf{c}$, it can generalize under arbitrary shift in \mathbf{e} . In Figure 1, even though $\mathbf{e}_T = \text{“colorful”}$ is significantly

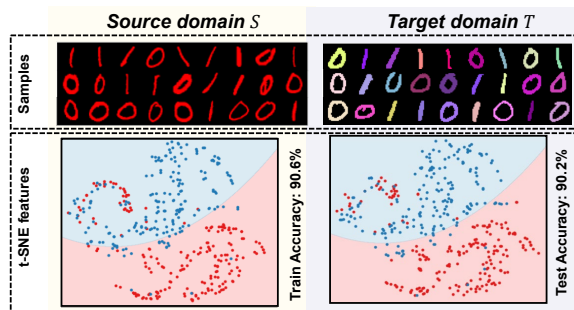


Figure 1: A digit classification model trained on red digits in S generalizes to colorful digits in T by disentangling digit shape (*i.e.*, the causal feature \mathbf{c}), as color in S (*i.e.*, \mathbf{e}_S) is not discriminative. \bullet : “0”, \bullet : “1”. The classification boundary is rendered blue and red.

can generalize under arbitrary shift in \mathbf{e} . In Figure 1, even though $\mathbf{e}_T = \text{“colorful”}$ is significantly

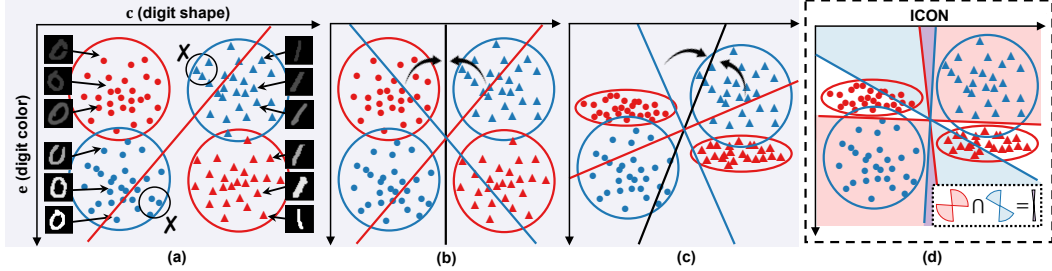


Figure 2: **Red:** S , **Blue:** T . \bullet : “0”, \blacktriangle : “1”. (a) A failure case of existing lineups. The red line denotes the classifier trained in S . (b) Classifier can be corrected (black line) by respecting the BCE loss between S (red line) and T (blue line). (c) A failure case of minimizing the combined BCE losses in S and T . Due to the low variance of e_S in each class, the optimal classifier *w.r.t.* the BCE loss in S (red line) looks “flat”, *i.e.*, it has large weight on e [41]. (d) Our ICON prevents such a failure case.

different from $e_S = \text{“red”}$, thanks to the non-discriminative “red” color in S , the model still easily disentangles c from $\Phi(c, e_S)$ and thus generalizes to digits of any color.

Unfortunately, in general, disentangling c by using only S is impossible as e_S can be also discriminative, *i.e.*, spuriously correlated with c . In Figure 2a, the color e_S is correlated with the shape c , *i.e.*, $c_1 = \text{“0”}$ and $c_2 = \text{“1”}$ in S tend to have $e_1 = \text{dark}$ and $e_2 = \text{light}$ digit colors, respectively. As both c and e_S are discriminative, a classifier trained in S will inevitably capture both (see the red line in Figure 2a). This leads to poor generalization when the correlation between e_T and c is different, *i.e.*, the colors of $c_1 = \text{“0”}$ and $c_2 = \text{“1”}$ in T tend to be $e_2 = \text{light}$ and $e_1 = \text{dark}$ instead.

To avoid spurious correlations between e_S and c , a practical setting called Unsupervised DA (UDA) introduces additional information about T through a set of unlabeled samples [17, 37, 14]. There are two main lineups: 1) Domain-alignment methods [60, 36, 12] learn the common feature space of S and T that minimizes the classification error in S . In Figure 2a, their goal is to unify “0”s and “1”s in S and T . However, they suffer from the well-known misalignment issue [33, 73], *e.g.*, collapsing “0”s and “1”s in T to “1”s and “0”s in S respectively also satisfies the training objective, but it generalizes poorly to T . 2) Self-training methods [74, 40, 54] use a classifier trained in S to pseudo-label samples in T , and jointly train the model with the labels in S and confident pseudo-labels in T . Yet, the pseudo-labels can be unreliable even with expensive threshold tweaking [34, 31], *e.g.*, the \times areas in Figure 2a have confident but wrong pseudo-labels. In fact, the recent WILDS 2.0 benchmark [49] on real-world UDA tasks shows that both lineups even under-perform an embarrassingly naive baseline: directly train in S and test in T .

Notably, although both methods respect the idiosyncrasies in S (by minimizing the classification error), they fail to account for the inherent distribution in T , *e.g.*, in Figure 2a, even though T is unlabelled, we can still identify the two sample clusters in T (enclosed by the blue circles), which provide additional supervision for the classifier trained in S . In particular, the classifier (red line) in Figure 2a breaks up the two clusters in T , showing that the correlation between color e_S and c in S is inconsistent with the clusters affected by e_T in T , which implies that color e is the environmental feature. Hence, to make the U in UDA matter, we aim to learn a classifier that is consistent with classification in S and clustering in T :

Consistency. We use the Binary Cross-Entropy (BCE) loss to encourage the prediction similarity of each sample pair from the same class of S or cluster of in T , and penalize the pairs from different ones. In Figure 2b, the red and blue lines denote the classifiers that minimize the BCE loss in S and T , respectively. By minimizing the combined loss, we learn the classifier consistent with the sample distribution in both domains (the upright black line), which predicts solely based on the causal feature c (*i.e.*, disentangling c from $\Phi(c, e)$).

Invariance. However, enforcing consistency is a tug-of-war between S and T , and the BCE loss in one domain can dominate the other to cause failures. Without loss of generality, we consider the case where S dominates in Figure 2c. Due to the strong correlation between e_S and c , the classifier that minimizes the combined losses of BCE (black line) deviates from the desired upright position as in Figure 2b. To tackle this, we aim to learn an invariant classifier [1] that is simultaneously optimal *w.r.t.* the BCE loss in S and T . In Figure 2d, for each of the two losses, we color the candidate regions where the classifier is optimal (*i.e.*, gradients ≈ 0) with red and blue, respectively. In particular, by

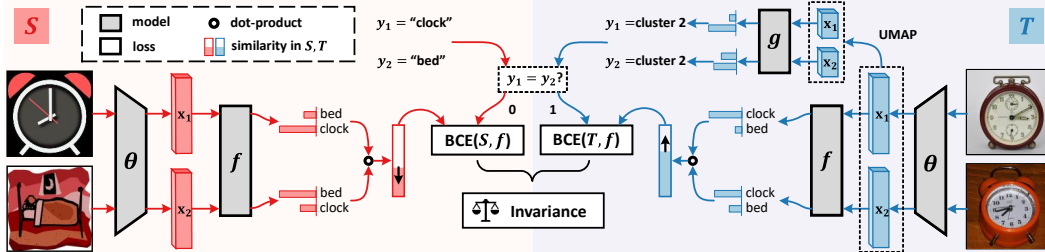


Figure 3: A training example where the pair of samples in S is from different classes and that in T is from the same cluster. The black arrows on the similarity bars denotes that minimizing the BCE losses in S and T will decrease the similarity in S (as $y_1 \neq y_2$) and increase that in T (as $y_1 = y_2$).

learning an invariant classifier lying in the intersection (purple region), we successfully recover the desired one as in Figure 2b, even when one domain dominates.

Overall, we term our approach as **Invariant CONSistency learning (ICON)**. Our contributions:

- ICON is a novel UDA method that can be plugged into different self-training baselines. It is simple to implement (Section 2), and disentangles the causal feature \mathbf{c} from $\Phi(\mathbf{c}, \mathbf{e})$ that generalizes to T with a theoretical guarantee (Section 3).
- ICON significantly improves the current state-of-the-art on classic UDA benchmarks [31, 6] with ResNet-50 [21] as the feature backbone: 75.8% (+2.6%) accuracy averaging over the 12 tasks in OFFICE-HOME [62], and 87.4% (+3.7%) mean-class accuracy in VISDA-2017 [45] (Section 5).
- Notably, as of NeurIPS 2023, ICON dominates the leaderboard of the challenging WILDS 2.0 UDA benchmark [49] (<https://wilds.stanford.edu/leaderboard/>), which includes 8 large-scale classification, regression and detection tasks on image, text and graph data (Section 5).

2 Algorithm

UDA aims to learn a model with the labelled training data $\{\mathbf{x}_i, y_i\}_{i=1}^N$ in the source domain S and unlabelled $\{\mathbf{x}_i\}_{i=1}^M$ in the target domain T , where \mathbf{x}_i denotes the feature of the i -th sample extracted by a backbone parameterized by θ (e.g., ResNet-50 [21]), and y_i is its ground-truth label. We drop the subscript i for simplicity when the context is clear. As shown in Figure 3, our model includes the backbone θ , a classification head f , and a cluster head g , where f and g output the softmax-normalized probability for each class and cluster, respectively. Note that f and g have the same output dimension as the class numbers are the same in S and T . As discussed in Section 1, the generalization objective of ICON is to learn an invariant f that is simultaneously consistent with the classification in S , and the clustering in T , identified by g . ICON is illustrated in Figure 3 and explained in details below.

2.1 Consistency with S

We use Binary Cross-Entropy (BCE) loss to enforce the consistency of f with S , denoted as $\text{BCE}(S, f)$. Specifically, $\text{BCE}(S, f)$ is a pair-wise loss given by:

$$\text{BCE}(S, f) = \mathbb{E}_{(\mathbf{x}_i, \mathbf{x}_j) \sim S} [b \log(\hat{y}) + (1 - b) \log(1 - \hat{y})], \quad (1)$$

where $b = \mathbb{1}(y_i = y_j)$ is a binary label indicating if the pair $(\mathbf{x}_i, \mathbf{x}_j)$ is from the same class, and $\hat{y} = f(\mathbf{x}_i)^\top f(\mathbf{x}_j)$ is the predictive similarity. Intuitively, BCE is a form of contrastive loss that clusters sample pairs with the same label (by increasing the dot-product similarity of their predictions) and pushes away pairs with different labels (by decreasing the similarity). Note that in practice, BCE loss is computed within each mini-batch, hence there is no N^2 prohibitive calculation.

2.2 Consistency with T

Unfortunately, $\text{BCE}(T, f)$ cannot be evaluated directly, as the binary label b in Eq. (1) is intractable given unlabeled T . To tackle this, we cluster T to capture its inherent distribution and use cluster labels to compute $\text{BCE}(T, f)$.

Clustering T . We adopt the rank-statistics algorithm [20], as it can be easily implemented in a mini-batch sampling manner for online clustering. The algorithm trains the cluster head g to group features whose top values have the same indices (details in Appendix). Ablation on clustering algorithm is in Section 5.4. Note that our g is trained from scratch, rather than initialized from f trained in S [58, 31, 34]. Hence g captures the distribution of T without being affected by that of S .

Computing $\text{BCE}(T, f)$. After training g , we replace $b = \mathbb{1}(y_i = y_j)$ for $\text{BCE}(S, f)$ in Eq. (1) with $b = \mathbb{1}(\arg\max g(\mathbf{x}_i) = \arg\max g(\mathbf{x}_j))$ for $\text{BCE}(T, f)$, which compares the cluster labels of each sample pair. We emphasize that it is necessary to use pairwise BCE loss to enforce the consistency of f with T . This is because cross-entropy loss is not applicable when the cluster labels in T are not aligned with the class indices in S .

2.3 Invariant Consistency (ICON)

ICON simultaneously enforces the consistency of f with S (*i.e.*, the decision boundary of f separates the classes) and T (*i.e.*, it also separates the clusters). The objective is given by:

$$\begin{aligned} & \min_{\theta, f} \overbrace{\text{BCE}(S, f) + \text{BCE}(T, f)}^{\text{Consistency}} + \overbrace{\text{CE}(S, f) + \alpha \mathcal{L}_{st}}^{S\text{-Supervision}} \\ \text{s.t. } & \underbrace{f \in \arg\min_{\bar{f}} \text{BCE}(S, \bar{f}) \cap \arg\min_{\bar{f}} \text{BCE}(T, \bar{f})}_{\text{Invariance}}. \end{aligned} \quad (2)$$

Line 1. The BCE losses train f to be consistent with the pair-wise label similarity in S and T . CE loss in S trains f to assign each sample its label (*e.g.*, predicting a clock feature as “clock”). \mathcal{L}_{st} is the self-training loss with weight α , which leverages the learned invariant classifier to generate accurate pseudo-labels in T for additional supervision (details in Section 5.2).

Line 2. The constraint below requires the backbone θ to elicit an invariant classifier f that simultaneously minimizes the BCE loss in S and T . This constraint prevents the case where one BCE loss is minimized at the cost of increasing the other to pursue a lower sum of the two (Figure 2c).

To avoid the challenging bi-level optimization, we use the following practical implementation:

$$\min_{\theta, f} \text{BCE}(S, f) + \text{BCE}(T, f) + \alpha \mathcal{L}_{st} + \beta \text{Var}(\{\text{BCE}(S, f), \text{BCE}(T, f)\}), \quad (3)$$

where the variance term $\text{Var}(\cdot)$ is known as the REx loss [27] implementing the invariance constraint. The self-training weight α and invariance weight β are later studied in ablation (Section 5.4).

Training and Testing. Overall, at the start of training, the backbone θ is initialized from a pre-trained weight (*e.g.*, on ImageNet [48]) and f, g are randomly initialized. Then ICON is trained by Eq. (3) until convergence. Only the backbone θ and the classifier f are used in testing.

3 Theory

Preliminaries. We adopt the group-theoretic definition of disentanglement [22, 64] to justify ICON. We start by introducing some basic concepts in group theory. \mathcal{G} is a group of semantics that generate data by group action, *e.g.*, a “turn darker” element $g \in \mathcal{G}$ transforms \mathbf{x} from white “0” to $g \circ \mathbf{x}$ as grey “0” (bottom-left to top-left in Figure 4b). A sketch of theoretical analysis is given below and interested readers are encouraged to read the full proof in Appendix.

Definition (Generalization). *Feature space \mathcal{X} generalizes under the direct product decomposition $\mathcal{G} = \mathcal{G}/\mathcal{H} \times \mathcal{H}$, if \mathcal{X} has a non-trivial \mathcal{G}/\mathcal{H} -representation, *i.e.*, the action of each $h \in \mathcal{H}$ corresponds to a trivial linear map in \mathcal{X} (*i.e.*, identity map), and the action of each $g \in \mathcal{G}/\mathcal{H}$ corresponds to a non-trivial one.*

This definition formalizes the notion of transforming each sample $\Phi(\mathbf{c}, \mathbf{e})$ to the causal feature \mathbf{c} . Subgroup $\mathcal{H} \subset \mathcal{G}$ and quotient group \mathcal{G}/\mathcal{H} transform the environmental feature \mathbf{e} and the causal \mathbf{c} , respectively. If \mathcal{X} has a non-trivial \mathcal{G}/\mathcal{H} -representation, samples in a class are transformed to the same feature $\mathbf{c} \in \mathcal{X}$ regardless of \mathbf{e} (*i.e.*, representation of \mathcal{G}/\mathcal{H}), and different classes have different \mathbf{c} (*i.e.*, non-trivial). This definition is indeed in accordance with the common view of good features [64], *i.e.*, they enable zero-shot generalization as \mathbf{c} is class-unique, and they are robust to the

domain shift by discarding the environmental features e from $\Phi(c, e)$. To achieve generalization, we need the following assumptions in ICON.

Assumption (Identifiability of \mathcal{G}/\mathcal{H}).

- 1) $\forall \mathbf{x}_i, \mathbf{x}_j \in T, y_i = y_j$ iff $\mathbf{x}_i \in \{h \circ \mathbf{x}_j \mid h \in \mathcal{H}\}$;
- 2) There exists no linear map $l : \mathcal{X} \rightarrow \mathbb{R}$ such that $l(\mathbf{x}) > l(h \circ \mathbf{x}), \forall \mathbf{x} \in S, h \circ \mathbf{x} \in T$ and $l(g \circ \mathbf{x}) < l(gh \circ \mathbf{x}), \forall g \circ \mathbf{x} \in S, gh \circ \mathbf{x} \in T$, where $h \neq e \in \mathcal{H}, g \neq e \in \mathcal{G}/\mathcal{H}$.

Assumption 1 states that classes in T are separated by clusters. This corresponds to the classic clustering assumption, a necessary assumption for learning with unlabeled data [61]. Figure 4a shows a failure case of Assumption 1, where the intra-class distance (i.e., dotted double-arrow) is much larger than the inter-class one (i.e., solid double-arrow) in T , i.e., shape c is less discriminative than color e_T . Hence the clusters in T are based on e , causing ICON to capture the wrong invariance. To help fulfill the assumption, UDA methods (and SSL in general) commonly leverage feature pre-training and data augmentations. We also specify the number of clusters as the class numbers in T , which is a necessary condition of this assumption.

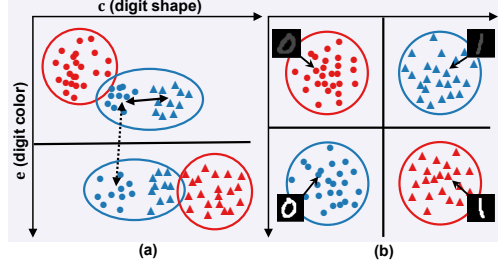


Figure 4: Red: S , Blue: T . \bullet : “0”, \blacktriangle : “1”. Black lines denote the invariant classifier satisfying the ICON objective. (a) Assumption 1 is violated. (b) Assumption 2 is violated.

Assumption 2 states that c is the only invariance between S and T . Figure 4b illustrates a failure case, where there exist two invariant classifiers: both the vertical one based on shape c and the horizontal one based on color e separate the classes in S and clusters in T . Yet as T is unlabelled, there is no way for the model to determine which one is based on c . In practice, this assumption can be met by collecting more diverse unlabelled samples in T (e.g., darker “0”s and lighter “1”s in Figure 4b), or introducing additional priors on what features should correspond to the causal one.

Theorem. When the above assumptions hold, ICON (optimizing Eq. (3)) learns a backbone θ mapping to a feature space \mathcal{X} that generalizes under $\mathcal{G}/\mathcal{H} \times \mathcal{H}$. In particular, the learned f is the optimal classifier in S and T .

By transforming each sample $\Phi(c, e)$ in S and T to $c \in \mathcal{X}$, the classifier easily associate each c to its corresponding class label using the labelled S to reach optimal.

4 Related Work

Domain-Alignment methods [15, 4, 59] map samples in S and T to a common feature space where they become indistinguishable. Existing works either minimize the discrepancy between the sample distributions of S and T [38, 2, 29], or learn features that deceive a domain discriminator [71, 65, 35]. However, they suffers from the misalignment issue under large domain shift from S to T [73, 33], such as shift in the support of the sample distribution [25] (e.g., Figure 2a).

Self-Training methods [19, 3, 55] are the mainstream in semi-supervised learning (SSL), and recently became a promising alternative in UDA, which focuses on generating accurate T pseudo-labels. This lineup explores different classifier design, such as k-NN [53], a teacher-student network [13] or an exponential moving average model [32]. Yet the pseudo-labels are error-prone due to the spurious correlations in S , hence the model performance is still inconsistent [34].

Alleviating Spurious Correlation has been studied in several existing works of UDA [58, 34, 31]. They train a T -specific classifier to generate pseudo-labels [58, 31]. However, their T -classifiers are still initialized based on a classifier trained in S , hence the spurious correlation in S are introduced to the T -classifiers right in the beginning. On the contrary, our ICON learns a cluster head in T without supervision in S , which is solely for discovering the sample clusters in T . Hence, the spurious correlation in S inconsistent with the clusters in T can be removed by ICON (Section 5).

Graph-based SSL methods [18, 44, 24] share a similar spirit with ICON, where they learn a classifier using the labeled samples that is consistent with the unlabelled sample similarities. However, they lack the invariant constraint in ICON, which can cause the failure cases (e.g., Figure 2c).

Dataset	OFFICE-HOME	ViSDA-2017	iWILDCAM	CAMELYON17	FMOW	POVERTYMAP	GLOBALWHEAT	OGB-MOLPCBA	CIVILCOMMENTS	AMAZON
Sample x	object image	object image	camera trap photo	tissue slide	satellite image	satellite image	wheat image	molecular graph	online comment	product review
Label y	65 categories	12 categories	182 species	tumor/not	62 land uses	asset wealth	wheat bbox	bioassays	toxic/not	5 review scores
Task	classification	classification	classification	classification	classification	regression	detection	classification	classification	classification
Source S	various*	synthetic images	photos from 243 traps	slides from hospital 1-3	images from 2002-2013	images in 14 countries	images in Europe	44,930 scaffold groups	online articles*	1,252 reviewers
Example S										
#Samples S	average 3,875	152,397	129,809	302,436	76,863	~10,000	2,943	350,343	269,038	245,502
Target T	various*	real photos	photos from 3215 traps	slides from hospital 5	images from 2016-2018	images in 5 countries	images across the world	43,793 scaffold groups	online articles*	1,334 reviewers
Example T										
#Samples T	average 3,875	55,388	819,120	600,030	173,208	261,396	42,445	517,048	1,551,515	268,761
Evaluation	average acc.	mean-class accuracy	macro-F1	acc.	worst-region acc.*	Pearson correlation* %	acc.	average precision	worst-group acc.*	10 th percentile acc.
Existing Methods	GVB		Empirical Risk Minimization (ERM)							
	70.4	75.3	47.0 / 32.2	90.6 / 82.0	60.6 / 34.8	65 / 48	77.8 / 51.0	- / 28.3	89.8 / 66.6	72.0 / 54.2
	TCM		CORAL							
	70.7	75.8	40.5 / 27.9	90.4 / 77.9	58.9 / 34.1	54 / 36	- / -	- / 26.6	- / -	71.7 / 53.3
	SENTRY		DANN							
	72.0	76.7	48.5 / 31.9	86.9 / 68.4	57.9 / 34.6	50 / 33	- / -	- / 20.4	- / -	71.7 / 53.3
	CST		Pseudo-Label							
	72.2	80.6	47.3 / 30.3	91.3 / 67.7	60.9 / 33.7	- / -	73.3 / 42.9	- / 19.7	90.3 / 66.9	71.6 / 52.3
	ToAlign		Noisy Student							
	72.7	77.8	47.5 / 32.1	93.2 / 86.7	61.3 / 37.8	61 / 42	78.1 / 46.8	- / 27.5	- / -	- / -
	FixBi		FixMatch							
	73.0	82.8	46.3 / 31.0	91.3 / 71.0	58.6 / 32.1	54 / 30	- / -	- / -	89.4 / 65.7	71.9 / 53.9
ATDOC		SwAV								
73.2	83.7	47.3 / 29.0	92.3 / 91.4	61.8 / 36.3	60 / 45	- / -	- / -	89.9 / 69.4	73.6 / 56.4	
ICON	75.8 +2.6	87.4 +3.7	50.6 / 34.5 +2.3	95.6 / 93.8 +2.4	62.2 / 39.9 +2.1	65 / 49 +1	78.6 / 52.3 +1.3	- / 28.3 +0.0	89.7 / 68.8 +1.9	71.9 / 54.7 +0.5

Table 1: Dataset details and the results of our ICON compared with existing methods. Details on * in Section 5.1. On WILDS 2.0 [49] datasets, models were evaluated on both the test data in S (first number) and in T (second number). The performance gain in T highlighted in red. “ERM (labeled T)” has full supervision in T . -/- means that the method is not applicable to the dataset. Other dataset details and the standard deviation of the results are in Appendix.

5 Experiment

5.1 Datasets

As shown in Table 1, we extensively evaluated ICON on 10 datasets with standard protocols [36, 49], including 2 classic ones: OFFICE-HOME [62], ViSDA-2017 [45], and 8 others from the recent WILDS 2.0 benchmark [49]. WILDS 2.0 offers large-scale UDA tasks in 3 modalities (image, text, graph) under real-world settings, *e.g.*, wildlife conservation, medical imaging and remote sensing. Notably, unlike the classic datasets where the same sample set in T is used in training (unlabelled) and testing, WILDS 2.0 provides unlabelled train set in T disjoint with the test set, which prevents explicit pattern mining of the test data. They also provide validation sets for model selection. For datasets in WILDS 2.0 benchmark, we drop their suffix -WILDS for simplicity (*e.g.*, denoting AMAZON-WILDS as AMAZON).

Details of * in Table 1: **OFFICE-HOME** has 4 domains: artistic images (Ar), clip art (Cl), product images (Pr), and real-world images (Rw), which forms 12 UDA tasks by permutation, and we only show Cl→Rw as an example in the table. **FMOW** considers the worst-case performance on different geographic regions, where the land use in Africa is significantly different. **POVERTYMAP** considers the lower of the Pearson correlations (%) on the urban or rural areas. **CIVILCOMENTS** correspond to a semi-supervised setting with no domain gap between S and T , and is evaluated on the worst-case performance among demographic identities.

On some WILDS 2.0 datasets, the test data may come from a separate domain different from the training target domain T , *e.g.*, iWILDCAM test data is captured by 48 camera traps disjoint from the 3215 camera traps in the training T . We highlight that ICON is still applicable on these datasets. This is because ICON works by learning the causal feature c and discarding the environmental feature e , and the theory holds as long as the definition of c and e is consistent across the training and test data, *e.g.*, iWILDCAM is about animal classification across the labeled, unlabeled, and test data.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
DANN [15] (2016)	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN [36] (2018)	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
SymNet [71] (2019)	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	74.5	52.6	82.7	67.6
MDD [72] (2019)	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
SHOT[30] (2020)	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
ALDA [7] (2020)	53.7	70.1	76.4	60.2	72.6	71.5	56.8	51.9	77.1	70.2	56.3	82.1	66.6
GVB [9] (2020)	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
TCM [69] (2021)	58.6	74.4	79.6	64.5	74.0	75.1	64.6	56.2	80.9	74.6	60.7	84.7	70.7
SENTRY [46] (2021)	61.8	77.4	80.1	66.3	71.6	74.7	66.8	63.0	80.9	74.0	66.3	84.1	72.2
CST [34] (2021)	59.0	79.6	83.4	68.4	77.1	76.7	68.9	56.4	83.0	75.3	62.2	85.1	73.0
ToAlign [66] (2021)	57.9	76.9	80.8	66.7	75.6	77.0	67.8	57.0	82.5	75.1	60.0	84.9	72.0
FixBi [42] (2021)	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
ATDOC [31] (2021)	60.2	77.8	82.2	68.5	78.6	77.9	68.4	58.4	83.1	74.8	61.5	87.2	73.2
SDAT [47] (2022)	58.2	77.1	82.2	66.3	77.6	76.8	63.3	57.0	82.2	74.9	64.7	86.0	72.2
MCC+NWD [6] (2022)	58.1	79.6	83.7	67.7	77.9	78.7	66.8	56.0	81.9	73.9	60.9	86.1	72.6
kSHOT* [57] (2022)	58.2	80.0	82.9	61.1	80.3	80.7	71.3	56.8	83.2	75.5	60.3	86.6	73.9
ICON (Ours)	63.3	81.3	84.5	70.3	82.1	81.0	70.3	61.8	83.7	75.6	68.6	87.3	75.8

Table 2: Break-down of the accuracies in each domain on OFFICE-HOME dataset [62]. *: kSHOT [57] additionally uses the prior knowledge on the percentage of samples in each class in the testing data. Published years are in the brackets after the method names.

5.2 Implementation Details

Feature Backbone. We used the followings pre-trained on ImageNet [48]: ResNet-50 [21] on OFFICE-HOME, VISDA-2017 and iWILDCAM, DenseNet-121 [23] on FMOW and FasterRCNN [16] on GLOBALWHEAT. We used DistilBERT [50] with pre-trained weights from the Transformers library on CIVILCOMMENTS and AMAZON. On CAMELYON17, we used DenseNet-121 [23] pre-trained by the self-supervised SwAV [5] with the training data in S and T . On POVERTYMAP and OGB-MOLPCBA with no pre-training available, we used multi-spectral ResNet-18 [21] and graph isomorphism network [68] trained with the labelled samples in the source domain S , respectively.

Self-training. A general form of the self-training loss \mathcal{L}_{st} in Eq. (3) is given below:

$$\mathcal{L}_{st} = \mathbb{E}_{\mathbf{x} \sim T} [\mathbb{1}(\max f(\mathbf{x}) > \tau) \text{CE}(f(\mathbf{x}), \text{argmax} f(\mathbf{x}))], \quad (4)$$

where τ is a confident threshold that discards low-confidence pseudo-labels, and $\text{CE}(\cdot, \cdot)$ is the cross-entropy loss. We detail the different implementations of Eq (4) in Appendix. ICON is agnostic to the choice of \mathcal{L}_{st} , and hence we choose the best performing self-training baseline for each dataset: FixMatch [55] on OFFICE-HOME and VISDA-2017, NoisyStudent [67] on iWILDCAM and FMOW, Pseudo-label [28] on CIVILCOMMENTS and AMAZON. ICON also does not rely on \mathcal{L}_{st} . We did not apply \mathcal{L}_{st} (*i.e.*, $\beta = 0$) on CAMELYON17, POVERTYMAP, GLOBALWHEAT and OGB-MOLPCBA where self-training does not work well.

Regression and Detection. For the regression task in POVERTYMAP, f is a regressor that outputs a real number. As the number of clusters in T is not well-defined, we directly used rank statistics to determine if two samples in T have the same label. For the BCE loss, we computed the similarity by replacing $f(\mathbf{x}_i)^\top f(\mathbf{x}_j)$ in Eq. (1) as $-(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2$. For detection in GLOBALWHEAT, we used the ICON objective in Eq. (3) to train the classification head without other modifications.

Other Details. We aim to provide a simple yet effective UDA method that works across data modalities, hence we did not use tricks tailored for a specific modality (*e.g.*, image), such as mixup [70] or extensive augmentations (16 are used in [13]). On classic UDA datasets, we used entropy minimization and SAM optimizer following CST [34]. On VISDA-2017, iWILDCAM, CAMELYON17 and FMOW, we pre-processed the features with UMAP [39] (output dimension as 50) and EqInv [63], which helps fulfill Assumption 1 (*i.e.*, highlight causal feature to improve clustering). ICON is also resource efficient—all our experiments can run on a single NVIDIA 2080Ti GPU.

5.3 Main Results

Classic UDA Benchmarks. In Table 1 (first 2 columns), ICON significantly outperforms the existing state-of-the-arts on OFFICE-HOME [62] and VISDA-2017 [45] by 2.6% and 3.7%, respectively. We include the breaks down of the performances of OFFICE-HOME in Table 2, where ICON wins in 10 out of 12 tasks, and significantly improves the hard tasks (*e.g.*, Ar→Cl, Rw→Cl). For VISDA-2017, we include more comparisons in Table 3. ICON beats the original competition winner (MT+16augs) [13] that averages predictions on 16 curated augmented views. ICON even outperforms all methods that use the much deeper ResNet-101 [21]. Notably, CST [34] and ATDOC [31] also aim to remove

Method	Backbone	Acc.
MT+16aug [13] (2018)	ResNet-50	82.8
MDD [72] (2019)	ResNet-50	77.8
GVB [9] (2020)	ResNet-50	75.3
TCM [69] (2021)	ResNet-50	75.8
SENTRY [46] (2021)	ResNet-50	76.7
CST [34] (2021)	ResNet-50	80.6
CAN [26] (2019)	ResNet-101	87.2
SHOT [30] (2020)	ResNet-101	82.9
FixBi [26] (2021)	ResNet-101	87.2
MCC+NWD [6] (2022)	ResNet-101	83.7
SDAC [47] (2022)	ResNet-101	84.3
kSHOT* [57] (2022)	ResNet-101	86.1
ICON (Ours)	ResNet-50	87.4

Table 3: Mean-class accuracy (Acc.) on VISDA-2017 Synthetic→Real task with the choice of feature backbone. *: details in Table 2 caption. Published years are in the brackets after the method names.

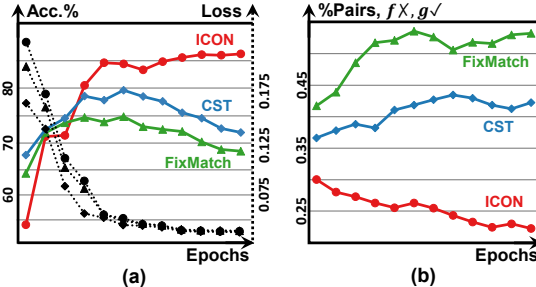


Figure 5: (a) Solid lines: test accuracy against training epochs for FixMatch, CST and ICON. Dotted black lines: average loss in each epoch with matching markers to denote the three methods. (b) Among the feature pairs where f fails, the percentage of pairs where g succeeds for the three methods.

the spurious correlation, and both attain competitive performance, validating the importance of this research objective. Yet their training scheme allows the correlation to persist (Section 4), hence ICON is more effective.

WILDS 2.0. In Table 1 (last 8 columns), ICON dominates the WILDS 2.0 leaderboard despite the challenging real-world data and stringent evaluation. For example, the animal images in iWILDCAM exhibit long-tail distribution that is domain-specific, because the rarity of species varies across locations. Hence under macro-F1 evaluation (sensitive to tail performance), the 2.3% improvement is noteworthy. On CIVILCOMMENTS with text data, ICON improves the worst performance across demographic groups by 2.2%, which promotes AI fairness and demonstrates the strength of ICON even under the SSL setting. On AMAZON, although ICON only improves 0.5% to reach 54.7%, the upper bound performance by training with labelled T (56.4%) is also low due to the strict evaluation metric. We also highlight that ICON increases the performance in T without sacrificing that in S (the first number). This empirically validates that the invariance objective of ICON resolves the tug-of-war scenario in Figure 2c. In particular, we tried implementing recent methods on this benchmark. Yet they are outperformed by ERM, or even not applicable (e.g., on text/graph data). This is in line with WILDS 2.0 organizers’ observation that ERM as a naive baseline usually has the best performance, which is exactly the point of WILDS 2.0 as a diagnosis of the ever-overlooked issues in UDA. Hence we believe that dominating this challenging leaderboard is a strong justification for ICON, and encourage more researchers to try this benchmark.

Failure Cases. ICON performance is relatively weak on POVERTYMAP and OGB-MOLPCBA with two possible reasons: 1) no pre-training available, the backbone is trained only on S and hence biased in the beginning, and 2) the ground-truth number of classes in T is not well-defined in both datasets, hence Assumption 1 in Section 3 may fail.

Learning Curve. In Figure 5a, we compare our ICON with two self-training baselines: FixMatch [55] and CST [34]. The test accuracy and training loss *w.r.t.* training epochs on VISDA-2017 is drawn in colored solid lines and black dotted lines, respectively. We observe that the training losses for all methods converge quickly. However, while the two baselines achieve higher accuracy in the beginning, their performances soon start to drop. In contrast, ICON’s accuracy grows almost steadily.

Effect of Supervision in T . To find the cause of the performance drop, we used the optimized g in Section 2.2 to probe the learned f in each epoch. Specifically, we sampled feature pairs $\mathbf{x}_1, \mathbf{x}_2$ in T and tested if the predictions of f, g are consistent with their labels, e.g., f succeeds if $\text{argmax} f(\mathbf{x}_1) = \text{argmax} f(\mathbf{x}_2)$ when $y_1 = y_2$ (vice versa for \neq). Then among the pairs where f failed, we computed the percentage of them where g succeeded, which is shown in Figure 5b. A large percentage implies that f disrespects the inherent distribution in T correctly identified by g . We observe that the percentage has a rising trend for the two baselines, i.e., they become more biased to the spurious correlations in S that do not hold in T . This also explains their dropping performance in Figure 5a. In particular, CST has a lower percentage compared to FixMatch, which suggests that it has some effects in removing the bias. In contrast, the percentage steadily drops for ICON, as we train f to respect the inherent distribution identified by g .

Method	OFFICE-HOME	VisDA-2017
FixMatch	69.1	76.6
FixMatch+CON	74.1	82.0
FixMatch+CON+INV	75.8	87.4
Cluster with $2 \times \#classes$	69.7	78.6
Cluster with $0.5 \times \#classes$	67.5	76.2
Cluster with k-NN	72.1	85.6

Table 4: Ablations on each ICON component. CON denotes the consistency loss in S and T . INV denotes the invariance constraint.

5.4 Ablations

Components. Recall that the main components of ICON include: 1) Consistency loss (CON) that leverages the supervision from the labels in S and clusters in T ; 2) the use of invariance constraint (INV) in Eq. (3); 2) clustering in T with $\#clusters = \#classes$. Their ablations are included in Table 4. From the first three lines, we observe that adding each component leads to consistent performance increase on top of the self-training baseline (FixMatch), which validates their effectiveness. From the next two lines, when $\#clusters \neq \#classes$, the model performs poorly, *e.g.*, using one cluster in Figure 2b (*i.e.*, $0.5 \times \#classes$), the invariant classifier needs to predict the two red circles in S as dissimilar (two classes) and the two blue ones in T as similar (one cluster), which in fact corresponds to the erroneous red line (*i.e.*, predictions in T are similarly ambiguous). In the last line, we tried k-means for clustering instead of rank-statistics, leading to decreased performance. We postulate that rank-statistics is better because its online clustering provides up-to-date cluster assignments. In the future, we will try other more recent online clustering methods, *e.g.*, an optimal transport-based method [5]. Overall, the empirical findings validate the importance of the three components, which are consistent with the theoretical analysis in Section 3.

Hyperparameters. We used two hyperparameters in our experiments: 1) the weight of the self-training loss denoted as α ; 2) the invariance penalty strength β in the REX [27] implementation of Eq. (3). Their ablations are in Figure 6 on the widely used classic UDA datasets. We observe that $\alpha \in [0.5, 1.0]$ and $\beta \in [0.1, 0.25]$ works well on both datasets. In particular, it is standard in the IRM community to choose a small β (*e.g.*, 0.1). Then, tuning α follows the same process as other self-training UDA methods. Hence the hyperparameter search is easy.

Confusion Matrix concretely shows how a method is biased to the correlations in the source domain S . We compare the state-of-the-art self-training method CST with ICON in Figure 7. The matrix on the left reveals that CST predicted most “knife”’s as “skateboard”’s (“board” for short), and some “truck”’s as “car”’s. On the right, we see that ICON predicts “knife” accurately and also improves the accuracy on “truck”.

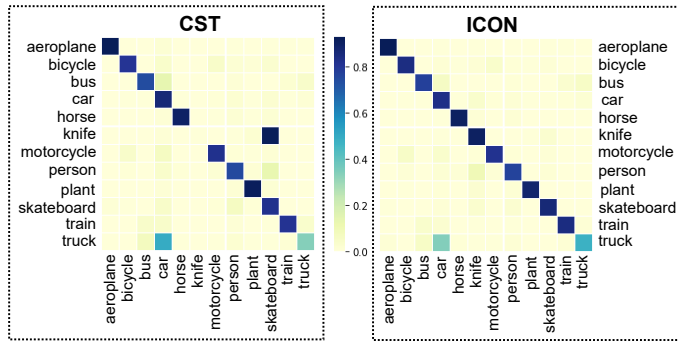


Figure 7: CST and ICON confusion matrix on VisDA-2017.

Grad-CAM Visualization. Why does CST predict “knife” in T as “board”? To answer this, in Figure 8, we visualized the GradCAM [52] of CST and ICON trained on VisDA-2017. In the first row, we notice that the self-training based CST predicts the two categories perfectly by leveraging their apparent differences: the curved blade versus the flat board ending with semi-circles. However, this leads to poor generalization on “knife” in T , where the background (image 1, 4), the knife component (2) or even the blade (3) can have board-like appearances. Moreover, biased to the standard “board” look in S , CST can leverage environmental feature for predicting “board” in T , *e.g.*, under side-view (5) or slight obstruction (6), which is prone to errors (*e.g.*, 8). As the board-like shape alone is non-discriminative for the two categories in T , in the second row, the cluster head g additionally leverages knife handles and wheels to distinguish them. In the third row, our ICON combines the supervision in S (first row) and T (second row) to learn an invariant classifier f . In S , ICON focuses on the overall knife shape instead of just blade, and incorporates wheels to predict “board”. In T , by

		OFFICE-HOME						VisDA-2017			
$\beta \backslash \alpha$	α	0.25	0.5	1.0	2.0	$\beta \backslash \alpha$	α	0.25	0.5	1.0	2.0
	0.05	0.05	73.8	75.0	75.5		73.9	0.05	0.05	85.2	86.2
0.1	0.1	73.9	75.4	75.8	74.4	0.1	0.1	85.6	86.4	86.7	82.6
0.25	0.25	70.2	72.8	75.2	74.5	0.25	0.25	76.8	81.2	87.4	86.3

Figure 6: Ablations of α, β on the classic UDA datasets.

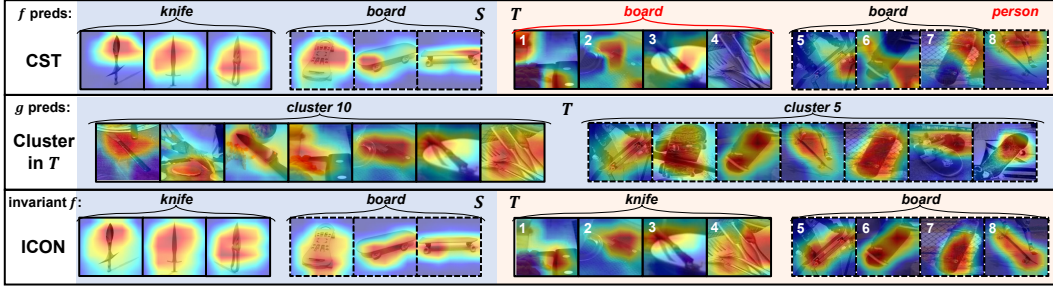


Figure 8: GradCAM on VisDA-2017 [45] using CST [34], the cluster head g and the proposed ICON. Ground-truth knives and boards are in solid and dotted boxes, respectively. Predictions from f , g are on top of each row. Red denotes wrong predictions. Pale blue and red background denotes training and testing, respectively. Note that g is trained without supervision in T .

comparing images with the same number in the first and third row, we see that ICON focuses more on the discriminative parts of the objects (e.g., handles and wheels), hence generalizes better in T .

6 Conclusion

We presented a novel UDA method called Invariant CONSistency learning (ICON) that removes the spurious correlation specific to the source domain S to generalize in the target domain T . ICON pursues an invariant classifier that is simultaneously consistent with the labels in S and the clusters in T , hence removing the spurious correlation inconsistent in T . We show that ICON achieves generalization in T with a theoretical guarantee. Our theory is verified by extensive evaluations on both the classic UDA benchmarks and the challenging WILDS 2.0 benchmark, where ICON outperforms all the conventional methods. We will seek more principled pre-training paradigms that disentangle the causal features and relax the assumption of knowing the class numbers in T to improve regression tasks.

7 Limitation and Broader Impact

Limitation. Our approach is based on the assumptions in Section 4, *i.e.*, the classes in T are separated by clusters, and there is enough diversity in T such that causal feature c is the only invariance. Assumption 1 is essentially the clustering assumption, which is a necessary assumption for learning with unlabeled data [61]. In fact, UDA methods (and semi-supervised learning in general) commonly leverage feature pre-training and data augmentations to help fulfill the assumptions. One can also use additional pre-trained knowledge from foundation models, or deploy improved clustering algorithm (e.g., an optimal transport-based method [5]). Assumption 2 is necessary to rule out corner cases, e.g., in Figure 4b, without additional prior, there is no way for any UDA method to tell if each one of the blue cluster (unlabeled) should be pseudo-labeled as "0" or "1". In practice, these corner cases can be avoided by collecting diverse unlabeled data, which is generally abundant, e.g., collecting unlabeled street images from a camera-equipped vehicle cruise.

Broader Impact. Our work aims to capture the causal feature in the unsupervised domain adaptation setting. In particular, we significantly improve the state-of-the-art on WILDS 2.0 benchmark, which includes practical tasks that benefit wildlife preservation, tumour classification in healthcare, remote sensing, vision in agriculture, *etc.* The pursue of causal feature can produce more robust, transparent and explainable models, broadening the applicability of ML models and promoting fairness in AI.

8 Acknowledgement

The authors would like to thank all reviewers and ACs for their constructive suggestions. This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-022), MOE Academic Research Fund Tier 2 (MOE2019-T2-2-062), Alibaba-NTU Singapore Joint Research Institute, the A*STAR under its AME YIRG Grant (Project No.A20E6c0101), and the Lee Kong Chian (LKC) Fellowship fund awarded by Singapore Management University.

Appendix

This appendix is organized as follows:

- Section A provides more details about our algorithms. In particular, Section A.1 describes the rank-statistics clustering algorithm. Section A.2 describes the REX implementation of invariance in Eq. (2). Section A.3 discusses the two implementation of the self-training used in our experiments: FixMatch [55] and Noisy Student [67].
- Section B gives the preliminaries of group theory.
- Section C gives the full proof of our theorem.
- Section D provides more details on the datasets, our feature backbone, how we use seeds in the evaluation, self-training details, and other implementation details.
- Section E shows more comparisons and standard deviations on WILDS 2.0 benchmark [49], VISDA-2017 [45], and the break-down of accuracy in each domain for OFFICE-HOME [62].
- Codes are also provided, which include the training and testing scripts on the two classic datasets, WILDS 2.0 benchmark, and the live demo code for Figure 2 using MNIST [10]. The setup instructions and commands used in our experiments are included in the README.md file.

Abbreviation/Symbol	Meaning
<i>Abbreviation</i>	
DA	Domain Adaptation
UDA	Unsupervised Domain Adaptation
BCE	Binary Cross-Entropy
CE	Cross-Entropy
<i>Symbol in Algorithm</i>	
S	Source domain
T	Target domain
\mathbf{c}	Causal feature
\mathbf{e}	Environmental feature
Φ	(Hidden) Data generator
\mathbf{x}	Sample feature
y	Sample label
θ	Parameters of the backbone
N, M	Number of samples in S, T , respectively
f	Classification head
g	Cluster head
\mathcal{L}_{st}	Self-training loss
α	Self-training loss weight
β	Invariance loss weight
<i>Symbol in Theory</i>	
\mathcal{G}	Group of semantics
\mathcal{X}	Feature space
\mathcal{H}	Subgroup of \mathcal{G} that transforms environmental feature \mathbf{e}
$g \circ \mathbf{x}$	Group action

Table A1: List of abbreviations and symbols used in the paper.

A Additional Details on Algorithm

A.1 Rank-Statistics Clustering

Recall that in UDA training, a set of M unlabelled sample features $\{\mathbf{x}_i\}_{i=1}^M$ (features extracted by the backbone θ) from C classes is available in the target domain T . Rank-statistics clustering algorithm [20] learns a cluster head g , whose output $g(\mathbf{x}_i)$ are the softmax-normalized probabilities for \mathbf{x}_i belonging to each of the C clusters. We use the pair-wise binary cross-entropy (BCE) loss to learn g :

$$\min_g \mathbb{E}_{\mathbf{x}_i, \mathbf{x}_j \in T} - y_{i,j} \log(g(\mathbf{x}_i)^\top g(\mathbf{x}_j)) - (1 - y_{i,j}) \log(1 - g(\mathbf{x}_i)^\top g(\mathbf{x}_j)), \quad (\text{A5})$$

where $y_{i,j} = 1$ if the top-5 values of $\mathbf{x}_i, \mathbf{x}_j$ have the same indices, and 0 otherwise. In practice, the above optimization can be implemented efficiently in a batch-wise sampling manner, where we compute the pair-wise BCE loss over each sample batch, instead of the entire unlabelled sample set in T .

A.2 REx

REx circumvents the challenging bi-level optimization from line 2 of Eq. (2) by adding the following invariant constraint loss \mathcal{L}_{REx} to line 1:

$$\mathcal{L}_{REx} = \beta \text{Var}(\{\text{BCE}(S, f), \text{BCE}(T, f)\}), \quad (\text{A6})$$

which computes the variance between $\text{BCE}(S, f)$ and $\text{BCE}(T, f)$ with a trade-off parameter β . We performed ablations on β in Section 5.4. Please refer to [27] for a theoretical explanation.

A.3 Self-Training

We adopt three implementations of self-training, and use a trade-off parameter α to balance the weight of self-training loss.

FixMatch [55]. Given the classifier f and a pair of features \mathbf{x}, \mathbf{x}' corresponding to a sample’s feature under weak and strong augmentations, respectively, the FixMatch self-training loss \mathcal{L}_{fm} is given by:

$$\mathcal{L}_{st} = \mathbb{1}(\text{argmax} f(\mathbf{x}) > \tau) \text{CE}(\mathbf{x}', \text{argmax} f(\mathbf{x})), \quad (\text{A7})$$

where $\mathbb{1}$ is the indicator function that returns 1 if and only if the condition in the bracket is true, τ is the confident threshold, and CE denotes the cross-entropy loss. In our experiments, we use the standard [34] fixed threshold $\tau = 0.97$ for the two classic datasets, and follow the standard threshold settings in WILDS 2.0 without searching. Hence we did not perform ablations on τ .

Noisy Student [67]. Compared to FixMatch, Noisy Student has the following differences: 1) Instead of minimizing the self-training loss in an online fashion, Noisy Student generates the pseudo-labels $\text{argmax} f(\mathbf{x})$ for all samples in T at the start of epoch using the current model, and the model is updated for one epoch using the fixed pseudo-labels. 2) It does not perform selection by confident threshold (*i.e.*, $\mathbb{1}(\cdot)$ in Eq. (A7)), and hence the pseudo-labels for all samples participate in training. 3) Unlike FixMatch that relies on the use of strong augmentations to prevent model collapsing, Noisy Student can still be used when only weak augmentations are well-defined on the dataset, allowing it to run on more datasets (Table 1).

Pseudo-Label [28]. We adopt the implementation by WILDS 2.0 benchmark:

$$\mathcal{L}_{st} = \mathbb{1}(\text{argmax} f(\mathbf{x}') > \tau) \text{CE}(\mathbf{x}', \text{argmax} f(\mathbf{x}')), \quad (\text{A8})$$

which uses only one augmented view of each sample, instead of the weak and strong augmented views in FixMatch. Note that the augmentation is optional in Eq. (A8), *e.g.*, on modalities where data augmentation is not available/difficult (*e.g.*, text data), one can use Pseudo-Label without augmentation. This implementation also linearly scales up the self-training weight from 0 to the specified α through the course of 40% total training steps.

B Group Theory Preliminaries

A group is a set together with a binary operation, which takes two elements in the group and maps them to another element. For example, the set of integers is a group under the binary operation of plus. We formalize the notion through the following definition.

Binary Operation. A binary operation \cdot on a set \mathcal{S} is a function mapping $\mathcal{S} \times \mathcal{S}$ into \mathcal{S} . For each $(s_1, s_2) \in \mathcal{S} \times \mathcal{S}$, we denote the element $\cdot(s_1, s_2)$ by $s_1 \cdot s_2$.

Group. A group $\langle \mathcal{G}, \cdot \rangle$ is a set \mathcal{G} , closed under a binary operation \cdot , such that the following axioms hold:

1. *Associativity.* $\forall g_1, g_2, g_3 \in \mathcal{G}$, we have $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$.
2. *Identity Element.* $\exists e \in \mathcal{G}$, such that $\forall g \in \mathcal{G}$, $e \cdot g = g \cdot e = g$.
3. *Inverse.* $\forall g \in \mathcal{G}$, $\exists g' \in \mathcal{G}$, such that $g \cdot g' = g' \cdot g = e$.

The binary operator \cdot is often omitted, *i.e.*, we write $g_1 g_2$ instead of $g_1 \cdot g_2$. Groups often arise as transformations of some space, such as a set, vector space, or topological space. Consider an equilateral triangle. The set of clockwise rotations *w.r.t.* its centroid to retain its appearance forms a group $\{60^\circ, 120^\circ, 180^\circ\}$, with the last element corresponding to an identity mapping. We say this group of rotations act on the triangle, formally defined below.

Group Action. Let \mathcal{G} be a group and \mathcal{S} be a set. An action of \mathcal{G} on \mathcal{S} is a map $\pi : \mathcal{G} \rightarrow \text{Hom}(\mathcal{S}, \mathcal{S})$ so that $\pi(e) = \text{id}_{\mathcal{S}}$ and $\pi(g)\pi(h) = \pi(gh)$, where $g, h \in \mathcal{G}$. $\forall g \in \mathcal{G}, s \in \mathcal{S}$, denote $\pi(g)(s)$ as $g \circ s$.

Direct Product of Group. Let $\mathcal{G}_1, \dots, \mathcal{G}_n$ be groups with the binary operation \cdot . Let $a_i, b_i \in \mathcal{G}_i$ for $i \in \{1, \dots, n\}$. Define $(a_1, \dots, a_n) \cdot (b_1, \dots, b_n)$ to be the element $(a_1 \cdot b_1, \dots, a_n \cdot b_n)$. Then $\mathcal{G}_1 \times \dots \times \mathcal{G}_n$ is the direct product of the groups $\mathcal{G}_1, \dots, \mathcal{G}_n$ under the binary operation \cdot .

Quotient Group. Let $\phi : \mathcal{G} \rightarrow \mathcal{G}'$ mapping to some group \mathcal{G}' be a group homomorphism, *i.e.*, $\phi(gh) = \phi(g)\phi(h)$, such that $\mathcal{H} = \{h \in \mathcal{G} \mid \phi(h) = e'\}$. Let $g \in \mathcal{G}$. Then the coset $a\mathcal{H}$ of \mathcal{H} is defined as $\{g \in \mathcal{G} \mid \phi(g) = \phi(a)\}$, and the cosets of \mathcal{H} form a quotient group \mathcal{G}/\mathcal{H} .

Theorem. Let $\mathcal{G} = \mathcal{K} \times \mathcal{H}$ be the direct product of group \mathcal{K} and \mathcal{H} . Let $\tilde{\mathcal{H}} = \{(e, h) \mid h \in \mathcal{H}\}$. Then $\mathcal{G}/\tilde{\mathcal{H}}$ is isomorphic to \mathcal{K} in a natural way.

In our formulation, we consider a semantic space $\mathcal{S} = \mathcal{C} \times \mathcal{E}$ as the Cartesian product of the causal feature space \mathcal{C} and the environmental feature space \mathcal{E} . The group \mathcal{G} acting on \mathcal{S} corresponds to all semantic transformations, *e.g.*, $g \in \mathcal{G}$ may correspond to “turn darker” of the digit color. In particular, \mathcal{G} has a corresponding direct product decomposition $\mathcal{G} = \mathcal{K} \times \mathcal{H}$, where \mathcal{K} acts on \mathcal{C} and \mathcal{H} acts on \mathcal{E} . With slight abuse of notation, we write \mathcal{K} as \mathcal{G}/\mathcal{H} with the above theorem.

\mathcal{G} -Equivariant Map. Let \mathcal{G} be a group acting on the set \mathcal{S} . Then $\Phi : \mathcal{S} \rightarrow \mathcal{X}$ mapping to the set \mathcal{X} is said to be a \mathcal{G} -equivariant map if $f(g \circ s) = g \circ f(s)$ for all $g \in \mathcal{G}, s \in \mathcal{S}$.

In our work, we consider the mapping from the causal and environmental features to the learned sample features as a \mathcal{G} -equivariant map, hence we write $g \circ \mathbf{x}$ for $g \in \mathcal{G}, \mathbf{x} \in \mathcal{X}$ as the transformed feature from \mathbf{x} by g .

Group Representation. Let \mathcal{G} be a group. A representation of \mathcal{G} (or \mathcal{G} -representation) is a pair (π, \mathcal{X}) , where \mathcal{X} is a vector space and $\pi : \mathcal{G} \rightarrow \text{Hom}_{\text{vect}}(\mathcal{X}, \mathcal{X})$ is a group action, *i.e.*, for each $g \in \mathcal{G}$, $\pi(g) : \mathcal{X} \rightarrow \mathcal{X}$ is a linear map.

Intuitively, each $g \in \mathcal{G}$ corresponds to a linear map, *i.e.*, a matrix \mathbf{M}_g that transforms a vector $\mathbf{x} \in \mathcal{X}$ to $\mathbf{M}_g \mathbf{x} \in \mathcal{X}$.

C Proof of Theorem

We include our assumptions and theorem in the main paper below for easy reference.

Assumption (Identifiability of \mathcal{G}/\mathcal{H}).

- 1) $\forall \mathbf{x}_i, \mathbf{x}_j \in T$, $y_i = y_j$ iff $\mathbf{x}_i \in \{h \circ \mathbf{x}_j \mid h \in \mathcal{H}\}$;
- 2) There exists no linear map $l : \mathcal{X} \rightarrow \mathbb{R}$ such that $l(\mathbf{x}) > l(h \circ \mathbf{x})$, $\forall \mathbf{x} \in S, h \circ \mathbf{x} \in T$ and $l(g \circ \mathbf{x}) < l(gh \circ \mathbf{x})$, $\forall g \circ \mathbf{x} \in S, gh \circ \mathbf{x} \in T$, where $h \neq e \in \mathcal{H}, g \neq e \in \mathcal{G}/\mathcal{H}$.

Theorem. When the above assumptions hold, ICON (optimizing Eq. 2) learns a backbone θ mapping to a feature space \mathcal{X} that generalizes under $\mathcal{G}/\mathcal{H} \times \mathcal{H}$. In particular, the learned f is the optimal classifier in S and T .

Proof sketch. We will start by proving the binary classification case: we first show that the classifier satisfying the invariance objective in Eq. (2) (i.e., line 2) separates the two classes among all samples in S and T . Then by minimizing line 1, samples are mapped to two class-specific features (i.e., one feature for each class). Finally, by minimizing CE using labelled samples in S , we learn the optimal classifier that assigns the correct labels to all samples in S and T . We expand the binary classification case to C -class classification by formulating it as C 1-versus- $(C - 1)$ binary classification problems.

Proof. Consider a binary classification problem, where f outputs a single probability of belong to the first class, $\mathcal{G}/\mathcal{H} = \{g, e\}$, i.e., $\mathbf{x}, g \circ \mathbf{x}$ are from different classes for all $\mathbf{x} \in \mathcal{X}$. Let f satisfies the invariance objective in Eq. (2), i.e., $f \in \operatorname{argmin}_{\bar{f}} \text{BCE}(S, \bar{f}) \cap \operatorname{argmin}_{\bar{f}} \text{BCE}(T, \bar{f})$. As $\text{BCE}(S, f)$ is computed using the ground-truth labels, we must have $f(\mathbf{x}_1) \rightarrow 1, f(\mathbf{x}_2) \rightarrow 0$ or $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1, \forall \mathbf{x}_1 \in \mathcal{H}(\mathbf{x}) \cap S, \mathbf{x}_2 \in \mathcal{H}(g \circ \mathbf{x}) \cap S$, where $\mathbf{x} \in \mathcal{X}$. To see this, we shall expand $\text{BCE}(S, f)$ on a pair of samples $\mathbf{x}, \mathbf{x}' \in S$ with label y, y' :

$$-\mathbb{1}(y = y') \log(\hat{\mathbf{y}}^\top \hat{\mathbf{y}}') - \mathbb{1}(y \neq y') \log(1 - \hat{\mathbf{y}}^\top \hat{\mathbf{y}}'), \quad (\text{A9})$$

where $\mathbb{1}$ is the indicator function, and $\hat{\mathbf{y}}$ is given by $(f(\mathbf{x}), 1 - f(\mathbf{x}))$ (similarly for $\hat{\mathbf{y}}'$). Note that it is not difficult to show that $\hat{\mathbf{y}}^\top \hat{\mathbf{y}}' \in [0, 1]$, where the lower (or upper) bound is attained when they take different (or same) values from $(0, 1), (1, 0)$. As f minimizes $\text{BCE}(S, \bar{f})$, it minimizes the loss for each $\mathbf{x}, \mathbf{x}' \in S$: if $y = y', \hat{\mathbf{y}}^\top \hat{\mathbf{y}}' \rightarrow 1$, and if $y \neq y', \hat{\mathbf{y}}^\top \hat{\mathbf{y}}' \rightarrow 0$. This is only possible when the aforementioned condition holds. In the target domain T , although there is no ground-truth label, we prove that the evaluation of $\mathbb{1}(y = y')$ (or \neq) using the cluster labels is always the same as using ground-truth class labels, for all $\mathbf{x}, \mathbf{x}' \in T$ given Assumption 1: 1) suppose that $\mathbf{x} \in \{h \circ \mathbf{x}' \mid h \in \mathcal{H}\}$, then by the sufficient condition of Assumption 1, $\mathbb{1}(y = y') = 1$, reflecting that they are from the same class; 2) suppose that $\mathbf{x} \notin \{h \circ \mathbf{x}' \mid h \in \mathcal{H}\}$, then by the contra-position of the necessary condition, $\mathbb{1}(y \neq y') = 1$, reflecting that they are from the different classes. Hence in the same way as S , we can show that $f(\mathbf{x}_1) \rightarrow 1, f(\mathbf{x}_2) \rightarrow 0$ or $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1, \forall \mathbf{x}_1 \in \mathcal{H}(\mathbf{x}) \cap T, \mathbf{x}_2 \in \mathcal{H}(g \circ \mathbf{x}) \cap T$. Now considering the samples in S and T . Denote $D = S \cup T$. There are two possibilities: **1)** $f(\mathbf{x}_1) \rightarrow 1, f(\mathbf{x}_2) \rightarrow 0$ or $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1, \forall \mathbf{x}_1 \in \mathcal{H}(\mathbf{x}) \cap D, \mathbf{x}_2 \in \mathcal{H}(g \circ \mathbf{x}) \cap D$; **2)** $f(\mathbf{x}_1) \rightarrow 1, f(\mathbf{x}_2) \rightarrow 0$ (or $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1$), $\forall \mathbf{x}_1 \in \mathcal{H}(\mathbf{x}) \cap S, \mathbf{x}_2 \in \mathcal{H}(g \circ \mathbf{x}) \cap S$ and $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1$ (or $f(\mathbf{x}_1) \rightarrow 0, f(\mathbf{x}_2) \rightarrow 1$), $\forall \mathbf{x}_1 \in \mathcal{H}(\mathbf{x}) \cap T, \mathbf{x}_2 \in \mathcal{H}(g \circ \mathbf{x}) \cap T$. Case 2 corresponds to the horizontal black line in Figure 4b. However, it contradicts with Assumption 2. Hence the only possibility is case 1, where f linearly separates the two classes in D , i.e., all samples in S and T . In particular, as Assumption 2 prevents the failure case where f separates the samples not based on the class (or causal feature c), it intuitively means that the only invariance between S and T is c . Overall, this shows that a linear map $f \in \operatorname{argmin}_{\bar{f}} \text{BCE}(S, \bar{f}) \cap \operatorname{argmin}_{\bar{f}} \text{BCE}(T, \bar{f})$ separates the two classes in D .

We will now analyze the general case of c -class classification. Specifically, $\mathbf{w} \in \mathbb{R}^{c \times d}$ corresponds to a linear map that transforms each sample feature $\mathbf{x} \in \mathbb{R}^d$ to the c -dimensional logits. The classifier f parameterized by \mathbf{w} outputs the softmax-normalized logits as the probability of belong to each of the c class. Without loss of generality, we consider the case where $\|\mathbf{w}_i\| = \|\mathbf{w}_j\|$ for $i, j \in \{1, \dots, c\}$ and $\|\mathbf{x}\| = M$ for all $\mathbf{x} \in \mathcal{X}$, e.g., through the common l2-regularization and feature normalization. From the previous analysis, we prove that a linear classifier satisfying the invariance objective separates the two classes in D . We can generalize the results to multi-class classification by considering it as C 1-versus- $(C - 1)$ binary classification problem (e.g., separating class 1 samples and the rest of samples). We can further show that minimizing line 1 of Eq. (2) leads to a feature space \mathcal{X} that generalizes under $\mathcal{G}/\mathcal{H} \times \mathcal{H}$. In particular, to minimize $\text{BCE}(S, f) + \text{BCE}(T, f)$ (expanded in Eq. (A9)), the prediction p of f on the correct class is maximized (i.e., $p \rightarrow 1$). Specifically, the probability of belonging to class $i \in \{1, \dots, c\}$ for a feature \mathbf{x} using softmax normalization is given by:

$$\frac{\exp(\mathbf{w}_i^\top \mathbf{x})}{\sum_{j=1}^c \exp(\mathbf{w}_j^\top \mathbf{x})}. \quad (\text{A10})$$

We show that to maximize p , a sample feature in class i is given by $M \bar{\mathbf{w}}_i$, where $M \rightarrow \infty, \bar{\mathbf{w}}_i$ is from normalizing \mathbf{w}_i , and for each $i \neq j \in \{1, \dots, c\}, \mathbf{w}_i^\top \mathbf{w}_j \leq 0$: the first condition is easily derived by maximizing $\exp(\mathbf{w}_i^\top \mathbf{x}) \rightarrow \infty$ under $\|\mathbf{x}\| = M$; the second condition is derived by limiting $p \rightarrow 1$, i.e., $\exp(\mathbf{w}_j^\top \mathbf{x}) = 0$ or $\exp(\mathbf{w}_j^\top \mathbf{x}) \rightarrow -\infty$. Overall, this shows that minimizing Eq. (2) leads to \mathcal{X}

where the samples in class i are mapped to the same feature $M\bar{\mathbf{w}}_i$, and samples in different classes are mapped to different features as $\mathbf{w}_i \neq \mathbf{w}_j$ for $i \neq j$ (from $\mathbf{w}_i^T \mathbf{w}_j \leq 0$). The classifier f is optimal by predicting the probability of each feature in S and T belonging to its ground-truth class as 1. Note that to further equip \mathcal{X} with the equivariant property as introduced in Section 1, one can leverage a specially designed backbone θ (e.g., equivariant neural networks [8]). Hence we complete the proof of our Theorem.

D Dataset and Implementations

Additional Dataset Details. For datasets in WILDS 2.0 benchmark, we drop their suffix -WILDS for simplicity in the main text (e.g., denoting AMAZON-WILDS as AMAZON). For IWILDCAM, the test data comes from a disjoint set of 48 camera traps (i.e., different from the 3215 camera traps in the training T). For POVERTYMAP, each sample is a multi-spectral satellite image with 8 channels. OGB-MOLPCBA corresponds to a multi-task classification setting, where each label y is a 128-dimensional binary vector, representing the binary outcomes of the biological assay results. y could contain NaN values, i.e., the corresponding biological assays were not performed on the given molecule. Hence there is no way to determine the ground-truth class numbers in this dataset. In AMAZON, S and T each contains a disjoint set of reviewers, and models are evaluated at their performance on the reviewer at the 10th percentile.

Feature Backbone. We used the followings pre-trained on ImageNet [48]: ResNet-50 [21] on 2 classic datasets and IWILDCAM, DenseNet-121 [23] on FMOW and Faster-RCNN [16] on GLOBALWHEAT. We used DistilBERT [50] with pre-trained weights from the Transformers library on CIVILCOMMENTS and AMAZON. On CAMELYON17, we used DenseNet-121 [23] pre-trained by the self-supervised SwAV [5] with the training data in S and T . On POVERTYMAP and OGB-MOLPCBA with no pre-training available, we used multi-spectral ResNet-18 [21] and graph isomorphism network [68] trained with the labelled samples in the source domain S , respectively.

Seeds. We followed the standard evaluation protocol: POVERTYMAP provides 5 different cross-validation folds that split the dataset differently. Hence we used a single seed (0), and average the performances on the 5 folds. We evaluated our model on CAMELYON17 with 10 seeds (0-9), and CIVILCOMMENTS with 5 seeds (0-4). We used 3 seeds (0-2) on other datasets in WILDS 2.0. We fixed the seed for Python, NumPy and PyTorch as per standard.

Self-Training Details. We used FixMatch [55] on the two classic UDA datasets. On the WILDS 2.0 datasets with text data, we used pseudo-label method [28], which is similar to FixMatch, but without the use of weak and strong augmentation. On other WILDS 2.0 datasets, we used Noisy Student [67] due to its superior performance.

Highlight causal feature c . We use two techniques to highlight the causal feature c , such that Assumption 1 is better fulfilled: 1) On large scale datasets VISDA-2017 and IWILDCAM, we deployed UMAP [39] to reduce the dimension of T features to 50 before clustering them. UMAP is empirically validated to faithfully capture the local data structure (e.g., high k-NN classification accuracy) as well as the global structure (e.g., more meaningful global relationships between different classes) in a low-dimensional space. Such structural information is however elusive in the original high-dimensional space due to the curse of dimensionality. 2) On image datasets VISDA-2017, IWILDCAM, CAMELYON and FMOW, we deploy EqInv [63] to further pursue the causal feature.

Other Implementation Details. For experiments on the classic UDA datasets, we modified the open-source code base of CST [34] (<https://github.com/Liuhong99/CST>). For experiments on WILDS 2.0 datasets, we implemented ICON on the official code base (<https://github.com/p-lambda/wilds>). On the classic UDA datasets, we followed the hyperparameter settings of CST except for the self-training weight, which we did an ablation in Figure 6. On WILDS 2.0 benchmark, we followed the released hyperparameter settings in their original experiments (<https://worksheets.codalab.org/worksheets/0x52cea64d1d3f4fa89de326b4e31aa50a>). In practice, we removed the $BCE(S, f)$ from line 1 of Eq. (2), as $CE(S, f)$ alone is enough to achieve consistency *w.r.t.* labels in S . We also selected sample pairs with confident binary label to compute $BCE(T, f)$ in line 1. Please refer to the attached code and the README.md for additional details.

	IWILDCAM2020-WILDS (Unlabeled extra, macro F1)		FMoW-WILDS (Unlabeled target, worst-region acc)	
	In-distribution	Out-of-distribution	In-distribution	Out-of-distribution
ERM (-data aug)	46.7 (0.6)	30.6 (1.1)	59.3 (0.7)	33.7 (1.5)
ERM	47.0 (1.4)	32.2 (1.2)	60.6 (0.6)	34.8 (1.5)
CORAL	40.5 (1.4)	27.9 (0.4)	59.3 (0.7)	33.7 (1.5)
DANN	48.5 (2.8)	31.9 (1.4)	57.9 (0.8)	34.6 (1.7)
Pseudo-Label	47.3 (0.4)	30.3 (0.4)	60.9 (0.5)	33.7 (0.2)
FixMatch	46.3 (0.5)	31.0 (1.3)	58.6 (2.4)	32.1 (2.0)
Noisy Student	47.5 (0.9)	32.1 (0.7)	61.3 (0.4)	37.8 (0.6)
SwAV	47.3 (1.4)	29.0 (2.0)	61.8 (1.0)	36.3 (1.0)
ERM (fully-labeled)	54.6 (1.5)	44.0 (2.3)	65.4 (0.4)	58.7 (1.4)
ICON (Ours)	50.6 (1.3)	34.5 (1.4)	62.2 (0.4)	39.9 (1.1)
	CAMELYON17-WILDS (Unlabeled target, avg acc)		POVERTYMAP-WILDS (Unlabeled target, worst U/R corr)	
	In-distribution	Out-of-distribution	In-distribution	Out-of-distribution
ERM (-data aug)	85.8 (1.9)	70.8 (7.2)	0.65 (0.03)	0.48 (0.04)
ERM	90.6 (1.2)	82.0 (7.4)	0.66 (0.04)	0.48 (0.05)
CORAL	90.4 (0.9)	77.9 (6.6)	0.54 (0.10)	0.36 (0.08)
DANN	86.9 (2.2)	68.4 (9.2)	0.50 (0.07)	0.33 (0.10)
Pseudo-Label	91.3 (1.3)	67.7 (8.2)	-	-
FixMatch	91.3 (1.1)	71.0 (4.9)	0.54 (0.11)	0.30 (0.11)
Noisy Student	93.2 (0.5)	86.7 (1.7)	0.61 (0.07)	0.42 (0.11)
SwAV	92.3 (0.4)	91.4 (2.0)	0.60 (0.13)	0.45 (0.05)
ICON (Ours)	95.6 (0.2)	93.8 (0.3)	0.65 (0.05)	0.49 (0.04)
	GLOBALWHEAT-WILDS (Unlabeled target, avg domain acc)		OGB-MOLPCBA (Unlabeled target, average AP)	
	In-distribution	Out-of-distribution	In-distribution	Out-of-distribution
ERM	77.8 (0.2)	51.0 (0.7)	-	28.3 (0.1)
CORAL	-	-	-	26.6 (0.2)
DANN	-	-	-	20.4 (0.8)
Pseudo-Label	73.3 (0.9)	42.9 (2.3)	-	19.7 (0.1)
Noisy Student	78.1 (0.3)	46.8 (1.2)	-	27.5 (0.1)
ICON (Ours)	78.6 (0.0)	52.3 (0.2)	-	28.3 (0.0)
	CIVILCOMMENTS-WILDS (Unlabeled extra, worst-group acc)		AMAZON-WILDS (Unlabeled target, 10th percentile acc)	
	In-distribution	Out-of-distribution	In-distribution	Out-of-distribution
ERM	89.8 (0.8)	66.6 (1.6)	72.0 (0.1)	54.2 (0.8)
CORAL	-	-	71.7 (0.1)	53.3 (0.0)
DANN	-	-	71.7 (0.1)	53.3 (0.0)
Pseudo-Label	90.3 (0.5)	66.9 (2.6)	71.6 (0.1)	52.3 (1.1)
Masked LM	89.4 (1.2)	65.7 (2.3)	71.9 (0.4)	53.9 (0.7)
ERM (fully-labelled)	89.9 (0.1)	69.4 (0.6)	73.6 (0.1)	56.4 (0.8)
ICON (Ours)	89.7 (0.1)	68.8 (1.3)	71.9 (0.1)	54.7 (0.0)

Table A2: Supplementary to Table 1. The in-distribution (ID) and out-of-distribution (OOD) performance of each method on each applicable dataset. The standard deviations of each method are reported in the brackets. We bold the highest non-fully-labeled OOD results.

E Additional Results

WILDS 2.0. In Table A2, we included the standard deviation on WILDS 2.0 benchmark, as well as additional results that are not included in the main paper due to space constraint. In particular, on datasets with data augmentations, we included results of empirical risk minimization (*i.e.*, train in S with cross-entropy loss) with/without data augmentations. We also included the results of an oracle (ERM fully-labeled) on IWILDCAM and FMoW (-wilds suffix excluded for convenience as in the main paper). On POVERTYMAP, we were not able to reproduce the results of ERM (with or without data augmentations) in the original paper, so we included the results that we obtained using the same code, command and environment.

References

- [1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.
- [2] Yogesh Balaji, Rama Chellappa, and Soheil Feizi. Normalized wasserstein for mixture distributions with applications in adversarial learning and domain adaptation. In *ICCV*, 2019.
- [3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Avital Oliver, Nicolas Papernot, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.
- [4] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NeurIPS*, 2016.
- [5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.
- [6] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *CVPR*, 2022.
- [7] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. In *AAAI*, 2020.
- [8] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *ICML*, 2016.
- [9] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, 2020.
- [10] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 2012.
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [12] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *CVPR*, 2021.
- [13] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *ICLR*, 2018.
- [14] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.
- [15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 2016.
- [16] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.
- [17] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.
- [18] Chen Gong, Tongliang Liu, Dacheng Tao, Keren Fu, Enmei Tu, and Jie Yang. Deformed graph laplacian for semisupervised learning. *IEEE transactions on neural networks and learning systems*, 2015.
- [19] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2004.
- [20] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *ICLR*, 2020.

- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [22] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.
- [23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.
- [24] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, 2019.
- [25] Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019.
- [26] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019.
- [27] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghui Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, 2021.
- [28] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013.
- [29] Shuang Li, Chi Liu, Qiuxia Lin, Binhui Xie, Zhengming Ding, Gao Huang, and Jian Tang. Domain conditioned adaptation network. In *AAAI*, 2020.
- [30] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.
- [31] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *CVPR*, 2021.
- [32] Jian Liang, Dapeng Hu, Jiashi Feng, and Ran He. Dine: Domain adaptation from single and multiple black-box predictors. In *CVPR*, 2022.
- [33] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, 2019.
- [34] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. In *NeurIPS*, 2021.
- [35] Xiaofeng Liu, Zhenhua Guo, Site Li, Fangxu Xing, Jane You, C-C Jay Kuo, Georges El Fakhri, and Jonghye Woo. Adversarial unsupervised domain adaptation with conditional and label shift: Infer, align and iterate. In *ICCV*, 2021.
- [36] Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018.
- [37] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, 2013.
- [38] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017.
- [39] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- [40] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *ECCV*, 2020.
- [41] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. 2013.

- [42] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *CVPR*, 2021.
- [43] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 2010.
- [44] Mohammad Peikari, Sherine Salama, Sharon Nofech-Mozes, and Anne L Martel. A cluster-then-label semi-supervised learning approach for pathology image classification. *Scientific reports*, 2018.
- [45] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.
- [46] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *ICCV*, 2021.
- [47] Harsh Rangwani, Sumukh K Aithal, Mayank Mishra, Arihant Jain, and Venkatesh Babu Radhakrishnan. A closer look at smoothness in domain adversarial training. In *ICML*, 2022.
- [48] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.
- [49] Shiori Sagawa, Pang Wei Koh, Tony Lee, Irena Gao, Sang Michael Xie, Kendrick Shen, Ananya Kumar, Weihua Hu, Michihiro Yasunaga, Henrik Marklund, et al. Extending the wilds benchmark for unsupervised adaptation. In *ICLR*, 2022.
- [50] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
- [51] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 2021.
- [52] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.
- [53] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *NeurIPS*, 2016.
- [54] Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In *ECCV*, 2020.
- [55] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.
- [56] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.
- [57] Tao Sun, Cheng Lu, and Haibin Ling. Prior knowledge guided unsupervised domain adaptation. In *ECCV*, 2022.
- [58] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, 2020.
- [59] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.
- [60] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
- [61] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 2020.

- [62] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.
- [63] Tan Wad, Qianru Sun, Sugiri Pranata, Karlekar Jayashree, and Hanwang Zhang. Equivariance and invariance inductive bias for learning from insufficient data. In *ECCV*, 2022.
- [64] Tan Wang, Zhongqi Yue, Jianqiang Huang, Qianru Sun, and Hanwang Zhang. Self-supervised learning disentangled group representation as feature. In *NeurIPS*, 2021.
- [65] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In *CVPR*, 2021.
- [66] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, Zhizheng Zhang, and Zhibo Chen. Toalign: Task-oriented alignment for unsupervised domain adaptation. In *NeurIPS*, 2021.
- [67] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.
- [68] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2018.
- [69] Zhongqi Yue, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Transporting causal mechanisms for unsupervised domain adaptation. In *ICCV*, 2021.
- [70] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.
- [71] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, 2019.
- [72] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, 2019.
- [73] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *ICML*, 2019.
- [74] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019.