Appendix

A Experimental Details

A.1 Description of Baselines

Average Thresholded Confidence (ATC). ATC first estimates a threshold t on the confidence of softmax prediction (or on negative entropy) such that the number of source labeled points that get a confidence greater than t match the fraction of correct examples, and then estimates the test error on on the target domain $\mathcal{D}_{\text{test}}$ as the expected number of target points that obtain a score less than t, i.e.,

$$\text{ATC}_{\mathcal{D}_{\text{test}}}(s) = \sum_{i=1}^{n} \mathbb{I}\left[s(f(x_i')) < t\right] ,$$

where t satisfies: $\sum_{i=1}^{j} \mathbb{I}\left[\max_{j\in\mathcal{Y}}(f_j(x_i)) < t\right] = \sum_{i=1}^{m} \mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x_i) \neq y_i\right]$

Average Confidence (AC). Error is estimated as the average value of the maximum softmax confidence on the target data, i.e, $\text{AC}_{\mathcal{D}_{\text{test}}} = \sum_{i=1}^{n} \max_{j\in\mathcal{Y}} f_j(x_i')$.

Difference Of Confidence (DOC). We estimate error on the target by subtracting the difference of confidences on source and target (as a surrogate to distributional distance [24]) from the error on source distribution, i.e., $\text{DOC}_{\mathcal{D}_{\text{test}}} = \sum_{i=1}^{n} \max_{j\in\mathcal{Y}} f_j(x_i') + \sum_{i=1}^{m} \mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x_i) \neq y_i\right] - \sum_{i=1}^{m} \max_{j\in\mathcal{Y}} f_j(x_i)$. This is referred to as DOC-Feat in [24].

Confidence Optimal Transport (COT). COT uses the empirical estimator of the Earth Mover's Distance between labels from the source domain and softmax outputs of samples from the target domain to provide accuracy estimates:

$$\text{COT}_{\mathcal{D}_{\text{test}}}(s) = \frac{1}{2} \min_{\pi\in\Pi(S^n, Y^m)} \sum_{i,j=1}^{n,m} \left\| s_i - e_{y_j} \right\|_2 \pi_{ij} ,$$

where $S^n = \{f(x_i')\}_{i=1}^n$ are the softmax outputs on the unlabeled target data and $Y^m = \{y_j\}_{j=1}^m$ are the labels on holdout source examples.

For all of the methods described above, we assume that $\{(x_i')\}_{i=1}^n$ are the unlabeled target samples and $\{(x_i, y_i)\}_{i=1}^m$ are hold-out labeled source samples.

A.2 Dataset Details

In this section, we provide additional details about the datasets used in our benchmark study.

- **CIFAR10** We use the original CIFAR10 dataset [36] as the source dataset. For target domains, we consider (i) synthetic shifts (CIFAR10-C) due to common corruptions [27]; and (ii) natural distribution shift, i.e., CIFAR10v2 [58, 68] due to differences in data collection strategy. We randomly sample 3 set of CIFAR-10-C datasets. Overall, we obtain 5 datasets (i.e., CIFAR10v1, CIFAR10v2, CIFAR10C-Frost (severity 4), CIFAR10C-Pixelate (severity 5), CIFAR10-C Saturate (severity 5)).
- **CIFAR100** Similar to CIFAR10, we use the original CIFAR100 set as the source dataset. For target domains we consider synthetic shifts (CIFAR100-C) due to common corruptions. We sample 4 CIFAR100-C datasets, overall obtaining 5 domains (i.e., CIFAR100, CIFAR100C-Fog (severity 4), CIFAR100C-Motion Blur (severity 2), CIFAR100C-Contrast (severity 4), CIFAR100C-spatter (severity 2)).
- **FMoW** In order to consider distribution shifts faced in the wild, we consider FMoW-WILDs [35, 11] from WILDS benchmark, which contains satellite images taken in different geographical regions and at different times. We use the original train as source and OOD val and OOD test splits as target domains as they are collected over different time-period. Overall, we obtain 3 different domains.
- **Camelyon17** Similar to FMoW, we consider tumor identification dataset from the wilds benchmark [4]. We use the default train as source and OOD val and OOD test splits as target domains as they are collected across different hospitals. Overall, we obtain 3 different domains.

- **BREEDs** We also consider BREEDs benchmark [65] in our setup to assess robustness to subpopulation shifts. BREEDs leverage class hierarchy in ImageNet to re-purpose original classes to be the subpopulations and defines a classification task on superclasses. We consider distribution shift due to subpopulation shift which is induced by directly making the subpopulations present in the training and test distributions disjoint. BREEDs benchmark contains 4 datasets **Entity-13**, **Entity-30**, **Living-17**, and **Non-living-26**, each focusing on different subtrees and levels in the hierarchy. We also consider natural shifts due to differences in the data collection process of ImageNet [63], e.g, ImageNetv2 [60] and a combination of both. Overall, for each of the 4 BREEDs datasets (i.e., Entity-13, Entity-30, Living-17, and Non-living-26), we obtain four different domains. We refer to them as follows: BREEDsv1 sub-population 1 (sampled from ImageNetv1), BREEDsv1 sub-population 2 (sampled from ImageNetv1), BREEDsv2 sub-population 1 (sampled from ImageNetv2), BREEDsv2 sub-population 2 (sampled from ImageNetv2). For each BREEDs dataset, we use BREEDsv1 sub-population A as source and the other three as target domains.
- **OfficeHome** We use four domains (art, clipart, product and real) from OfficeHome dataset [69]. We use the product domain as source and the other domains as target.
- **DomainNet** We use four domains (clipart, painting, real, sketch) from the Domainnet dataset [53]. We use real domain as the source and the other domains as target.
- **Visda** We use three domains (train, val and test) from the Visda dataset [52]. While 'train' domain contains synthetic renditions of the objects, 'val' and 'test' domains contain real world images. To avoid confusing, the domain names with their roles as splits, we rename them as 'synthetic', 'Real-1' and 'Real-2'. We use the synthetic (original train set) as the source domain and use the other domains as target.

A.3 Setup and Protocols

Architecture Details For all datasets, we used the same architecture across different algorithms:

- CIFAR-10: Resnet-18 [26] pretrained on Imagenet
- CIFAR-100: Resnet-18 [26] pretrained on Imagenet
- Camelyon: Densenet-121 [28] *not* pretrained on Imagenet as per the suggestion made in [35]
- FMoW: Densenet-121 [28] pretrained on Imagenet
- BREEDs (Entity13, Entity30, Living17, Nonliving26): Resnet-18 [26] *not* pretrained on Imagenet as per the suggestion in [65]. The main rationale is to avoid pre-training on the superset dataset where we are simulating sub-population shift.
- Officehome: Resnet-50 [26] pretrained on Imagenet
- Domainnet: Resnet-50 [26] pretrained on Imagenet
- Visda: Resnet-50 [26] pretrained on Imagenet

Except for Resnets on CIFAR datasets, we used the standard pytorch implementation [19]. For Resnet on cifar, we refer to the implementation here: `https://github.com/kuangliu/pytorch-cifar`. For all the architectures, whenever applicable, we add antialiasing [71]. We use the official library released with the paper.

For imagenet-pretrained models with standard architectures, we use the publicly available models here: `https://pytorch.org/vision/stable/models.html`. For imagenet-pretrained models on the reduced input size images (e.g. CIFAR-10), we train a model on Imagenet on reduced input size from scratch. We include the model with our publicly available repository.

Hyperparameter details First, we tune learning rate and ℓ_2 regularization parameter by fixing batch size for each dataset that correspond to maximum we can fit to 15GB GPU memory. We set the number of epochs for training as per the suggestions of the authors of respective benchmarks. Note that we define the number of epochs as a full pass over the labeled training source data. We summarize learning rate, batch size, number of epochs, and ℓ_2 regularization parameter used in our study in Table A.3.

For each algorithm, we use the hyperparameters reported in the initial papers. For domain-adversarial methods (DANN and CDANN), we refer to the suggestions made in Transfer Learning Library [31]. We tabulate hyperparameters for each algorithm next:

Dataset	Source	Target
CIFAR10	CIFAR10v1	CIFAR10v1, CIFAR10v2, CIFAR10C-Frost (severity 4), CIFAR10C-Pixelate (severity 5), CIFAR10-C Saturate (severity 5)
CIFAR100	CIFAR100	CIFAR100, CIFAR100C-Fog (severity 4), CIFAR100C-Motion Blur (severity 2), CIFAR100C-Contrast (severity 4), CIFAR100C-spatter (severity 2)
Camelyon	Camelyon (Hospital 1–3)	Camelyon (Hospital 1–3), Camelyon (Hospital 4), Camelyon (Hospital 5)
FMoW	FMoW (2002–'13)	FMoW (2002–'13), FMoW (2013–'16), FMoW (2016–'18)
Entity13	Entity13 (ImageNetv1 sub-population 1)	Entity13 (ImageNetv1 sub-population 1), Entity13 (ImageNetv1 sub-population 2), Entity13 (ImageNetv2 sub-population 1), Entity13 (ImageNetv2 sub-population 2)
Entity30	Entity30 (ImageNetv1 sub-population 1)	Entity30 (ImageNetv1 sub-population 1), Entity30 (ImageNetv1 sub-population 2), Entity30 (ImageNetv2 sub-population 1), Entity30 (ImageNetv2 sub-population 2)
Living17	Living17 (ImageNetv1 sub-population 1)	Living17 (ImageNetv1 sub-population 1), Living17 (ImageNetv1 sub-population 2), Living17 (ImageNetv2 sub-population 1), Living17 (ImageNetv2 sub-population 2)
Nonliving26	Nonliving26 (ImageNetv1 sub-population 1)	Nonliving26 (ImageNetv1 sub-population 1), Nonliving26 (ImageNetv1 sub-population 2), Nonliving26 (ImageNetv2 sub-population 1), Nonliving26 (ImageNetv2 sub-population 2)
Officehome	Product	Product, Art, ClipArt, Real
DomainNet	Real	Real, Painiting, Sketch, ClipArt
Visda	Synthetic (originally referred to as train)	Synthetic, Real-1 (originally referred to as val), Real-2 (originally referred to as test)

Table A.2: Details of the source and target datasets in our testbed.

Dataset	Epoch	Batch size	ℓ_2 regularization	Learning rate
CIFAR10	50	200	0.0001 (chosen from $\{0.0001, 0.001$,1e-5, $0.0\}$)	0.01 (chosen from $\{0.001, 0.01, 0.0001\}$)
CIFAR100	50	200	0.0001 (chosen from $\{0.0001, 0.001$,1e-5, $0.0\}$)	0.01 (chosen from $\{0.001, 0.01, 0.0001\}$)
Camelyon	10	96	0.01 (chosen from $\{0.01, 0.001, 0.0001, 0.0\}$)	0.03 (chosen from $\{0.003, 0.3, 0.0003, 0.03\}$)
FMoW	30	64	0.0 (chosen from $\{0.0001, 0.001$,1e-5,$0.0\}$)	0.0001 (chosen from $\{0.001, 0.01, 0.0001\}$)
Entity13	40	256	5e-5 (chosen from {5e-5, 5e-4, 1e-4, 1e-5})	0.2 (chosen from $\{0.1, 0.5, 0.2, 0.01, 0.0\}$)
Entity30	40	256	5e-5 (chosen from {5e-5, 5e-4, 1e-4, 1e-5})	0.2 (chosen from $\{0.1, 0.5, 0.2, 0.01, 0.0\}$)
Living17	40	256	5e-5 (chosen from {5e-5, 5e-4, 1e-4, 1e-5})	0.2 (chosen from $\{0.1, 0.5, 0.2, 0.01, 0.0\}$)
Nonliving26	40	256	0 5e-5 (chosen from {5e-5, 5e-4, 1e-4, 1e-5})	0.2 (chosen from $\{0.1, 0.5, 0.2, 0.01, 0.0\}$)
Officehome	50	96	0.0001 (chosen from $\{0.0001, 0.001$,1e-5, $0.0\}$)	0.01 (chosen from $\{0.001, 0.01, 0.0001\}$)
DomainNet	15	96	0.0001 (chosen from $\{0.0001, 0.001$,1e-5, $0.0\}$)	0.01 (chosen from $\{0.001, 0.01, 0.0001\}$)
Visda	10	96	0.0001 (chosen from $\{0.0001, 0.001$,1e-5, $0.0\}$)	0.01 (chosen from $\{0.001, 0.01, 0.0001\}$)

Table A.3: Details of the learning rate and batch size considered in our testbed

- **DANN, CDANN,** As per Transfer Learning Library suggestion, we use a learning rate multiplier of 0.1 for the featurizer when initializing with a pre-trained network and 1.0 otherwise. We default to a penalty weight of 1.0 for all datasets with pre-trained initialization.

- **FixMatch** We use the lambda is 1.0 and use threshold τ as 0.9.

Compute Infrastructure Our experiments were performed across a combination of Nvidia T4, A6000, and V100 GPUs.

B Comparing Disagreement Losses

We define the alternate losses for maximizing disagreement:

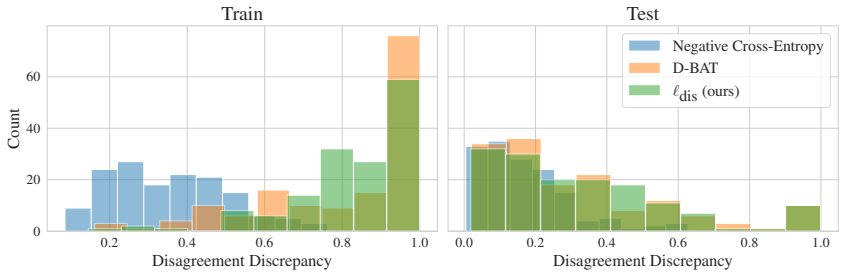
1. Chuang et al. [12] minimize the negative cross-entropy loss, which is concave in the model logits. That is, they add the term $\log \text{softmax}(h(x)_y)$ to the objective they are minimizing. This loss results in substantially lower disagreement discrepancy than the other two.
2. Pagliardini et al. [50] use a loss which is not too different from ours. They define the disagreement objective for a point (x, y) as

$$\log\left(1 + \frac{\exp(h(x)_y)}{\sum_{\hat{y}\neq y}\exp(h(x)_{\hat{y}})}\right). \tag{1}$$

For comparison, ℓ_{dis} can be rewritten as

$$\log\left(1 + \frac{\exp(h(x)_y)}{\exp\left(\frac{1}{|\mathcal{Y}|-1}\sum_{\hat{y}\neq y} h(x)_{\hat{y}}\right)}\right), \tag{2}$$

where the incorrect logits are averaged and the exponential is pushed outside the sum. This modification results in (2) being convex in the logits and an upper bound to the disagreement 0-1 loss, whereas (1) is neither.



Loss	Mean Discrepancy (Train)	Mean Discrepancy (Test)
Neg. X-Ent [12]	$0.3555 \pm .0124$	$0.1694 \pm .0105$
D-BAT [50]	$0.8145 \pm .0177$	$0.3224 \pm .0212$
ℓ_{dis} (Ours)	$\underline{0.8333 \pm .0132}$	$\mathbf{0.3322 \pm .0205}$

Figure B.1 & Table B.3: Histogram of disagreement discrepancies for each of the three losses, and the average values across all datasets. **Bold** (resp. Underline) indicates the method has higher average discrepancy under a paired t-test at significance $p = .01$ (resp. $p = .05$).

Figure B.1 displays histograms of the achieved disagreement discrepancy across all distributions for each of the disagreement losses (all hyperparameters and random seeds are the same for all three losses). The table below it reports the mean disagreement discrepancy on the train and test sets. We find that the negative cross-entropy, being a concave function, results in very low discrepancy. The loss (1) is reasonably competitive with our loss (2) on average, seemingly because it gets very high discrepancy on a subset of shifts. This suggests that it may be particularly suited for a specific type of distribution shift, though it is less good overall. Though the averages are reasonably close, the samples are not independent, so we run a paired t-test and we find that the increases to average train and test discrepancies achieved by ℓ_{dis} are significant at levels $p = 0.024$ and $p = 0.009$, respectively. However, with enough holdout data, a reasonable approach would be to split the data in two: one subset to validate critics trained on either of the two losses, and another to evaluate the discrepancy of whichever one is ultimately selected.

C Exploration of the Validity Score

To experiment with reducing the complexity of the class $\mathcal{H}$, we evaluate DIS^2 on progressively fewer top principal components (PCs) of the features. Precisely, for features of dimension d, we evaluate DIS^2 on the same features projected onto their top d/k components, for $k \in [1, 4, 16, 32, 64, 128]$ (Figure C.2). We see that while projecting to fewer and fewer PCs does reduce the error bound value, unlike the logits it is a rather crude way to reduce complexity of $\mathcal{H}$, meaning at some point it goes too far and results in invalid error bounds.

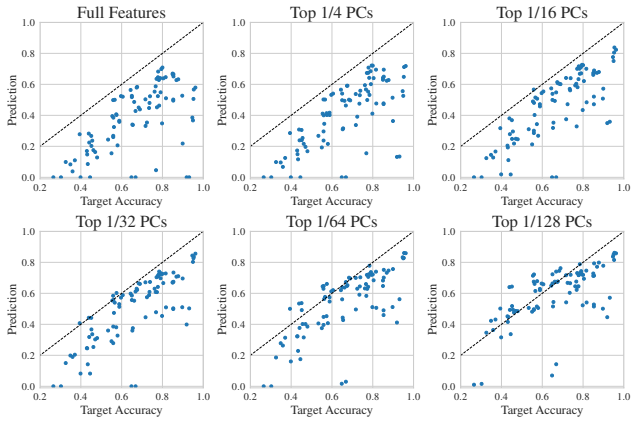


Figure C.2: **DIS^2 bound as fewer principal components are kept.** Reducing the number of top principal components crudely reduces complexity of $\mathcal{H}$—this leads to lower error estimates, but at some point the bounds become invalid for a large fraction of shifts.

However, during the optimization process we observe that around when this violation occurs, the task of training a critic to both agree on $\mathcal{S}$ and disagree on $\mathcal{T}$ goes from "easy" to "hard". Figure C.3 shows that on the full features, the critic rapidly ascends to maximum agreement on $\mathcal{S}$, followed by slow decay (due to both overfitting and learning to simultaneously disagree on $\mathcal{T}$). As we drop more and more components, this optimization becomes slower.

Figure C.3: **Agreement on one shift between $\hat{h}$ and h' on $\hat{\mathcal{S}}$ during optimization.** We observe that as the number of top PCs retained drops, the optimization occurs more slowly and less monotonically. For this particular shift, the bound becomes invalid when keeping only the top $1/128$ components, depicted by the brown line.

We therefore design a "validity score" intended to capture this phenomenon which we refer to as the *cumulative ℓ_1 ratio*. This is defined as the maximum agreement achieved, divided by the cumulative sum of absolute differences in agreement across all epochs up until the maximum was achieved.

Formally, let $\{a_i\}_{i=1}^T$ represent the agreement between h' and $\hat{h}$ after epoch i, i.e. $1 - \epsilon_{\hat{\mathcal{S}}}(\hat{h}, h'_i)$, and define $m := \arg\max_{i \in [T]} a_i$. The cumulative ℓ_1 ratio is then $\frac{a_m}{a_1 + \sum_{i=2}^{m} |a_i - a_{i-1}|}$. Thus, if the agreement rapidly ascends to its maximum without ever going down over the course of an epoch, this ratio will be equal to 1, and if it non-monotonically ascends then the ratio will be significantly less. This definition was simply the first metric we considered which approximately captures the behavior we observed; we expect it could be greatly improved.

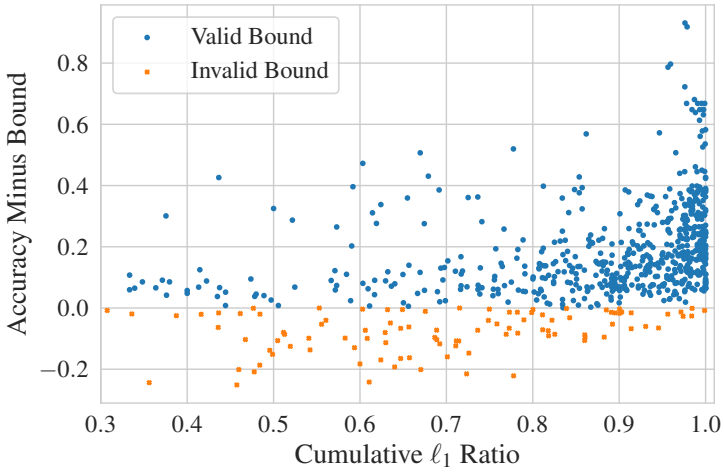


Figure C.4: **Cumulative ℓ_1 ratio versus error prediction gap.** Despite its simplicity, the ratio captures the information encoded in the optimization trajectory, roughly linearly correlating with the tightness and validity of a given prediction. It is thus a useful metric for identifying the ideal number of top PCs to use.

Figure C.4 displays a scatter plot of the cumulative ℓ_1 ratio versus the difference in estimated and true error for DIS^2 evaluated on the full range of top PCs. A negative value implies that we have underestimated the error (i.e., the bound is not valid). We see that even this very simply metric roughly linearly correlates with the tightness of the bound, which suggests that evaluating over a range of top PC counts and only keeping predictions whose ℓ_1 ratio is above a certain threshold can improve raw predictive accuracy without reducing coverage by too much. Figure C.5 shows that this is indeed the case: compared to DIS^2 evaluated on the logits, keeping all predictions above a score threshold can produce more accurate error estimates, without *too* severely underestimating error in the worst case.

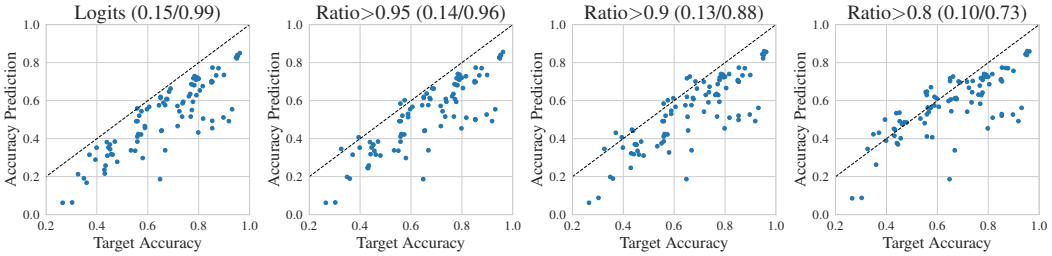


Figure C.5: **DIS^2 bounds and MAE / coverage as the cumulative ℓ_1 ratio threshold is lowered.** Values in parenthesis are (MAE / coverage). By only keeping predictions with ratio above a varying threshold, we can smoothly interpolate between bound validity and raw error prediction accuracy.

D Making Baselines More Conservative with LOOCV

To more thoroughly compare DIS^2 to prior estimation techniques, we consider a strengthening of the baselines which may give them higher coverage without too much cost to prediction accuracy. Specifically, for each desired coverage level $\alpha \in [0.9, 0.95, 0.99]$, we use all but one of the datasets to learn a parameter to either scale or shift a method's predictions enough to achieve coverage α. We then evaluate this scaled or shifted prediction on the distribution shifts of the remaining dataset, and we repeat this for each one.

The results, found in Table D.4, demonstrate that prior methods can indeed be made to have much higher coverage, although as expected their MAE suffers. Furthermore, they still underestimate error on the tail distribution shifts by quite a bit, and they rarely achieve the desired coverage on the heldout dataset—though they usually come reasonably close. In particular, ATC [21] and COT [40] do well with a shift parameter, e.g. at the desired coverage $\alpha = 0.95$ ATC matches DIS^2 in MAE and gets 94.4% coverage (compared to 98.9% by DIS^2). However, its conditional average overestimation is quite high, almost 9%. COT gets much lower overestimation (particularly for higher coverage levels), and it also appears to suffer less on the tail distribution shifts in the sense that $\alpha = 0.99$ does not induce nearly as high MAE as it does for ATC. However, at that level it only achieves 95.6% coverage, and it averages almost 5% accuracy overestimation on the shifts it does not correctly bound (compared to 0.1% by DIS^2). Also, its MAE is still substantially higher than DIS^2, despite getting lower coverage. Finally, we evaluate the scale/shift approach on our DIS^2 bound without the lower order term, but based on the metrics we report there appears to be little reason to prefer it over the untransformed version, one of the baselines, or the original DIS^2 bound.

Taken together, these results imply that if one's goal is predictive accuracy and tail behavior is not important (worst ~10%), ATC or COT will likely get reasonable coverage with a shift parameter—though they still significantly underestimate error on a non-negligible fraction of shifts. If one cares about the long tail of distribution shifts, or prioritizes being conservative at a slight cost to average accuracy, DIS^2 is clearly preferable. Finally, we observe that the randomness which determines which shifts are not correctly bounded by DIS^2 is "decoupled" from the distributions themselves under Theorem 3.6, in the sense that it is an artifact of the random samples, rather than a property of the distribution (recall Figure 4(b)). This is in contrast with the shift/scale approach which would produce almost identical results under larger sample sizes because it does not account for finite sample effects. This implies that some distribution shifts are simply "unsuitable" for prior methods because they do not satisfy whatever condition these methods rely on, and observing more samples will not remedy this problem. It is clear that working to understand these conditions is crucial for reliability and interpretability, since we are not currently able to identify which distributions are suitable a priori.

		MAE (↓)			Coverage (↑)			Overest. (↓)		
	$\alpha \rightarrow$	0.9	0.95	0.99	0.9	0.95	0.99	0.9	0.95	0.99
Method	Adjustment									
AC	none		0.106			0.122			0.118	
	shift	0.153	0.201	0.465	0.878	0.922	0.956	0.119	0.138	0.149
	scale	0.195	0.221	0.416	0.911	0.922	0.967	0.135	0.097	0.145
DoC	none		0.105			0.167			0.122	
	shift	0.158	0.200	0.467	0.878	0.911	0.956	0.116	0.125	0.154
	scale	0.195	0.223	0.417	0.900	0.944	0.967	0.123	0.139	0.139
ATC NE	none		0.067			0.289			0.083	
	shift	0.117	0.150	0.309	0.900	0.944	0.978	0.072	0.088	0.127
	scale	0.128	0.153	0.357	0.889	0.933	0.978	0.062	0.074	0.144
COT	none		0.069			0.256			0.085	
	shift	0.115	0.140	0.232	0.878	0.944	0.956	0.049	0.065	0.048
	scale	0.150	0.193	0.248	0.889	0.944	0.956	0.074	0.066	0.044
DIS^2 (w/o δ)	none		0.083			0.756			0.072	
	shift	0.159	0.169	0.197	0.889	0.933	0.989	0.021	0.010	0.017
	scale	0.149	0.168	0.197	0.889	0.933	0.989	0.023	0.021	0.004
DIS^2 ($\delta = 10^{-2}$)	none		0.150			0.989			0.001	
DIS^2 ($\delta = 10^{-3}$)	none		0.174			1.000			0.000	

Table D.4: MAE, coverage, and conditional average overestimation for the strengthened baselines with a shift or scale parameter on non-domain-adversarial representations. Because a desired coverage α is only used when an adjustment is learned, "none"—representing no adjustment—does not vary with α.

E Proving that Assumption 3.5 Holds for Some Datasets

Here we describe how the equivalence of Assumption 3.5 and the bound in Theorem 3.6 allow us to prove that the assumption holds with high probability. By repeating essentially the same proof as Theorem 3.6 in the other direction, we get the following corollary:

Corollary E.1. *If Assumption 3.5 does* not *hold, then with probability* $\geq 1 - \delta$*,*

$$\epsilon_{\hat{\mathcal{T}}}(\hat{h}) > \epsilon_{\hat{\mathcal{S}}}(\hat{h}) + \hat{\Delta}(\hat{h}, h') - \sqrt{\frac{2(n_S + n_T)\log 1/\delta}{n_S n_T}}.$$

Note that the last term here is different from Theorem 3.6 because we are bounding the empirical target error, rather than the true target error. The reason for this change is that now we can make direct use of its contrapositive:

Corollary E.2. *If it is the case that*

$$\epsilon_{\hat{\mathcal{T}}}(\hat{h}) \leq \epsilon_{\hat{\mathcal{S}}}(\hat{h}) + \hat{\Delta}(\hat{h}, h') - \sqrt{\frac{2(n_S + n_T)\log 1/\delta}{n_S n_T}},$$

then either Assumption 3.5 holds, or an event has occurred which had probability $\leq \delta$ *over the randomness of the samples* $\hat{\mathcal{S}}, \hat{\mathcal{T}}$*.*

We evaluate this bound on non-domain-adversarial shifts with $\delta = 10^{-6}$. As some of the BREEDS shifts have as few as 68 test samples, we restrict ourselves to shifts with $n_T \geq 500$ to ignore those where the finite-sample term heavily dominates; this removes a little over 20% of all shifts. Among the remainder, we find that the bound in Corollary E.2 holds 55.7% of the time when using full features and 25.7% of the time when using logits. This means that for these shifts, we can be essentially certain that Assumption 3.5—and therefore also Assumption 3.3—is true.

Note that the fact that the bound is *not* violated for a given shift does not at all imply that the assumption is not true. In general, the only rigorous way to prove that Assumption 3.5 does not hold would be to show that for a fixed δ, the fraction of shifts for which the bound in Theorem 3.6 does not hold is larger than δ (in a manner that is statistically significant under the appropriate hypothesis test). Because this never occurs in our experiments, we cannot conclude that the assumption is ever false. At the same time, the fact that the bound *does* hold at least $1 - \delta$ of the time does not prove that the assumption is true—it merely suggests that it is reasonable and that the bound should continue to hold in the future. This is why it is important for Assumption 3.5 to be simple and intuitive, so that we can trust that it will persist and anticipate when it will not.

However, Corollary E.2 allows us to make a substantially stronger statement. In fact, it says that for *any* distribution shift, with enough samples, we can prove a posteriori whether or not Assumption 3.5 holds, because the gap between these two bounds will shrink with increasing sample size.

F Figure 1 Stratified by Training Method

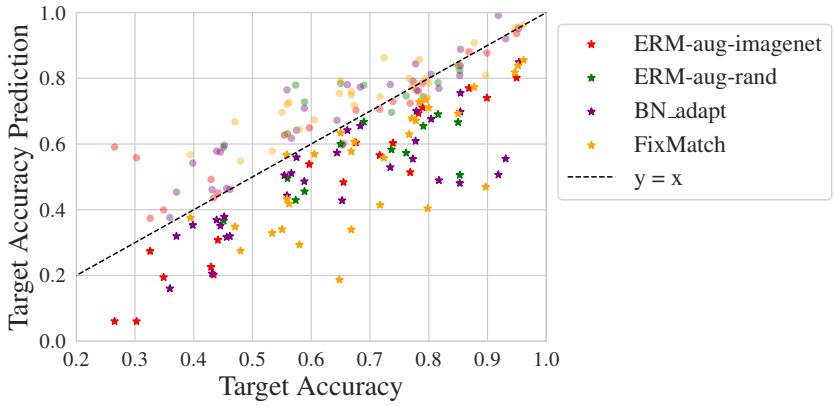


Figure F.6: **Error prediction stratified by training method.** Stars denote DIS^2, circles are ATC NE. We see that DIS^2 maintains its validity across different training methods.