(Almost) Provable Error Bounds Under Distribution Shift via Disagreement Discrepancy

Elan Rosenfeld [1] **Saurabh Garg** [1]

Abstract

We derive an (almost) guaranteed upper bound on the error of deep neural networks under distribution shift using unlabeled test data. Prior methods either give bounds that are vacuous in practice or give *estimates* that are accurate on average but heavily underestimate error for a sizeable fraction of shifts. Our bound requires a simple, intuitive condition which is well justified by prior empirical works and holds in practice effectively 100% of the time. The bound is inspired by $\mathcal{H}\Delta\mathcal{H}$-divergence but is easier to evaluate and substantially tighter, consistently providing non-vacuous guarantees. Estimating the bound requires optimizing one multiclass classifier to disagree with another, for which some prior works have used sub-optimal proxy losses; we devise a "disagreement loss" which is theoretically justified and performs better in practice. Across a wide range of benchmarks, our method gives valid error bounds while achieving average accuracy comparable to competitive estimation baselines.

1. Introduction

When deploying a model, it is important to be confident in its stability under inevitable distribution shift. Unfortunately, standard methods for this are typically vacuous in practice (Mansour et al., 2009; Ben-David et al., 2006) or assume a precise model for which it is impossible to determine the reasonableness of the assumptions (Rahimian and Mehrotra, 2019). To better estimate stability in the wild, some recent work instead tries to directly predict accuracy of neural networks using unlabeled data from the test distribution (Garg et al., 2022; Baek et al., 2022; Lu et al., 2023). While these methods are accurate, they lack pointwise trustworthiness:

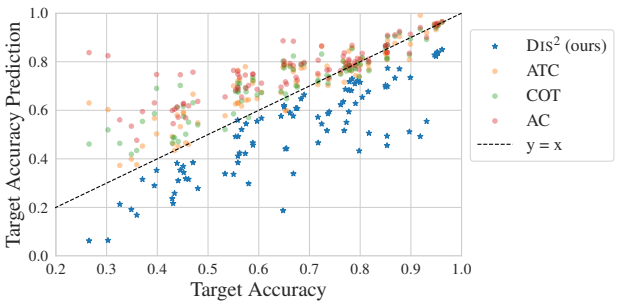


Figure 1. **Our bound vs. three prior methods for estimation across a wide variety of shift benchmarks and training methods.** Prior methods are accurate on average, but it is impossible to know if a given prediction is reliable. Worse, they usually overestimate accuracy, with the gap growing as test accuracy decreases—*this is precisely when a reliable, conservative estimate is most desirable.* Instead, DIS^2 maximizes the **dis**agreement **dis**crepancy to give a reliable error bound which holds effectively 100% of the time.

their estimates are good on average, but they provide no signal of the quality of any individual prediction (here, each point is a *distribution*, for which a method predicts a classifier's accuracy). Indeed, it is reasonably common for them to substantially overestimate test accuracy, which is problematic when optimistic deployment is costly. Worse yet, we find that this gap *grows with test error* (Figure 1), making these predictions least reliable precisely when their reliability is most important. **Though it is impossible to give test error upper bounds for all shifts,** there is still potential for bounds that are intuitive and reasonably trustworthy.

In this work, we develop a method for (almost) provably bounding test error of classifiers under distribution shift using unlabeled test points. Our bound's only requirement is a simple, intuitive, condition which describes the ability of a hypothesis class to achieve small loss on a objective defined over the (unlabeled) train and test distributions. Inspired by $\mathcal{H}\Delta\mathcal{H}$-divergence (Ben-David et al., 2010), our method trains a critic to maximize agreement with the classifier of interest on the source distribution while maximizing *dis*agreement on the target distribution; we refer to this joint objective as the *disagreement discrepancy*, and so we name the method DIS^2. We optimize this discrepancy over linear classifiers using deep features—or linear functions thereof—finetuned on the training set. Recent evidence suggests that

[1]Machine Learning Department, Carnegie Mellon University. Correspondence to: Elan Rosenfeld <elan@cmu.edu>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Honolulu, Hawaii, USA. Copyright 2023 by the author(s).

such representations are sufficient for expressive classifiers even under large distribution shift (Rosenfeld et al., 2022). Experimentally, we find that our bound is valid effectively 100% of the time,[1] consistently giving non-trivial lower bounds on test accuracy which are comparable to competitive baselines. We also show that it is possible to test this bound's likelihood of being satisfied, and we use this to construct a score which can relax the original bound into successively tighter-yet-less-conservative estimates.

While maximizing agreement is statistically well understood, our method also calls for maximizing *dis*agreement on the target distribution. We observe that prior works use losses which do not correspond to minimizing the 0-1 loss of interest and are non-convex (or even *concave*) in the model logits (Chuang et al., 2020; Pagliardini et al., 2023). To rectify this, we derive a new "disagreement loss" which serves as an effective proxy for maximizing multiclass disagreement. Experimentally, we find that minimizing this loss results in higher disagreement.

Experiments across numerous vision datasets demonstrate the effectiveness of our bound. Though DIS^2 is competitive with prior methods for error estimation, **we emphasize that our focus is *not* on improving raw predictive accuracy**—rather, we hope to obtain reliable (i.e., conservative), reasonably tight bounds on the test error of a given classifier under distribution shift. See Appendix F for related work.

2. Deriving an (Almost) Provable Error Bound

Notation. Let $\mathcal{S}, \mathcal{T}$ denote the source and target (train and test) distributions, respectively, over labeled inputs $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and let $\hat{\mathcal{S}}, \hat{\mathcal{T}}$ denote sets of samples from them with cardinalities n_S and n_T (they also denote the corresponding empirical distributions). Recall that we observe only the covariates x without the label y when a sample is drawn from $\mathcal{T}$. We consider classifiers $h : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$ which output a vector of logits, and we let $\hat{h}$ denote the particular classifier whose error we aim to bound. Generally, we use $\mathcal{H}$ to denote a hypothesis class of such classifiers. Occasionally, where clear from context, we use $h(x)$ to refer to the argmax logit, i.e. the predicted class. We treat these classifiers as deterministic throughout, though our analysis can easily be extended to probabilistic classifiers and labels. For a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, let $\epsilon_{\mathcal{D}}(h, h') := \mathbb{E}_{\mathcal{D}}[\mathbf{1}\{\arg\max_y h(x)_y \neq \arg\max_y h'(x)_y\}]$ denote the one-hot disagreement between classifiers h and h' on $\mathcal{D}$. Let y^* represent the true labeling function such that $y^*(x) = y$ for all samples (x, y); with some abuse of notation, we write $\epsilon_{\mathcal{D}}(h)$ to mean $\epsilon_{\mathcal{D}}(h, y^*)$, i.e. the 0-1 error of classifier h on distribution $\mathcal{D}$.

[1]The few violations are expected a priori, have an obvious explanation, and only occur for a specific type of learned representation. We defer a more detailed discussion of this until after we present the bound.

The bound we derive in this work is extremely simple and relies on one new concept:

Definition 2.1. The *disagreement discrepancy* $\Delta(h, h')$ is the disagreement between h and h' on $\mathcal{T}$ minus their disagreement on $\mathcal{S}$: $\Delta(h, h') := \epsilon_{\mathcal{T}}(h, h') - \epsilon_{\mathcal{S}}(h, h')$.

We leave the dependence on $\mathcal{S}, \mathcal{T}$ implicit. Note that this term is symmetric and signed—it can be negative. With this definition, we now have the following lemma:

Lemma 2.2. *For any h, $\epsilon_{\mathcal{T}}(h) = \epsilon_{\mathcal{S}}(h) + \Delta(h, y^*)$.*

We cannot directly use Lemma 2.2 to estimate $\epsilon_{\mathcal{T}}(\hat{h})$ because the second term is unknown. However, observe that y^* is *fixed*. That is, while a learned $\hat{h}$ will depend on y^*—and therefore $\Delta(\hat{h}, y^*)$ may be large under large distribution shift—y^* **is *not* chosen to maximize $\Delta(\hat{h}, y^*)$ in response to the $\hat{h}$ we have learned.** This means that for an expressive hypothesis class $\mathcal{H}$, it should be possible to identify an alternative function $h' \in \mathcal{H}$ for which $\Delta(\hat{h}, h') \geq \Delta(\hat{h}, y^*)$ (we refer to such h' as the *critic*). In other words, we should be able to find an $h' \in \mathcal{H}$ which, *if it were the true labeling function*, would imply at least as large of a drop in accuracy from train to test as occurs in reality.

In this work we consider the class $\mathcal{H}$ of linear critics, with $\mathcal{X}$ defined as source-finetuned deep neural representations or the resulting logits output by $\hat{h}$. Prior work provides strong evidence that this class has surprising capacity under distribution shift, including the possibility that functions very similar to y^* lie in $\mathcal{H}$ (Kang et al., 2020; Rosenfeld et al., 2022; Kirichenko et al., 2022). We formalize this intuition with the following assumption:

Assumption 2.3. *Define $h^* := \arg\max_{h' \in \mathcal{H}} \Delta(\hat{h}, h')$. We assume $\Delta(\hat{h}, y^*) \leq \Delta(\hat{h}, h^*)$.*

Note that this statement is guaranteed for $y^* \in \mathcal{H}$; it becomes meaningful when considering restricted $\mathcal{H}$, as we do here. Note also that this assumption is made on a per-classifier basis. This is important because while the above may not hold for every classifier $\hat{h}$, it need only hold for the classifiers whose error we would hope to bound, which is in practice a very small subset of all classifiers. From Lemma 2.2, we immediately have the following result:

Proposition 2.4. *Under Assumption 2.3, $\epsilon_{\mathcal{T}}(\hat{h}) \leq \epsilon_{\mathcal{S}}(\hat{h}) + \Delta(\hat{h}, h^*)$.*

Unfortunately, identifying the optimal h^* is intractable, so this bound is still not estimable—we present it as an intermediate result for clarity. To derive the practical bound, we need one more step. In Section 3, we derive a "disagreement loss" which we use to maximize the empirical disagreement discrepancy. Relying on this loss, we instead assume:

Assumption 2.5. *Suppose we identify the critic $h' \in \mathcal{H}$ which maximizes a concave surrogate to the empirical disagreement discrepancy. We assume $\Delta(\hat{h}, y^*) \leq \Delta(\hat{h}, h')$.*

This is slightly stronger than Assumption 2.3—the difference in strength between these two assumptions shrinks as the number of available samples grows and as the quality of our surrogate objective improves. Ultimately, our bound holds without these terms, implying that the stronger assumption is reasonable. We can now give our main result:

Theorem 2.6 (Main Bound)**.** *Under Assumption 2.5, with probability $\geq 1 - \delta$, $\epsilon_{\mathcal{T}}(\hat{h}) \leq \epsilon_{\hat{\mathcal{S}}}(\hat{h}) + \hat{\Delta}(\hat{h}, h') + \sqrt{\frac{(n_S + 4n_T)\log 1/\delta}{2n_S n_T}}$.*

The proof is in Appendix H. The core message behind Theorem 2.6 is that if there is a simple (i.e., linear) critic h' with large discrepancy, the true y^* could plausibly be this function, implying $\hat{h}$ could have high error—likewise, if no simple y^* could hypothetically result in high error, we should expect low error.

Remark 2.7. Bounding error under distribution shift is impossible without assumptions. Prior works which estimate accuracy with unlabeled data rely on experiments, suggesting that whatever condition allows their method to work holds in a variety of settings (Garg et al., 2022; Baek et al., 2022; Lu et al., 2023; Guillory et al., 2021); using these methods is *implicitly* assuming that it will hold for future shifts. Understanding these conditions is thus crucial for assessing whether they can be expected to be satisfied. It is therefore of great practical value that Assumption 2.5 is simple and intuitive: below we demonstrate that this simplicity allows us to identify potential failure cases *a priori*.

DIS^2 vs. $\mathcal{H}\Delta\mathcal{H}$-Divergence One early attempt at bounding error under shift was *$\mathcal{H}$-divergence* (Ben-David et al., 2006; Mansour et al., 2009) which measures the ability of a binary hypothesis class to discriminate between $\mathcal{S}$ and $\mathcal{T}$. This was later refined to *$\mathcal{H}\Delta\mathcal{H}$-divergence* (Ben-David et al., 2010), which is equal to $\mathcal{H}$-divergence where the discriminator class comprises all xors between pairs from the original class. Though this measure can in principle provide non-vacuous bounds, it usually does not, and evaluating it is intractable. Furthermore, these bounds are too conservative even for simple functions and distribution shifts because they use uniform convergence. In practice, *we do not care* about bounding the error of all classifiers in $\mathcal{H}$—we only care to bound the error of $\hat{h}$. More importantly, one should not expect the distribution shift to be *truly* worst case, because the $\mathcal{T}$ and y^* are not chosen adversarially with respect to $\hat{h}$. Figure G.6 in the appendix gives a simple demonstration of this point, along with a detailed discussion.

A setting where DIS^2 may be invalid. There is one setting where Assumption 2.5 is less likely to be satisfied: when the representation we are using is regularized to keep $\max_{h' \in \mathcal{H}} \Delta(\hat{h}, h')$ small. This occurs for domain-adversarial training methods which penalize the ability to discriminate between $\mathcal{S}$ and $\mathcal{T}$ in feature space. It follows that for these methods DIS^2 should not be expected to hold universally, and we observe this in practice (Figure G.7). Nevertheless, when DIS^2 does overestimate accuracy, it does so by significantly less than prior methods.

3. Efficiently Maximizing the Discrepancy

For a classifier $\hat{h}$, Theorem 2.6 clearly prescribes how to bound its error; the difficulty remains in identifying the maximizing $h' \in \mathcal{H}$. We can approximately minimize $\epsilon_{\mathcal{S}}(\hat{h}, h')$ by minimizing the convex surrogate $\ell_{\text{log}} := -\log \text{softmax}(h(x))_y$ as justified by statistical learning theory, but it is less clear how to maximize $\epsilon_{\mathcal{T}}(\hat{h}, h')$. A few prior works suggest proxy losses for multiclass disagreement (Chuang et al., 2020; Pagliardini et al., 2023). We observe that these losses are not theoretically justified, as they do not upper bound the 0-1 disagreement loss we hope to minimize and are non-convex (or even concave) in the model logits. Instead, we derive a new loss which satisfies the above desiderata and thus serves as a more principled approach to maximizing disagreement.

Definition 3.1. The *disagreement logistic loss* of a classifier h on a labeled sample (x, y) is defined as $\ell_{\text{dis}}(h(x), y) := \frac{1}{\log 2} \log\left(1 + \exp\left(h(x)_y - \frac{1}{|\mathcal{Y}|-1}\sum_{\hat{y} \neq y} h(x)_{\hat{y}}\right)\right)$.

Fact 3.2. The disagreement logistic loss is convex in $h(x)$ and upper bounds the 0-1 disagreement loss (i.e., $\mathbf{1}\{\arg\max_{\hat{y}} h(x)_{\hat{y}} = y\}$). For binary classification, it is equivalent to the logistic loss with the label flipped.

We expect that ℓ_{dis} can serve as a useful drop-in replacement for any future algorithm which requires maximizing disagreement in a principled manner. We combine ℓ_{log} and ℓ_{dis} to get the empirical disagreement discrepancy objective:

$$\hat{\mathcal{L}} := \frac{1}{|\hat{\mathcal{S}}|}\sum_{\hat{\mathcal{S}}} \ell_{\text{log}}(h'(x), \hat{h}(x)) + \frac{1}{|\hat{\mathcal{T}}|}\sum_{\hat{\mathcal{T}}} \ell_{\text{dis}}(h'(x), \hat{h}(x)).$$

In practice we optimize this objective with multiple initializations and hyperparameters and select the solution with the largest empirical discrepancy on a holdout set to ensure a conservative bound. Experimentally, we find that replacing ℓ_{dis} with either of the surrogate losses from (Chuang et al., 2020; Pagliardini et al., 2023) results in smaller discrepancy; we present these results in Appendix B.

Tightening the bound by optimizing over logits. It is clear that the value of the bound in Theorem 2.6 will decrease as $\mathcal{H}$ is restricted. Since the number of features is large, one may expect that Assumption 2.5 holds even for a

	MAE (↓)		Coverage (↑)		Overest. (↓)	
DA?	✗	✓	✗	✓	✗	✓
Prediction Method						
AC (Guo et al., 2017)	0.1055	0.1077	0.1222	0.0167	0.1178	0.1089
DoC (Guillory et al., 2021)	0.1046	0.1091	0.1667	0.0167	0.1224	0.1104
ATC NE (Garg et al., 2022)	0.0670	0.0838	0.3000	0.1833	0.0842	0.0999
COT (Lu et al., 2023)	0.0689	0.0812	0.2556	0.1833	0.0851	0.0973
DIS^2 (Features)	0.2807	0.1918	1.0000	1.0000	0.0000	0.0000
DIS^2 (Logits)	0.1504	0.0935	0.9889	0.7500	0.0011	0.0534
DIS^2 (Logits w/o δ)	0.0829	0.0639	0.7556	0.4167	0.0724	0.0888

Table 1. **Comparing the DIS^2 bound to prior methods for predicting accuracy.** DA denotes if the representations were learned via a domain-adversarial algorithm. In addition to mean absolute error (MAE), we report what fraction of predictions correctly bound the true error (Coverage), and the average prediction error among shifts whose accuracy is overestimated (Overest.). DIS^2 has reasonably competitive MAE but substantially higher coverage. By dropping the concentration term in Theorem 2.6 we can do even better, at some cost to coverage.

reduced feature set. In particular, it is well documented that deep networks experience *neural collapse* (Papyan et al., 2020), giving representations whose effective rank is approximately equal to the number of classes. This suggests that the logits themselves should contain most of the features' information about $\mathcal{S}$ and $\mathcal{T}$. To test this, we evaluate DIS^2 on the full features, the logits output by $\hat{h}$, and various fractions of the top principal components (PCs) of the features. We observe that using logits indeed results in tighter error bounds *while still remaining valid*—in contrast, using fewer top PCs also results in smaller error bounds, but at some point they become invalid (Figure C.2). The bounds we report in this work are thus evaluated on the logits of $\hat{h}$, except where we provide explicit comparisons in Section 4.

Identifying the ideal number of PCs via a "validity score". Even though reducing the feature dimensionality eventually results in an invalid bound, we may hope to identify approximately when this occurs, giving a more accurate (though less conservative) prediction. We find that *the optimization trajectory itself* provides meaningful signal about this change. We design a "validity score" which captures this information and we observe that it is roughly linearly correlated with the tightness of the bound (Figure C.4). We can thus evaluate DIS^2 with successively fewer PCs and only retain those above a certain score threshold, reducing MAE while remaining reasonably conservative (Figure C.5). For further details, see Appendix C.

4. Experiments

Datasets. We conduct experiments across 11 diverse vision benchmark datasets for distribution shift on datasets that span applications in object classification, satellite imagery, and medicine. Because distribution shifts vary widely in scope, prior evaluations which focus on only one specific type of shift (e.g., corruptions) or algorithm often do not convey the full story. **We therefore emphasize the need for more comprehensive evaluations across many different types of shifts and training methods**, as we present here. We also experiment with Unsupervised Domain Adaptation (UDA) methods which aim to improve target performance with unlabeled target data.

Methods and metrics. We compare DIS^2 to four competitive baselines: *Average Confidence* (AC), *Difference of Confidences* (DoC), *Average Thresholded Confidence* (ATC), and *Confidence Optimal Transport* (COT). We give detailed descriptions of these methods in Appendix A. For DIS^2, we report bounds evaluated on both full features and logits as described in Section 3. Unless specified otherwise, we set $\delta = .01$ everywhere. We also experiment with dropping the lower order term in Theorem 2.6. As is standard, we report the *mean absolute error* (MAE)—since our emphasis is on conservative error bounds, we also report the *coverage*, i.e. the fraction of predictions for which the true error does not exceed the predicted error. Finally, we measure the average overestimation: this is the MAE among predictions which overestimate the accuracy.

Results. Table 1 reports metrics for all methods. We stratify only by whether the training method is domain-adversarial (DA), as this affects Assumption 2.5. We find that DIS^2 achieves competitive MAE while maintaining substantially higher coverage, even for DA features. When it does overestimate accuracy, it does so by much less, implying that it is ideal for conservative estimation even when any given error bound is not technically satisfied. We also visualize performance on individual distribution shifts, plotting each source-target pair as a single point for DA (Figure G.7) and non-DA methods (Figure 1). These plots do not include DoC, as it performed comparably to AC. Figure G.8(a) directly compares variants of DIS^2. Finally, Figure G.8(b) plots the observed violation rate (i.e. $1-$coverage) of DIS^2 on non-DA methods for varying δ. We observe that it lies at or below the line $y = x$, meaning the probabilistic bound in Theorem 2.6 holds across a range of failure probabilities.

Strengthening the baselines to improve coverage. Since the baselines prioritize predictive accuracy over conservative estimates, their coverage might be improvable without too much increase in error. We attempt this with a simple post-hoc adjustment in Appendix D. We find that (i) the baselines do not achieve the desired coverage level, though they get somewhat close; and (ii) the adjustment causes them to suffer higher MAE than DIS^2. Thus DIS^2 is on the Pareto frontier of MAE and coverage, and is preferable when conservative bounds are desirable.

5. Conclusion

The ability to evaluate *trustworthy*, non-vacuous error bounds for deep neural networks under distribution shift remains an extremely important open problem. Prior methods which estimate accuracy using extra information—such as unlabeled test samples—rely on opaque conditions whose likelihood of being satisfied is difficult to predict, and so they sometimes provide large overestimates of test accuracy with no warning signs. This work attempts to bridge this gap with a simple, intuitive condition and a new disagreement loss which together result in competitive error *prediction*, while simultaneously providing an (almost) provable probabilistic error *bound*. We also study how the process of evaluating the bound can provide even more useful signal. We expect there is potential to push further in each of these directions, hopefully extending the current accuracy-reliability Pareto frontier for test error bounds under distribution shift.

References

Christina Baek, Yiding Jiang, Aditi Raghunathan, and J Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. In *Advances in Neural Information Processing Systems*, 2022.

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 2018.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. *Advances in Neural Information Processing Systems*, 34: 14980–14992, 2021a.

Mayee Chen, Karan Goel, Nimit S Sohoni, Fait Poms, Kayvon Fatahalian, and Christopher Ré. Mandoline: Model evaluation under distribution shift. In *International Conference on Machine Learning*, pages 1617–1629. PMLR, 2021b.

Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Ching-Yao Chuang, Antonio Torralba, and Stefanie Jegelka. Estimating generalization under distribution shifts via domain-invariant representations. *International conference on machine learning*, 2020.

Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15069–15078, 2021.

Weijian Deng, Stephen Gould, and Liang Zheng. What does rotation prediction tell us about classifier accuracy under varying testing environments? *arXiv preprint arXiv:2106.05961*, 2021.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1): 2096–2030, jan 2016. ISSN 1532-4435.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, pages 7576–7586, 2018.

Saurabh Garg, Sivaraman Balakrishnan, Zachary Chase Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations*, 2022.

Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. *arXiv preprint arXiv:2107.03315*, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition, pages 4700–4708, 2017.

Junguang Jiang, Yang Shu, Jianmin Wang, and Mingsheng Long. Transferability in deep learning: A survey, 2022.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.

Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. Technical report, Citeseer, 2009.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Yuzhe Lu, Zhenlin Wang, Runtian Zhai, Soheil Kolouri, Joseph Campbell, and Katia Sycara. Predicting out-of-distribution error with confidence optimal transport. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. Agree to disagree: Diversity through disagreement for better transferability. In *The Eleventh International Conference on Learning Representations*, 2023.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Xingchao Peng, Ben Usman, Kuniaki Saito, Neela Kaushik, Judy Hoffman, and Kate Saenko. Syn2real: A new benchmark forsynthetic-to-real visual domain adaptation. *arXiv preprint arXiv:1806.09755*, 2018.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. Domain-adjusted regression or: Erm may already learn features sufficient for out-of-distribution generalization. *arXiv preprint arXiv:2202.06856*, 2022.

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*, 115(3):211–252, 2015.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*, 2020.

Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11): 1958–1970, 2008.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019.

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 (Guillory et al., 2021)) 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 (Guillory et al., 2021).

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 (Krizhevsky and Hinton, 2009) as the source dataset. For target domains, we consider (i) synthetic shifts (CIFAR10-C) due to common corruptions (Hendrycks and Dietterich, 2019); and (ii) natural distribution shift, i.e., CIFAR10v2 (Recht et al., 2018; Torralba et al., 2008) 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 (Koh et al., 2021; Christie et al., 2018) 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 (Bandi et al., 2018). 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 (Santurkar et al., 2020) 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 (Russakovsky et al., 2015), e.g, ImageNetv2 (Recht et al., 2019) 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 (Venkateswara et al., 2017). 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 (Peng et al., 2019). 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 (Peng et al., 2018). 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.

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.

A.3. Setup and Protocols

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

- CIFAR-10: Resnet-18 (He et al., 2016) pretrained on Imagenet
- CIFAR-100: Resnet-18 (He et al., 2016) pretrained on Imagenet
- Camelyon: Densenet-121 (Huang et al., 2017) *not* pretrained on Imagenet as per the suggestion made in (Koh et al., 2021)
- FMoW: Densenet-121 (Huang et al., 2017) pretrained on Imagenet
- BREEDs (Entity13, Entity30, Living17, Nonliving26): Resnet-18 (He et al., 2016) *not* pretrained on Imagenet as per the suggestion in (Santurkar et al., 2020). The main rationale is to avoid pre-training on the superset dataset where we are simulating sub-population shift.
- Officehome: Resnet-50 (He et al., 2016) pretrained on Imagenet
- Domainnet: Resnet-50 (He et al., 2016) pretrained on Imagenet
- Visda: Resnet-50 (He et al., 2016) pretrained on Imagenet

Except for Resnets on CIFAR datasets, we used the standard pytorch implementation (Gardner et al., 2018). 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 (Zhang, 2019). 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.

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

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 (Jiang et al., 2022). We tabulate hyperparameters for each algorithm next:

- **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:

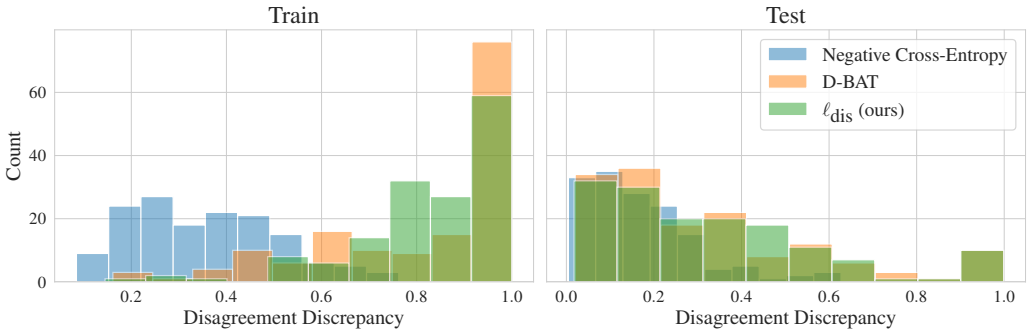
1. Chuang et al. (2020) minimize the negative cross-entropy loss, which is concave in the model logits. That is, they add the term $\log \text{softmax}(h(x)_y)$ to the objective they are minimizing. This loss results in substantially lower disagreement discrepancy than the other two.
2. Pagliardini et al. (2023) 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 (Chuang et al., 2020)	$0.3555 \pm .0124$	$0.1694 \pm .0105$
D-BAT (Pagliardini et al., 2023)	$0.8145 \pm .0177$	$0.3224 \pm .0212$
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.

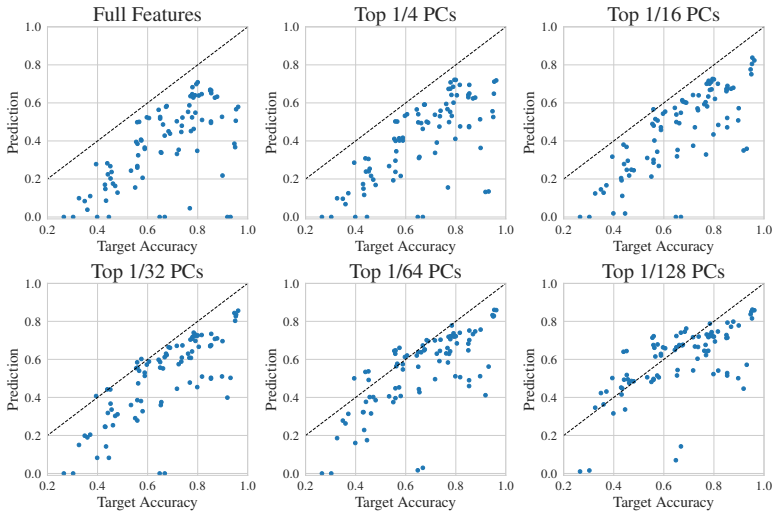


Figure C.2. **DIS^2 bound as fewer principal components are kept.** Reducing the number of top principal components crudely reduces complexity of $\mathcal{H}$—this leads to lower error estimates, but at some point the bounds become invalid for a large fraction of shifts.

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.

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. $1 - \epsilon_{\mathcal{S}}(\hat{h}, h'_i)$, 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 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.

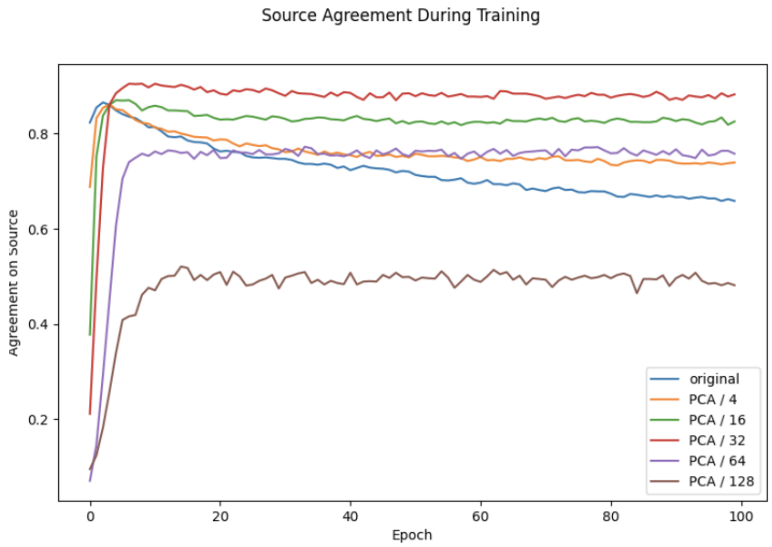


Figure C.3. **Agreement on one shift between $\hat{h}$ and h' on $\hat{\mathcal{S}}$ during optimization.** We observe that as the number of top PCs retained drops, the optimization occurs more slowly and less monotonically. For this particular shift, the bound becomes invalid when keeping only the top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

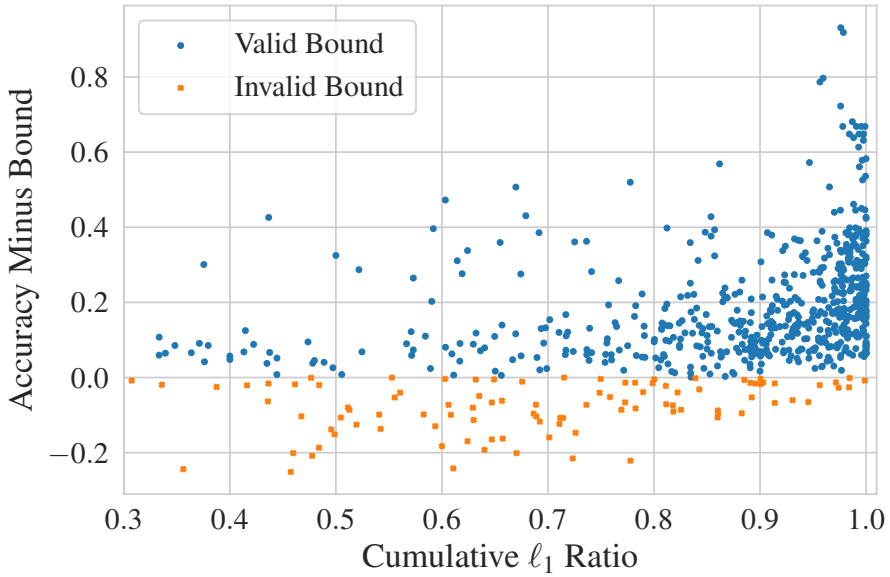


Figure C.4. **Cumulative ℓ_1 ratio versus error prediction gap.** Despite its simplicity, the ratio captures the information encoded in the optimization trajectory, roughly linearly correlating with the tightness and validity of a given prediction. It is thus a useful metric for identifying the ideal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

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.

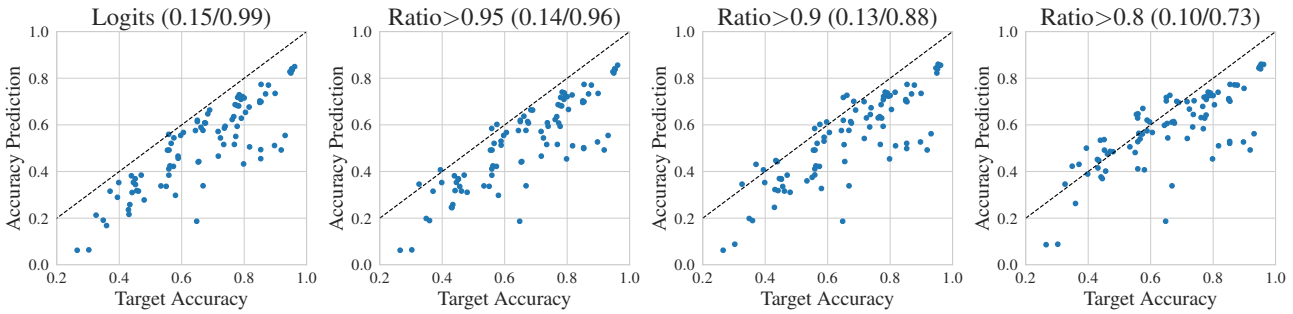


Figure C.5. **DIS^2 bounds and MAE / coverage as the cumulative ℓ_1 ratio threshold is lowered.** Values in parenthesis are (MAE / coverage). By only keeping predictions with ratio above a varying threshold, we can smoothly interpolate between bound validity and raw error prediction accuracy.

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 (Garg et al., 2022) and COT (Lu et al., 2023) 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 2.6, in the sense that it is an artifact of the random samples, rather than a property of the distribution (recall Figure G.8(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.

E. Proving that Assumption 2.5 Holds

Here we describe how the equivalence of Assumption 2.5 and the bound in Theorem 2.6 allow us to prove that the assumption holds with high probability. By repeating essentially the same proof as Theorem 2.6 in the other direction, we get the following corollary:

Corollary E.1. *If Assumption 2.5 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 2.6 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:

		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.

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 2.5 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 2.5—and therefore also Assumption 2.3—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 2.5 does not hold would be to show that for a fixed δ, the fraction of shifts for which the bound in Theorem 2.6 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 2.5 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 2.5 holds, because the gap between these two bounds will shrink with increasing sample size.

F. Related Work

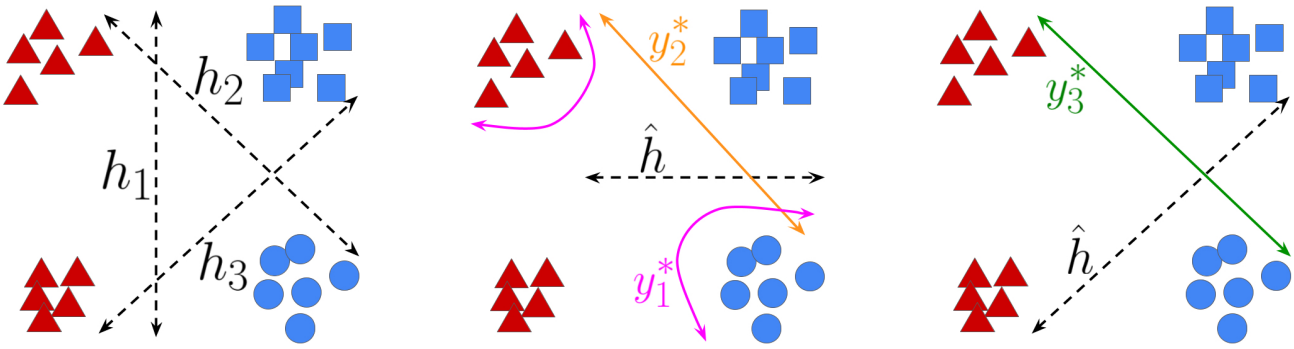
Estimating test error with unlabeled data. There are several methods that predict the error of a classifier under distribution shift with unlabeled test data: (i) methods that explicitly predict the correctness of the model on individual unlabeled points (Deng and Zheng, 2021; Deng et al., 2021; Chen et al., 2021a); and (ii) methods that directly estimate the overall error without making a pointwise prediction (Chen et al., 2021b; Guillory et al., 2021; Chuang et al., 2020; Garg et al., 2022; Baek et al., 2022). Many of these methods do not provide any sort of guarantee. Among those that do, it is common to require calibration on the target domain (Guillory et al., 2021). However, *evaluating* this property is impossible without test labels. Further, these methods often yield poor estimates because models calibrated on a source domain are not typically calibrated on new, unseen domains (Ovadia et al., 2019). Additionally, (Deng and Zheng, 2021; Guillory et al., 2021) require a subset of labeled target domains to learn a regression function—but this requires significant a priori knowledge about the nature of shift which, in practice, is usually not available before a model is deployed in the wild.

Closest to our work is (Chuang et al., 2020), where the authors use domain-invariant predictors as a proxy for unknown target labels. However, there are several crucial differences. First, like other works, their method only *estimates* the target accuracy—the error bounds they derive are not tractably computable. Second, their method relies on multiple approximations, numerous hyperparameters, and their algorithm is computationally demanding; as a result, it does not scale to modern deep networks. Finally, they suggest minimizing the (concave) negative cross-entropy loss to maximize disagreement; we propose a more suitable replacement which performs much better in practice.

Uniform convergence bounds. Our bound is inspired by classic analyses using $\mathcal{H}$- and $\mathcal{H}\Delta\mathcal{H}$-divergence (Mansour et al., 2009; Ben-David et al., 2006; 2010). These provide error bounds via a complexity measure that is both data- and hypothesis-class-dependent, but such bounds are often intractable to evaluate and are usually vacuous in real world settings.

G. Additional Figures and Discussion

G.1. How does DIS^2 Improve over $\mathcal{H}\Delta\mathcal{H}$-Divergence?



(a) (b) (c)

Figure G.6. **The advantage of DIS^2 over bounds based on $\mathcal{H}$- and $\mathcal{H}\Delta\mathcal{H}$-divergence.** Consider the task of classifying circles and squares (triangles are unlabeled). **(a):** Because h_1 and $h_2 \oplus h_3$ perfectly discriminate between $\mathcal{S}$ (blue) and $\mathcal{T}$ (red), $\mathcal{H}$- and $\mathcal{H}\Delta\mathcal{H}$-divergence bounds are always vacuous. In contrast, DIS^2 is only vacuous when 0% accuracy is induced by a reasonably likely ground truth (such as y_3^* in **(c)**, but not y_1^* in **(b)**), and can often give non-vacuous bounds (such as y_2^* in **(b)**).

Consider the task of learning a linear classifier to discriminate between squares and circles on the source distribution $\mathcal{S}$ (blue) and then bounding the error of this classifier on the target distribution $\mathcal{T}$ (red), whose true labels are unknown and are therefore depicted as triangles. Figure G.6(a) demonstrates that both $\mathcal{H}$- and $\mathcal{H}\Delta\mathcal{H}$-divergence achieve their maximal value of 1, because both h_1 and $h_2 \oplus h_3$ perfectly discriminate between $\mathcal{S}$ and $\mathcal{T}$. Thus both bounds would be vacuous.

Now, suppose we were to learn the max-margin $\hat{h}$ on the source distribution (Figure G.6(b)). It is *possible* that the true labels are given by the worst-case boundary as depicted by y_1^* (pink), thus "flipping" the labels and causing $\hat{h}$ to have 0 accuracy on $\mathcal{T}$. In this setting, a vacuous bound is correct. However, this seems rather unlikely to occur in practice—instead, recent experimental evidence (Kang et al., 2020; Rosenfeld et al., 2022; Kirichenko et al., 2022) suggests that the true y^* will be much simpler. The maximum disagreement discrepancy here would be approximately 0.5, giving a test accuracy lower bound of 0.5—this is consistent with plausible alternative labeling functions such as y_2^* (orange). Even if y^* is not linear,

we still expect that *some* linear function will induce larger discrepancy; this is precisely Assumption 2.3. Suppose instead we learn $\hat{h}$ as depicted in Figure G.6(c). Then a simple ground truth such as y_3^* (green) is plausible, which would mean $\hat{h}$ has 0 accuracy on $\mathcal{T}$. In this case, y_3^* is also a critic with disagreement discrepancy equal to 1, and so DIS^2 would correctly output an error upper bound of 1.

G.2. DIS^2 on Domain-Adversarial Training Methods

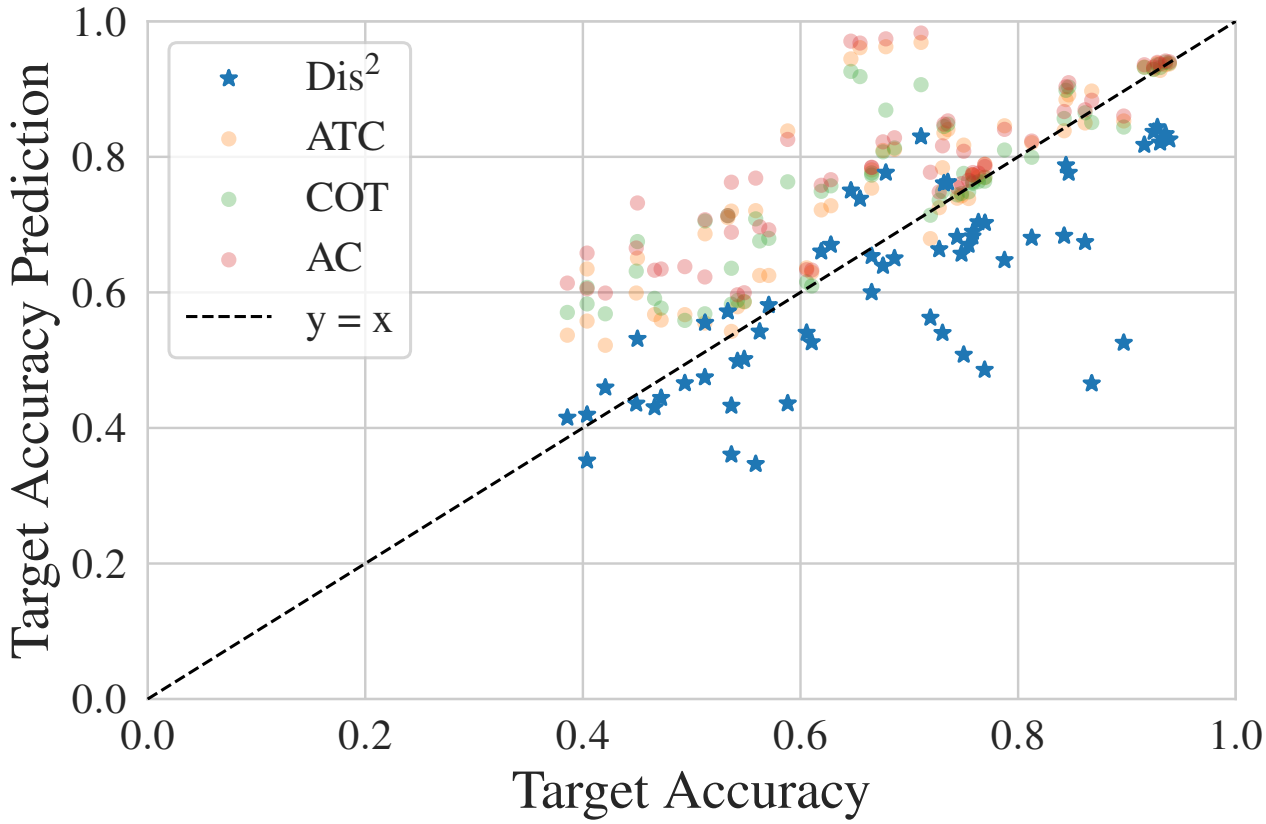


Figure G.7. **DIS^2 may be invalid when the features are explicitly learned to violate Assumption 2.5.** Domain-adversarial representation learning algorithms such as DANN (Ganin et al., 2016) and CDAN (Long et al., 2018) indirectly minimize $\max_{h' \in \mathcal{H}} \Delta(\hat{h}, h')$, meaning the necessary condition is less likely to be satisfied. Nevertheless, when DIS^2 does overestimate accuracy, it almost always does so by less than prior methods.

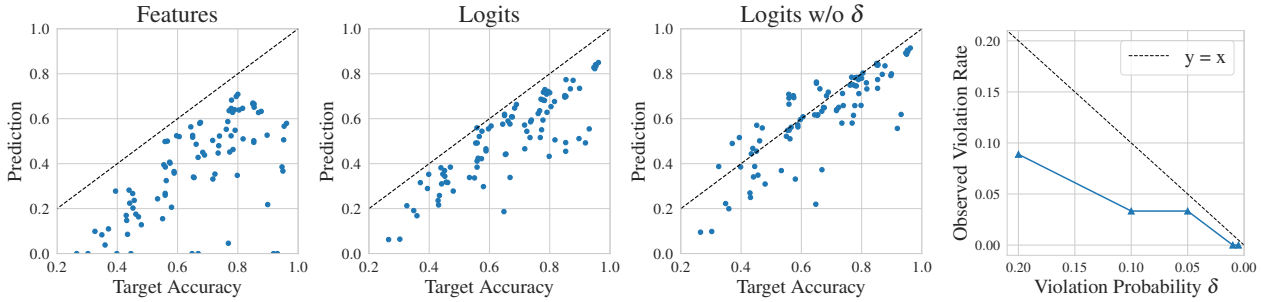
G.3. Comparing DIS^2 on Features vs. Logits and Observed Violation Rate

H. Proof of Theorem 2.6

Proof. Assumption 2.5 gives $\epsilon_{\mathcal{T}}(\hat{h}) \leq \epsilon_{\mathcal{S}}(\hat{h}) + \Delta(\hat{h}, h') = \epsilon_{\mathcal{S}}(\hat{h}, y^*) + \epsilon_{\mathcal{T}}(\hat{h}, h') - \epsilon_{\mathcal{S}}(\hat{h}, h')$. We now define the random variables for $\hat{\mathcal{S}} \cup \hat{\mathcal{T}}$:

$$r_i = \begin{cases} 1/n_S, & h'(x_i) = \hat{h}(x_i) \neq y_i,\ x_i \in \hat{\mathcal{S}} \\ -1/n_S, & h'(x_i) \neq \hat{h}(x_i) = y_i,\ x_i \in \hat{\mathcal{S}} \\ 1/n_T, & \hat{h}(x_i) \neq h'(x_i),\ x_i \in \hat{\mathcal{T}}, \\ 0, & \text{otherwise.} \end{cases}$$

Noting that the expectation of their sum is exactly the above three terms, we apply Hoeffding's inequality: the probability that the expectation exceeds their sum by t is no more than $\exp\left(-\frac{2t^2}{n_S(2/n_S)^2 + n_T(1/n_T)^2}\right)$. Now simply solve for t. □



(a) (b)

Figure G.8. **(a):** Scatter plots depicting DIS^2 estimated bound vs. true error for a variety of shifts. "w/o δ" indicates that the lower-order term of Theorem 2.6 has been dropped. **(b):** Observed bound violation rate vs. desired probability δ. Observe that the true rate lies at or below $y = x$ across a range of values.