A Learning Based Hypothesis Test for Harmful Covariate Shift

Tom Ginsberg & Zhongyuan Liang & Rahul G. Krishnan
Department of Computer Science
University of Toronto
Toronto, ON M5S 1A1
{tomginsberg,zhongyuanliang,rahukgk}@cs.toronto.edu

Abstract

Quickly and accurately identifying covariate shift at test time is a critical and often overlooked component of safe machine learning systems deployed in high-risk domains. In this work, we give an intuitive definition of *harmful covariate shift* (HCS) as a change in distribution that may weaken the generalization of a classification model. To detect HCS, we use the discordance between classifiers trained to agree on training data and disagree on test data. We derive a loss function for training these models and show that their disagreement rate and entropy represent powerful discriminative statistics for HCS. Empirically, we demonstrate the ability of our method to detect harmful covariate shift with statistical certainty on a variety of high-dimensional datasets. Across numerous domains and modalities, we show state-of-the-art performance compared to existing methods, particularly when the number of observed test samples is small. This work is part of an extended submission; link to project code.

1 Introduction

Machine learning models operate on the assumption, albeit incorrectly that they will be deployed on data distributed identically to what they were trained on. The violation of this assumption is known as distribution shift and can often result in significant degradation of performance (Bickel et al., 2009; Rabanser et al., 2019; Otles et al., 2021; Ovadia et al., 2019). There are several cases where a mismatch between training and deployment data results in very real consequences on human beings. In healthcare, machine learning models have been deployed for predicting the likelihood of sepsis. Yet, as (Habib et al., 2021) show, such models can be miscalibrated for large groups of individuals, directly affecting the quality of care they experience. The deployment of classifiers in the criminal justice system (Hao, 2019), hiring and recruitment pipelines (Dastin, 2018) and self-driving cars (Smiley, 2022) have all seen humans affected by the failures of learning models. The need for methods that quickly detect, characterize and respond to covariate shift is, therefore, a fundamental problem in trustworthy machine learning. In this work, we study a special case of

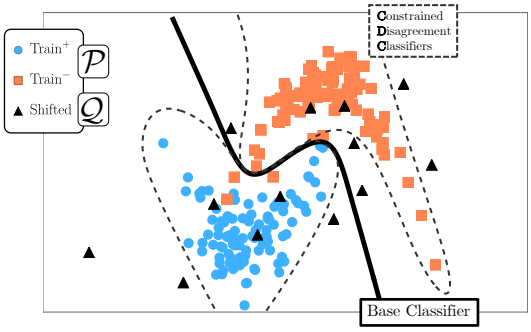

Figure 1: **Overview of Detectron:** Starting with a base classifier trained on labeled samples from distribution $\mathcal{P}$ we train new ***C**onstrained **D**isagreement **C**lassifiers* to maximize classification disagreement on $\mathcal{Q}$ while constrained to agree with the base classifier on $\mathcal{P}$. The rate that CDCs disagree, is a powerful and sample efficient statistic for identifying covariate shift $\mathcal{P} \neq \mathcal{Q}$.

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

distribution shift, commonly known as *covariate shift*, which considers shifts only in the distribution of input data $\mathbb{P}(X)$ while the relation between the inputs and outputs $\mathbb{P}(Y|X)$ remains fixed.

Our work develops a practical, model-based hypothesis test, named *The Detectron*, to identify potentially harmful covariate shifts given any existing classification model already in deployment.

We make the following key contributions:

- We show how to construct classifiers that maximize out-of-domain disagreement while behaving consistently in the training domain. We propose the *disagreement cross entropy* for models learned via continuous gradient-based methods (e.g., neural networks), as well as a generalization for those learned via discrete optimization (e.g., random forest).
- We show that the rejection rate and the entropy of the learning ensemble can be used to define a model-aware hypothesis test for covariate shift, the Detectron.
- On high-dimensional image and tabular data, using both neural networks and gradient boosted decision trees, our method outperforms state-of-the-art techniques for detecting covariate shifts, particularly when given access to as few as ten test examples.

2 Detectron

Problem Setup. Let $f : X \to Y$ be classification model from a function class F that maps from space of covariates X to a discrete set of classes $Y = \{1, \ldots, N\}$. We assume f was trained on a labeled dataset $\mathbf{P} = \{(x_i, y_i)\}_{i=1}^n$ where each x_i is drawn identically from a distribution $\mathcal{P}$ over X. In deployment, f is then made to predict on new unlabeled samples $\mathbf{Q}$ from a distribution $\mathcal{Q}$ over X. Our goal is to determine whether f may be trusted to do so accurately. The problem we address is how to automatically detect, from only a set of finite samples $\mathbf{Q} = \{\tilde{x}_i\}_{i=1}^m$, if the new covariate distribution $\mathcal{Q}$ has shifted from $\mathcal{P}$ in such a way that f can no longer be assumed to generalize — we refer to this type of dataset shift as *harmful covariate shift*.

Harmful Covariate Shift. A shift in the data distribution is not always harmful. In many practical problems, a practitioner may use domain knowledge to embed invariances with the explicit goal of ensuring the predictive performance of a classifier does not, by construction, change under certain shifts. This may be done directly via translation invariance in convolutional neural networks or indirectly via data augmentation or domain adaptation. Such practical heuristics can lead to models generalizing to a more broad range of distributions than can be characterized by just $\mathbf{P}$. We refer to such an induced generalization set as $\mathcal{R}$. Although $\mathcal{R}$ is difficult to characterize and will in general depend on the model architecture, learning algorithm and training dataset, we seek a practical method for detecting shift that is explicitly tied to $\mathcal{R}$. Our approach is based intuition from learning theory: if there exists a set of classifiers with the same generalization set $\mathcal{R}$ but behave inconsistently on samples from a distribution $\mathcal{Q}$, then $\mathcal{Q}$ must not be a member $\mathcal{R}$. Our strategy will be to create an ensemble of *constrained disagreement classifiers* (CDCs), classifiers constrained to predict consistently (i.e., predict the same as f) on $\mathcal{R}$ but as differently as possible on $\mathcal{Q}$. If $\mathcal{Q}$ is within $\mathcal{R}$ then such an ensemble will fail to predict differently. Hence, when we can find an ensemble that exhibits inconsistent behaviour on $\mathcal{Q}$, there must be covariate shift that explicitly lies outside $\mathcal{R}$. See Appendix B for a formal definition of harmful covariate shift.

Learning to Disagree. We introduce the *disagreement cross entropy* (DCE) as a smooth objective function to encourage a continuously optimized classifier to disagree with a set of labels. Expressed in Equation 5, DCE corresponds to the cross entropy between the predicted probability vector $g(x)$ from a classifier g and the uniform distribution over all classes except y.

$$\text{DCE}(g(x), y) = \frac{1}{1-N} \sum_{c=1}^{N} \delta_{y \neq c} \log\left(g(x)_c\right) \tag{1}$$

When training CDCs in practice we fine tune the pretrained classifier f using a loss function composed both of the original training objective for f along with the DCE loss. This process create a classifier that behaves as well as f on $\mathbf{P}$, but as differently as possible on $\mathbf{Q}$. See Appendix C for a more detailed description of CDC learning including a generalization to arbitrary black box models.

The Detectron Test for Harmful Shift.

Our proposed method is to train two CDCs; $g_{\mathbf{Q}}$ and $g_{\mathbf{P}}$. First, $g_{\mathbf{Q}}$ is trained to disagree with f on $\mathbf{Q}$, while still learning the original training objective. Next, we train $g_{\mathbf{P}}$ as a baseline to disagree on set of unseen samples $\mathbf{P}^\star$ drawn from $\mathcal{P}$ where $m := |\mathbf{P}^\star| = |\mathbf{Q}|$. To maximize sample efficiency, we take a transductive approach to detecting shift by analyzing the outputs of $g_{\mathbf{Q}}$ and $g_{\mathbf{P}}$ on the sets $\mathbf{Q}$ and $\mathbf{P}^\star$ themselves. We define the number of samples from 0 to m that $g_{\mathbf{P}}$ disagrees on with respect to f as $\phi_{\mathbf{P}}$, and similarly for $g_{\mathbf{Q}}$ as $\phi_{\mathbf{Q}}$. Note that $\phi_{\mathbf{P}}$ and $\phi_{\mathbf{Q}}$ are themselves random variables based both on sets $\mathbf{Q}/\mathbf{P}^\star$ as well as the dynamics of the CDC learning algorithm. Under the null hypothesis, if $\mathcal{Q}$ is not a harmful shift from $\mathcal{P}$, we have $\mathcal{H}_0 : \mathbb{E}[\phi_{\mathbf{Q}}] \leq \mathbb{E}[\phi_{\mathbf{P}}]$. Harmful shift is expressed as the one-sided alternative $\mathcal{H}_a : \mathbb{E}[\phi_{\mathbf{Q}}] > \mathbb{E}[\phi_{\mathbf{P}}]$ (i.e., $g_{\mathbf{Q}}$ is expected to disagree on more samples than $g_{\mathbf{P}}$). See Figure 2 for a visual depiction of CDC training.

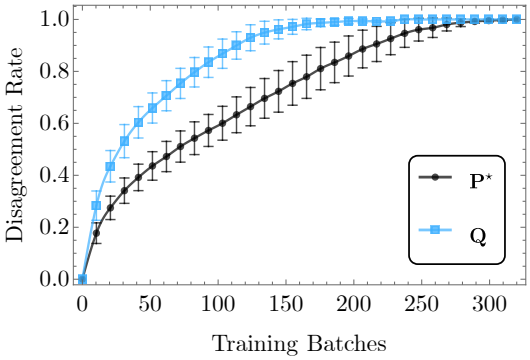

Figure 2: **CDC Training Dynamics:** In blue we train CDCs to disagree on a set of 100 samples from CIFAR 10.1 ($\mathbf{Q}$) – a near OOD test set for CIFAR 10 – while in black we force CDCs to disagree on the original CIFAR 10 test set ($\mathbf{P}^\star$). We see that even after a small number of training batches all models disagree on a significantly larger portion of CIFAR 10.1 compared to CIFAR 10

We refer to this test as *Detectron Disagreement*. To compute the test result at a significance level α we first estimate the null distribution of $\phi_{\mathbf{P}}$ for a fixed sample size m by training K calibration rounds of $g_{\mathbf{P}}$ with different random sets $\mathbf{P}^\star$ of size m. The test result is significant if the observed disagreement rate $\phi_{\mathbf{Q}}$ is greater than the $(1-\alpha)$ quantile of the null distribution.

We consider an additional variant, *Detectron Entropy* (DE), which computes the prediction entropy of each sample under the CDC instead of relying solely on disagreement rates. The intuition for DE draws from the fact that when CDCs satisfy their objective (i.e., in the case of harmful shift) they learn to predict with high entropy on $\mathbf{Q}$ and low entropy on $\mathbf{P}^\star$, resulting in a natural way to distinguish between distributions. See Appendix D for a detailed description of both tests.

3 Empirical Evaluation

Our experiments are carried out on natural distribution shifts across multiple domains, modalities, and model types. We use the *CIFAR-10.1* dataset (Recht et al., 2019) where shift comes from subtle changes in the dataset creation processes, the *Camelyon17 dataset* (Veeling et al., 2018) for metastases detection in histopathological slides from multiple hospitals, as well as the *UCI heart disease* dataset (Janosi et al., 1988) which contains tabular features collected across international health systems and indicators of heart disease.

Shift Detection Setup. We evaluate the Detectron in a standard two-sample testing scenario similar to prior work (Zhao et al., 2022). Given two datasets $\mathbf{P}$ (drawn from $\mathcal{P}$) and $\mathbf{Q}$ (drawn from $\mathcal{Q}$) and classifier f, we seek to rule out the null hypothesis ($\mathcal{P} = \mathcal{Q}$) at the 5% significance level. To guarantee fixed significance we employ a permutation test by first sampling from the distribution of p-values derived where the null hypothesis $\mathcal{P} = \mathcal{Q}$ holds (i.e., $\mathbf{Q}$ is drawn $\mathcal{P}$). We then compute a threshold over the observed test statistic that sets the false positive rate to 5%. For each dataset, we begin by training a base classifier on the unshifted dataset. We evaluate the detection of covariate shift on 100 randomly selected test sets of $m = 10$, 20 and 50 samples from $\mathcal{Q}$. For ensemble methods, we use an ensemble size of 5. Hyperparameters and training details for all models can be found in Appendix F.

Evaluation. We report the *True Positive Rate at 5% Significance Level (TPR@5)* aggregated over 100 randomly selected sets $\mathbf{Q}$. This signifies how often our method correctly identifies covariate shift ($\mathcal{P} \neq \mathcal{Q}$) while only incorrectly identifying shift 5% of the time.

Baselines. We compare the Detectron against several methods for OOD detection, uncertainty estimation and covariate shift detection. See subsection F.3 for further details and citations for each method in Table 1.

Results. The TPR@5 for the detection of harmful shift with sample sizes of 10, 20 and 50 on all datasets are shown in Table 1. We report the mean and standard error over 100 random runs.

Table 1: **Results (true positive rate at the 5% significance level) for detection of harmful covariate shift** on CIFAR-10.1, Camelyon 17 and UCI Heart Disease benchmarks. The **best** result for each column is bolded, results that are within 2% of the best are underlined and the *best baseline* method is italicized.

	CIFAR 10.1			**Camelyon 17**			**UCI Heart Disease**		
$\|Q\|$	10	20	50	10	20	50	10	20	50
BBSD	$.07 \pm .03$	$.05 \pm .02$	$.12 \pm .03$	$.16 \pm .04$	$.38 \pm .05$	$.87 \pm .03$	$.13 \pm .03$	$.22 \pm .04$	$.46 \pm .05$
Rel. Mahalanobis	$.05 \pm .02$	$.03 \pm .03$	$.04 \pm .02$	$.16 \pm .04$	$.40 \pm .05$	*$.89 \pm .03$*	$.11 \pm .03$	$.36 \pm .05$	$.66 \pm .05$
Deep Ensemble (Dis)	$.23 \pm .04$	$.40 \pm .05$	*$.74 \pm .04$*	$.10 \pm .03$	$.11 \pm .03$	$.13 \pm .03$	$.02 \pm .01$	$.00 \pm .00$	$.32 \pm .05$
Deep Ensemble (Entropy)	*$.33 \pm .05$*	*$.52 \pm .05$*	$.68 \pm .05$	$.14 \pm .03$	$.26 \pm .04$	$.82 \pm .04$	$.13 \pm .03$	$.32 \pm .05$	$.64 \pm .05$
CTST	$.03 \pm .02$	$.04 \pm .02$	$.04 \pm .02$	$.11 \pm .03$	$.59 \pm .05$	$.59 \pm .05$	*$.15 \pm .04$*	*$.51 \pm .05$*	*$\underline{.98 \pm .01}$*
MMD-D	$.24 \pm .04$	$.10 \pm .03$	$.05 \pm .02$	*$.42 \pm .05$*	*$.62 \pm .05$*	$.69 \pm .05$	$.09 \pm .03$	$.12 \pm .03$	$.27 \pm .04$
H-Div	$.02 \pm .01$	$.05 \pm .02$	$.04 \pm .02$	$.03 \pm .02$	$.07 \pm .03$	$.23 \pm .04$	*$.15 \pm .04$*	$.26 \pm .04$	$.37 \pm .05$
Detectron (Dis)	$\mathbf{.37 \pm .05}$	$\underline{.54 \pm .05}$	$.83 \pm .04$	$\mathbf{.97 \pm .02}$	$\mathbf{1.0 \pm .00}$	$.96 \pm .02$	$.24 \pm 0.04$	$.57 \pm 0.05$	$.82 \pm 0.04$
Detectron (Entropy)	$\underline{.35 \pm .05}$	$\mathbf{.56 \pm .05}$	$\mathbf{.92 \pm .03}$	$\mathbf{.97 \pm .02}$	$\mathbf{1.0 \pm .00}$	$\mathbf{1.0 \pm .00}$	$\mathbf{.45 \pm .05}$	$\mathbf{.88 \pm 0.03}$	$\mathbf{1.0 \pm .00}$

We observe in the bottom rows of Table 1 that Detectron methods outperform all baselines across all three tasks confirming our intuition that distribution tests based on transductive properties of learning algorithms is a promising avenue of research. See Appendix G for an extended discussion of the experimental results

4 Conclusion, Limitations and Future Work

Our work presents a practical method capable of detecting covariate shifts given a pre-existing classifier. We showcase the efficacy of our method in being sensitive enough to detect covariate shift using a small number of unlabelled examples across several real-world datasets. We remark on several characteristics of our algorithm that represent potential directions for future work:

On Computational Cost: Our method is more computationally expensive than some existing methods such as BBSD and Mahalanobis Scores, but is similar complexity to other approaches such as Ensembles, MMD-D and H-Divergence which may require training multiple deep models. However, as the Detectron leverages a pretrained model already in deployment, we find in practice that only a small number of training rounds are required to create each CDC. For instance, on CIFAR 10/10.1 a CDC using a ResNet 18 architecture can train in under $\approx$ 30s using an unoptimized PyTorch implementation on 1 GPU. We present a deeper analysis of the runtme behaviour in Appendix H.

Beyond Classification: Our work here focuses on the case of classification (since a large number of pre-existing benchmarks in the literature focus on the same). However, we believe there is a viable extension of our work to regression models where constrained predictors are explicitly learned to maximize test error according to the existing metric, such as mean squared error. We leave this exploration for future work.

Beyond Covariate Shifts: While covariate shift is the only type of shift that can be discovered from unlabeled data without additional assumptions, we acknowledge that other types of shift, such as label and concept shift, are prevalent in the real world. Building learning-based methods to identify these types of shifts is another direction for future work.

Finally, we wish to highlight that while auditing systems such as the Detectron show promise to ease concerns when using learning systems in high-risk domains, practitioners interfacing with these systems should not place blind trust in their outputs.

References

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/file/b1b0432ceafb0ce714426e9114852ac7-Paper.pdf.

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(75):2137–2155, 2009. URL http://jmlr.org/papers/v10/bickel09a.html.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *CoRR*, abs/2110.01889, 2021. URL https://arxiv.org/abs/2110.01889.

Dipankar Chaki, Arkadeep Das, and Moinul Islam Zaber. A comparison of three discrete methods for classification of heart disease data. *Bangladesh Journal of Scientific and Industrial Research*, 50:293–296, 2015.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL http://doi.acm.org/10.1145/2939672.2939785.

Jeffrey Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G, 2018. (Accessed on 05/11/2022).

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Shafi Goldwasser, Adam Tauman Kalai, Yael Kalai, and Omar Montasser. Beyond Perturbations: Learning Guarantees with Arbitrary Adversarial Test Examples. In H Larochelle, M Ranzato, R Hadsell, M F Balcan, and H Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15859–15870. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/b6c8cf4c587f2ead0c08955ee6e2502b-Paper.pdf.

Anand R. Habib, Anthony L. Lin, and Richard W. Grant. The Epic Sepsis Model Falls Short—The Importance of External Validation. *JAMA Internal Medicine*, 181(8):1040–1041, 08 2021.

Karen Hao. Ai is sending people to jail—and getting it wrong. https://www.technologyreview.com/2019/01/21/137783/algorithms-criminal-justice-ai/, 2019. (Accessed on 05/11/2022).

David Haussler. Probably approximately correct learning. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pp. 1101–1108. AAAI Press, 1990.

J. L. Hodges. The significance probability of the smirnov two-sample test. *Arkiv för Matematik*, 3(5): 469–486, 1958. doi: 10.1007/BF02589501. URL https://doi.org/10.1007/BF02589501.

Wolfram Research, Inc. Mathematica, Version 13.0.0. URL https://www.wolfram.com/mathematica. Champaign, IL, 2021.

Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. Heart Disease. UCI Machine Learning Repository, 1988.

Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pp. 180–191. VLDB Endowment, 2004. ISBN 0120884690.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

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 Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5637–5664. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/koh21a.html.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5), 2014.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1VGkIxRZ.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pp. 3122–3130. PMLR, 2018.

Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J Sutherland. Learning deep kernels for non-parametric two-sample tests. In *International conference on machine learning*, pp. 6316–6326. PMLR, 2020.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJkXfE5xx.

Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pp. 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589336. doi: 10.1145/1873951.1874254. URL https://doi.org/10.1145/1873951.1874254.

Warren Morningstar, Cusuh Ham, Andrew Gallagher, Balaji Lakshminarayanan, Alex Alemi, and Joshua Dillon. Density of states estimation for out of distribution detection. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3232–3240. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/morningstar21a.html.

Erkin Otles, Jeeheh Oh, Benjamin Li, Michelle Bochinski, Hyeon Joo, Justin Ortwine, Erica Shenoy, Laraine Washer, Vincent B Young, Krishna Rao, et al. Mind the performance gap: examining dataset shift during prospective validation. In *Machine Learning for Healthcare Conference*, pp. 506–534. PMLR, 2021.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Stephan Rabanser, Stephan Günnemann, and Zachary C. Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 1394–1406, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/846c260d715e5b854ffad5f70a516c88-Abstract.html.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10?, 2018. URL https://arxiv.org/abs/1806.00451.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/recht19a.html.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, 32, 2019.

Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *ICML workshop on Uncertainty and Robustness in Deep Learning*, 2021. URL http://www.gatsby.ucl.ac.uk/~balaji/udl2021/accepted-papers/UDL2021-paper-007.pdf.

Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with Gram matrices. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8491–8501. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/sastry20a.html.

Lauren Smiley. 'i'm the operator': The aftermath of a self-driving tragedy. https://www.wired.com/story/uber-self-driving-car-fatal-crash/, 2022. (Accessed on 05/11/2022).

Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. June 2018.

Lily Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. In *International Conference on Machine Learning*, pp. 12427–12436. PMLR, 2021.

Shengjia Zhao, Abhishek Sinha, Yutong He, Aidan Perreault, Jiaming Song, and Stefano Ermon. Comparing distributions by measuring differences that affect decision making. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KB5onONJIAU.

Appendix

A Related Work

A.1 Covariate Shift Detection

Covariate shift is the tendency for a distribution at test time $p_{\text{test}}(x)$ to differ from that seen during training $p_{\text{train}}(x)$ while the underlying prediction concept y remains fixed e.g. $p_{\text{train}}(y|x) = p_{\text{test}}(y|x)$. Many methods for detecting shift apply dimensionality reduction followed by statistical hypothesis tests for distributional differences in the outputs (from a reference and target) (Rabanser et al., 2019). Rabanser et al. show that using the softmax outputs of a pretrained classifier as low dimensional representations for performing univariate KS-tests, a method known as black box shift detection (BBSD) (Lipton et al., 2018), is effective at confidently identifying several synthetic covariate shifts in imaging data (e.g. crops, rotations) given approximately 200 i.i.d samples. However, applying statistical tests to non-invertible representations of data can never guarantee to capture arbitrary covariate shifts, as there may always exist multiple distributions that collapse to the same test statistic (Zhang et al., 2021). Kifer et al. (2004); Ben-David et al. (2006) introduce some of the earliest learning theoretic approaches for identifying and correcting for covariate shift based on discriminative learning with finite samples. More recent approaches for covariate shift detection including classifier two sample tests (Lopez-Paz & Oquab, 2017), deep kernel MMD (Liu et al., 2020) and H-Divergence (Zhao et al., 2022) rely on analyzing the outputs of unsupervised learning models. In our work we take a transductive learning approach and construct a method to directly use the structure of a supervised classification problem to improve the statistical power for detecting shifts.

A.2 Out of Distribution Detection

Out of distribution (OOD) detection focuses on identifying when a specific data point x' admits low likelihood under the original training distributions ($p_{\text{train}}(x') \approx 0$)—a useful tool to have at inference time. Ren et al. (2019); Morningstar et al. (2021) represent a broad class of work that uses density estimation to pose the identification of covariate shift as anomaly detection. However, in finite samples, density estimation for high-dimensional data can be difficult which in turn affects the accuracy of anomaly detection (Zhang et al., 2021). Others, including ODIN (Liang et al., 2018), Deep Mahalanobis Detectors (Lee et al., 2018) and, Gram Matrices (Sastry & Oore, 2020) directly use the predictive model (e.g. information from the intermediate representations of neural networks). Such methods are largely based on heuristics on the manifold of neural networks offering little to no theoretical guarantees on detecting subtle types of covariate shifts encountered in real-world settings. Furthermore, the majority of methods in this space have been designed exclusively for deep neural networks, an uncommon modelling choice particularly for tabular data (Borisov et al., 2021). Related to OOD, uncertainty estimation concerns developing models that identify sources of uncertainty in their predictions (Lakshminarayanan et al., 2017; Ovadia et al., 2019). Naturally, uncertainty should be large when samples are OOD, however (Ovadia et al., 2019) perform a large-scale empirical comparison of uncertainty estimation methods and find that while deep ensembles generally provide the best results, the quality of uncertainty estimations, regardless of method, consistently degrades with increasing covariate shift.

A.3 Selective Classification and PQ Learning

Selective classification concerns building classifiers that may either predict on or reject on test samples (Geifman & El-Yaniv, 2019). Recent work by (Goldwasser et al., 2020) develops a formal framework known as PQ learning which extends probably approximately correct (PAC) learning (Haussler, 1990) to arbitrary test distributions by allowing for selective classification. While PAC learning concerns the development of a classifier with a bounded finite-sample error rate on its training distribution, PQ learning seeks a selective classifier with jointly bounded finite-sample error and rejection rates on arbitrary test distributions. The Rejectron algorithms proposed therein builds an ensemble of models that produce different outputs relative to a perfect baseline on a set of unlabeled

test samples. PQ-learning represents a major theoretical leap for learning guarantees under covariate shift; however, the majority of the underlying ideas have not been implemented/tested experimentally using real-world data. We show how to build a PQ learner by generalizing the Rejectron algorithm, overcoming several limitations and assumptions made by the original work including extending beyond simple binary classification to general multiclass/multilabel tasks and reducing the number of samples required for learning at each iteration. We go on to show how a PQ learner can be used to characterize covariate shifts in real-world data.

B Harmful Covariate Shift and Connection to Classical Learning Theory

Given a labeled training set $\mathbf{P}$ as well as another labeled dataset $\mathbf{Q}$, one can identify using standard statistical estimation if a model f performs more poorly on $\mathbf{Q}$ compared to $\mathbf{P}$. However, in a practical scenario, decision models are deployed on unlabeled datasets; hence directly computing model performance is impossible. To decide then if $\mathbf{Q}$ has been drawn from a distribution that may cause f to fail, we formulate an adversarial learning style definition of *harmful covariate shift* that does not require access to labeled examples.

Definition: (ℓ, α, F)-Harmful Covariate Shift.
A covariate shift from distributions $\mathcal{P} \to \mathcal{Q}$ over X is (ℓ, α, F)-harmful with respect to a set of decision models F, if there exists any subset of models of two or more models $\mathbf{f}$ in F that achieve a source domain loss $\ell(f, \mathcal{P}) \leq \alpha$ for all $f \in \mathbf{f}$ while being more likely to disagree with each other on an unseen sample from $\mathcal{Q}$ compared to $\mathcal{P}$.

$$\begin{aligned} &\exists\, \mathbf{f} \subseteq F, \text{ s.t. } \forall f \in \mathbf{f}\ \ell(f, \mathcal{P}) \geq \alpha \text{ and} \\ &\qquad \mathbb{P}_{x\sim\mathcal{Q}}(\exists\, f_i, f_j \in \mathbf{f} \text{ s.t. } f_i(x) \neq f_j(x)) > \mathbb{P}_{x\sim\mathcal{P}}(\exists\, f_i, f_j \in \mathbf{f} \text{ s.t. } f_i(x) \neq f_j(x)) \end{aligned} \tag{2}$$

In plainer words, we define harmful covariate shift based on the existence of multiple *good* models on $\mathcal{P}$ that tend to disagree on $\mathcal{Q}$. The Detectron algorithm is designed to learn these models (constrained disagreement classifiers) and statistically test their disagreement rates.

We can connect our definition of harmfulness to the well-studied concept of $\mathcal{A}$ distance from Kifer et al. (2004). The $\mathcal{A}$ is a generalization of the total variation to an arbitrary collection of measurable events $\mathcal{A}$.

$$d_{\mathcal{A}}(\mathcal{P}, \mathcal{Q}) = 2 \sup_{A \in \mathcal{A}} |\Pr_{\mathcal{D}}[A] - \Pr_{\mathcal{D}'}[A]| \tag{3}$$

Ben-David et al. (2006) shows that when they chose a class of events whose characteristic functions are functions in F, the $\mathcal{A}$ distance in connection with VC theory (Vapnik, 1995) allows for finite sample generalization bounds on the performance of arbitrary decision models from F under covariate shift. Ben-David et al. (2006) go on to show that the $\mathcal{A}$ distance defined for a binary function class F is equal to

$$d_F(\mathcal{P}, \mathcal{Q}) = 2\left(1 - 2 \min_{f \in F} \text{err}(f)\right) \tag{4}$$

where $\min_{f\in F} \text{err}(f)$ is the minimum error that a domain classifier from F can achieve on the task of distinguishing samples from $\mathcal{P}$ and $\mathcal{Q}$ (i.e. if $\mathcal{P} = \mathcal{Q}$ the best domain classifier will have error of 0.5 and $d_F(\mathcal{P}, \mathcal{Q}) = 0$ and if $\mathcal{P}$ and $\mathcal{Q}$ can be perfectly discriminated by some $f \in F$ the $d_F(\mathcal{P}, \mathcal{Q})$ is maximized and equal to 2). In our characterization of harmful covariate shift, we consider not just the discriminative power of F but the broader generalization region induced by training f to achieve a certain source domain loss on $\mathcal{P}$. For instance, if a model naturally learns rotational invariance, one would also want to use a shift detector that will not detect shifts that only comprise of rotations. Beyond the concept of harmfulness, we empirically show that learning to detect shifts using CDCs instead of domain classifiers improves shift detection performance.

C Constrained Disagreement Classifiers

C.1 Notation

We use lowercase letters to denote classifiers that predict probability vectors over N classes e.g.,

$$f(x) = [f(x)_1, f(x)_2, \ldots, f(x)_N]^\top$$

We use bold letters to denote the *hard classifier* that outputs the class where a classifier predicts the largest probability e.g.,

$$\mathbf{f}(x) = \arg\max_i \{f(x)_i \mid i \in \{1, \ldots, N\}\}$$

The classifier f represents a base model that was trained using a **labeled dataset**

$$\mathbf{P} = \{(x_1, y_1), \ldots, (x_{|\mathbf{P}|}, y_{|\mathbf{P}|})\}$$

where each sample x_i is drawn iid from a training distribution $\mathcal{P}$ over X. Each label y_i corresponds to the ground truth label for the classification problem of interest. We use $\mathbf{P}^\star$ to represent a set of data also drawn from $\mathcal{P}$ but has not been seen by f during training.

Similarly we define an **unlabeled dataset**

$$\mathbf{Q} = \{\tilde{x}_1, \ldots, \tilde{x}_{|\mathbf{Q}|}\}$$

where each sample x_i is drawn iid from a distribution $\mathcal{Q}$ over X.

C.2 Learning to Disagree via Continuous Optimization

A constrained disagreement classifier g is a classifier with the same functional form as base classifier f that is trained to disagree with f on an unlabeled dataset $\mathbf{Q}$, while **constrained** to agree with it on a labeled dataset $\mathbf{P}$.

For models learned via. continuous optimization (e.g., Neural Networks), we train CDCs using the *disagreement cross entropy* (DCE) as a smooth objective function to encourage a disagreement with a set of labels. Expressed in Equation 5, DCE corresponds to the cross entropy between the predicted probability vector $g(x)$ and the uniform distribution over all classes except the class predicted by f denoted as $\mathbf{f}(x)$.

$$\text{DCE}(g(x), \mathbf{f}(x)) = \frac{1}{1-N} \sum_{c=1}^{N} \delta_{\mathbf{f}(x) \neq c} \log\left(g(x)_c\right) \tag{5}$$

Assuming we begin with a model f that is trained using a loss function $\mathcal{L}_0$ on a labeled dataset $\mathbf{P}$ (for instance $\mathcal{L}_0$ may be the standard cross entropy loss with additional regularization), a constrained disagreement classifier g is trained using the following joint objective function:

$$\mathcal{L}_{\text{CDC}}(g; \mathbf{P}, \mathbf{Q}, \mathbf{f}) = \frac{1}{|\mathbf{P}| + |\mathbf{Q}|} \left(\sum_{(x,y) \in \mathbf{P}} \mathcal{L}_0(g(x), y) + \lambda \times \sum_{\tilde{x} \in \mathbf{Q}} \text{DCE}\left(g(x), \mathbf{f}(x)\right) \right) \tag{6}$$

where λ is a hyperparameter that controls the agreement/disagreement tradeoff.

In plain words $\mathcal{L}_{\text{CDC}}$ is equivalent to optimizing the original objective for samples from $\mathbf{P}$ but optimizing the disagreement objective for samples from $\mathbf{Q}$. In practice, if we initialize g using the base classifier f $\mathcal{L}_{\text{CDC}}$ becomes a *fine tuning* strategy that attempts to keep the decision boundary of g close to that of f under the support of $\mathcal{P}$ but as wildly different for the support of $\mathbf{Q}$ (see Figure 1 for a visual depiction of this process).

In the following sections we will discuss (1) how to choose the hyperparameter λ to make sure agreement on $\mathcal{P}$ is approximately constrained and (2) a simple generalization of DCE that works naturally with any black box model.

C.3 Choosing λ

We can choose the scalar parameter λ in Equation 6 to set learning $\mathbf{P}$ as the primary learning objective for g and only when it cannot be improved, we allow g to learn how to disagree on $\mathbf{Q}$. The reasoning is a simple counting argument. Suppose agreeing on each sample in $\mathbf{P}$ incurs a reward of 1 and disagreeing with each sample in $\mathbf{Q}$ a reward of λ. To encourage agreement on $\mathbf{P}$ as the primary objective, we set λ such that the extra reward obtained by going from *zero* to *all* disagreements on $\mathbf{Q}$ is less than that achieved with only one extra agreement on $\mathbf{P}$, this gives $\lambda|\mathbf{Q}| < 1$. Practically, we chose $\lambda = 1/(|\mathbf{Q}| + 1)$ and find that no tuning is required. However, assigning $\lambda = \alpha/|\mathbf{Q}|$ for some $\alpha < 1$ is a valid reparameterization that allows for more finetrained control over constrained disagreement.

C.4 Learning to Disagree with Black Box Learning Algorithms

When training models with arbitrary discrete or non-differentiable parameters concerning their objective (e.g., random forest), we must find a more general solution for creating CDCs. Such a solution should (1) reduce to the DCE when the model is, in fact, continuous and trained using the standard cross-entropy, and (2) reduces to label flipping when $N = 2$ (binary classification). Our simple solution is to replicate every sample in $\mathbf{Q}$ exactly $N - 1$ times and create a unique label for each from the set $\mathcal{S} \coloneqq \{1, ..., N\} \backslash \{t\}$ where t is the disagreement target. We also each a sample a weight of $1/(N - 1)$. In the case of $N = 2$, this corresponds to no replication and simply assigning the opposite label. In the case where the model learns by cross-entropy, it equals Equation 5.

Proof. We prove this statement starting with the definition of the cross entropy

$$\text{CE}(f(x), y) = \sum_{c=1}^{N} \delta_{c=y} \log(f(x)_c) \tag{7}$$

Now we consider the sum of the cross entropy for each label in $\mathcal{S}$:

$$\sum_{y \in \mathcal{S}} \text{CE}(f(x), y) = \sum_{y \in \mathcal{S}} \sum_{c=1}^{N} \delta_{c=y} \log(f(x)_c) \tag{8}$$

$$= \sum_{c=1}^{N} \delta_{c \neq t} \log(f(x)_c) \tag{9}$$

$$= (N - 1)\text{DCE}(f(x), y) \tag{10}$$

Hence when giving each sample a weight of $(N - 1)^{-1}$ we recover the exact from of DCE. □

D Two Sample Testing

D.1 Two Sample Testing Methodology

Our method to detect covariate shift, like prior work, is to perform a statistical hypothesis test between the distributions of one or low dimensional quantities derived using each element of a possibly shifted target dataset $\mathbf{Q}$ and a known in-distribution source dataset $\mathbf{P}^\star$ that has not been observed in during model development. A significant motivation for our work was that the majority of statistical hypothesis tests used by prior work are formulated in a fashion that is independent of the particular target dataset being tested (e.g., BBSD (Lipton et al., 2018) which uses a pre-trained classifier as an ansatz for a dimensionality reduction). However, with the Detectron we follow a transductive approach by building a statistical test by training classifiers to meet a carefully crafted objective (i.e., constrained disagreement) on the target data. A drawback of this approach is that low dimensional representations are, in general, not be iid; hence to perform a fair statistical test, we must run the Detectron on a known in-distribution dataset under the **same** experimental conditions (e.g., sample size, learning rate, ensemble size). For other baseline methods that do not take the transductive approach (e.g., Mahalanobis, BBSD, Ensemble), we are not limited to choosing a source dataset $\mathbf{P}^\star$ of the same size as the samples can again be assumed to be iid. In practice, for iid methods, we fix the size of $\mathbf{P}^\star$ to 1000 for CIFAR and Camelyon, and in UCI Heart Disease, we use only 120 as the dataset is significantly smaller (920 samples).

D.2 Statistical Tests Used in Methods/Baselines

We provide a summary in the context of our work on the three types of statistical tests used in our experiments. We also explain technical details on how we use each test in our experiments.

Kolmogorov–Smirnov (KS) Test. The KS test is one of the most common non-parametric univariate statistical tests. It the two sample setting given datasets $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ the test statistic is computed as the maximum difference between the empirical CDFs of X and Y. An asymptotically correct p-value can is computed using a closed-form expression of the test statistic and sample sizes n and m. An exact p-value can also be found by considering the fraction of

every possible pair of empirical CDFs that lie within the region with a maximum bounded difference; more details can be found in Hodges (1958). In practice we use the KS test implementation found in `scipy.stats.ks_2samp` which automatically computes exact p-values when $\max\{n, m\} \leq 10,000$ and otherwise defaults to the asymptotic approximation.

We use KS tests for any distributions derived from continuous scores within our methods and baselines. For the *Relative Mahalanobis Score* test (Ren et al., 2021), we compute the p-value for shift via a KS test between the Mahalanobis score for the possibly shifted target data $\mathbf{Q}$ and a source dataset of unseen in distribution samples $\mathbf{P}^\star$. In *BBSD* (Lipton et al., 2018) we compute a KS test on each dimension of the softmax output of a classifier between the source and target datasets, the final p-value is found via Bonferroni correction as is done by Rabanser et al. (2019), which simply takes the minimum p-value and divides it by the number of tests (e.g the softmax dimension). Similarly, using *Deep Ensemble (Entropy)* and *Detectron (Entropy)*, we perform a KS test directly on the distribution of entropy values computed from each sample in the source and target datasets, respectively. See Figure 4 for a full description of the Detectron entropy test.

Binomial Test. The binomial test is simple to state and has an elegant closed-form solution. We consider a binomially distributed random variable with rate q $X \sim \text{Binomial}(n, q)$ for which we observe a single sample x. Since the binomial distribution is defined as a sum of iid Bernoulli random variables with the same rate, x may equivalently be interpreted as a set of n samples of which x are 1 and $n - x$ are 0. Given a a baseline rate p we wish to determine the probability of observing an event at least as rare as $X = x$ under the null hypothesis that $p = q$, this quantity can be computed exactly using the symmetry of the binomial distribution.

$$\begin{aligned}
\mathbb{P}_{X\sim\text{Bin}(n,p)}(X \text{ is rarer then } x) &= 2 \times \mathbb{P}_{X\sim\text{Bin}(n,p)}(X \geq x) \\
&= 2\sum_{k=x}^{n} \mathbb{P}[X = k] \\
&= 2\sum_{k=x}^{n} (1-p)^{n-k} p^k \binom{n}{k} = 2\frac{B_p(x, n-x+1)}{B(x, n-x+1)}
\end{aligned}$$

Where $B_z(\alpha, \beta)$ is the incomplete Beta function and $B(\alpha, \beta)$ is the beta function. Binomial testing is used in the *Deep Ensemble (Disagreement)* baseline method where we estimate p as the disagreement rate of a deep ensemble on the set $\mathbf{P}^\star$ (i.e the number of samples in $\mathbf{P}^\star$ where the ensemble does not predict unanimously divided by the size of $\mathbf{P}^\star$) and test for distribution shift based on the result of a binomial test on the observed disagreement on $\mathcal{Q}$. Binomial testing is also used for the classifier two sample test method (CTST) (Lopez-Paz & Oquab, 2017). First a domain classifier is trained to separate source and target data then its performance is tested on a set of unseen data where the number of samples of a total of N it correctly assigns a domain label to is compared to the null distribution $\text{Bin}(N, 0.5)$ (i.e. random guessing). For implementation purposes we use `scipy.stats.binomtest`.

Permutation Test. Our ultimate goal is to detect covariate shift $\mathcal{P} \neq \mathcal{Q}$ at a bounded significance level (i.e. bounded probability of outputting $\mathcal{P} \neq \mathcal{Q}$ when in fact $\mathcal{P} = \mathcal{Q}$). To bound the significance level, we follow the simple and principled approach of the permutation test. Suppose we wish to run the Detectron to test for shift on a set $\mathbf{Q}$ from a baseline $\mathbf{P}^\star$ (each of N samples) while Detectron, or any other test, computes a p-value on some low dimensional samples derived from $\mathbf{Q}$ and $\mathbf{P}^\star$, the significance threshold on that test will not in general correspond precisely to the significance for rejecting the original null hypothesis $\mathcal{P} = \mathcal{Q}$. The permutation test allows us to reclaim statistical guarantees by first performing several tests where the null hypothesis holds (e.g., we draw $\mathbf{Q}$ from $\mathcal{P}$) and find a cutoff for the significance of a p-value that sets the false positive rate at exactly 5%.

Our experiments run the Detectron 100 times for each sample size on random sets $\mathbf{Q}$ drawn from $\mathcal{P}$. Based on these runs we compute 95^{th} percentile (τ) on the final rejection rate. We then run the actual test using a set of samples $\mathbf{Q}$ drawn from $\mathcal{Q}$ which we deem significant at the 5% level if the number of rejected samples is greater than τ. A visual description of this method can be found in *Figure 3*.

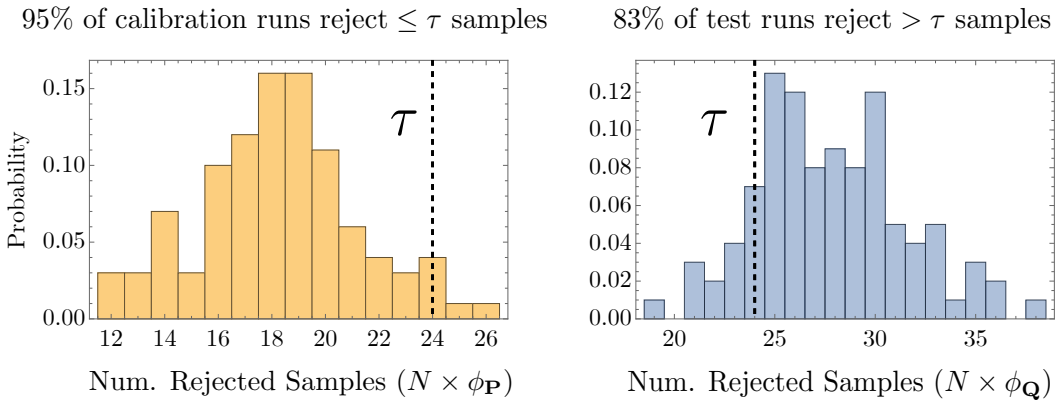

Figure 3: **The Detectron disagreement test:** In this example (taken from our experiment where $\mathcal{P}$ = CIFAR10 and $\mathcal{Q}$ = CIFAR10.1 and sample size $N = 50$) pictured we start by training an ensemble of CDCs (we use and ensemble size of 5) to reject/disagree on a set of N unseen samples from the original training distribution ($\mathbf{P}^\star$) while constrained to perform consistently with a base model on the original training and validation sets used to train the base model on CIFAR10. We perform 100 of these calibration runs using different random seeds and samples for $\mathbf{P}^\star$ to estimate a threshold τ such that 95% of the runs reject fewer then τ samples — thereby fixing the significance level of the test to 5%. To estimate the test power, we train CDCs using **the exact same configuration** as the calibration runs except we replace $\mathbf{P}^\star$ with a random set of N samples $\mathbf{Q}$ from $\mathcal{Q}$ (CIFAR 10.1). By averaging the number of runs the reject more than τ samples we can compute the power (or true positive rate) of the test for the configuration.

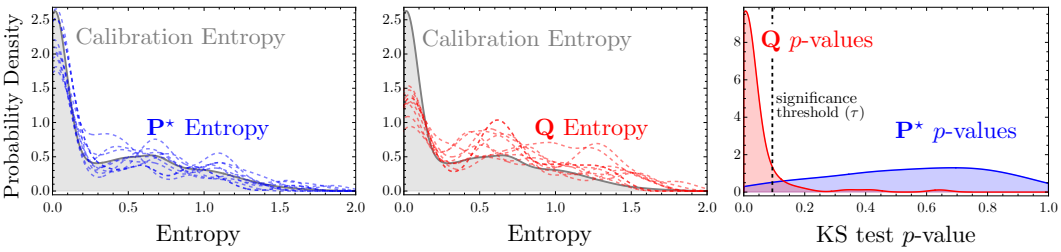

Figure 4: **The Detectron entropy test:** Following the same experimental setup as Figure 3, we start (left) by computing a KS test between the continuous entropy values for each calibration run $\mathbf{P}^\star$ with the flattened set of entropy values from all other 99 calibration runs. Then (center) we compute a KS test from each test run $\mathbf{Q}$ with a random set of all but one calibration runs. Finally (right), we find a threshold τ on the distribution of p-values obtained from step 1 as the α quantile to guarantee a false positive rate of α. The power of the test is computed as the fraction of p-values computed from 100 test runs $\mathbf{Q}$ that are below τ.

E Datasets

E.1 Sources and Licensees

- CIFAR-10 (MIT License Copyright (c) 2013 Valay Shah)
- Camelyon-17 (CC0 1.0 Universal Public Domain Dedication)
- UCI Heart Disease (Creative Commons Attribution 4.0 International)

E.2 Prepossessing, Shift Descriptions and Model Performance

We provide full details on the three datasets used in our experiments, including any preprocessing steps and what splits we considered as source domain $\mathcal{P}$ and target domain $\mathcal{Q}$.

CIFAR 10/10.1. We use the well known CIFAR 10 dataset (Krizhevsky et al., 2014) as the source domain for training base classifiers (subsection F.1) and the new CIFAR 10 test set CIFAR 10.1

containing 2000 class balanced images (Recht et al., 2019) as a source of harmful distribution shift. Although the images in CIFAR 10.1 appear to be visually very similar to CIFAR 10 most classifiers trained on CIFAR 10 drop significantly in performance (3% to 15% (Recht et al., 2019)) when tested on CIFAR 10.1.

Figure 5: Cifar 10 vs Cifar 10.1. (Image borrowed from the technical report "Do CIFAR-10 Classifiers Generalize to CIFAR-10?" (Recht et al., 2018))

Camelyon 17. As described by the original authors (Veeling et al., 2018) the Camelyon benchmark is a new and challenging image classification dataset consisting of 327,680 color images (96×96) extracted from histopathologic scans of lymph node sections. Each image is annotated with a binary label indicating the presence of metastatic tissue in the center 32×32 pixel region. In our experiments, we use the WILDS (Koh et al., 2021) framework to facilitate download, preprocessing, as well as source/target splits for Camelyon. As shown in Figure 6, the source domain is chosen as data from hospitals $1, 2, 3$. In contrast, the test domain is collected from hospital 5, which visually shows significantly higher contrast due to different data acquisition equipment/methods.

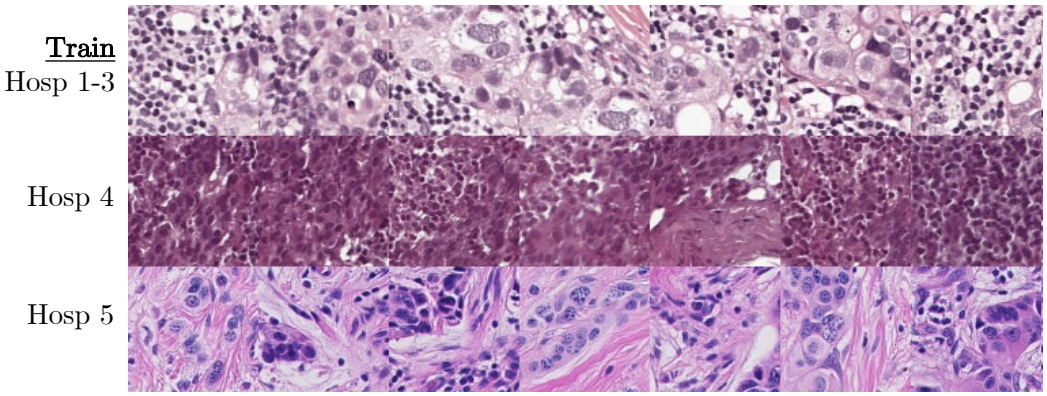

Figure 6: Samples images from the Camelyon 17 dataset (Veeling et al., 2018). Using the standard set by the WILDS framework (Koh et al., 2021) we use hospitals 1-3 as the source domain for training and validating models, and hospital 5 as the target domain for assessing distribution shift.

UCI Heart Disease. The UCI Heart Disease (UCI-HD) dataset (Janosi et al., 1988) consists of 76 attributes collected from four unique patient databases in Cleveland, Hungary, Switzerland, and the VA Long Beach. We select nine features out of the commonly used 14 to minimize the portion of missing values. These features are {age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise-induced angina}. The prediction task is to determine the diagnosis of heart disease (also known as angiographic disease status), which is given in a range from 0-4, where 0 indicates healthy

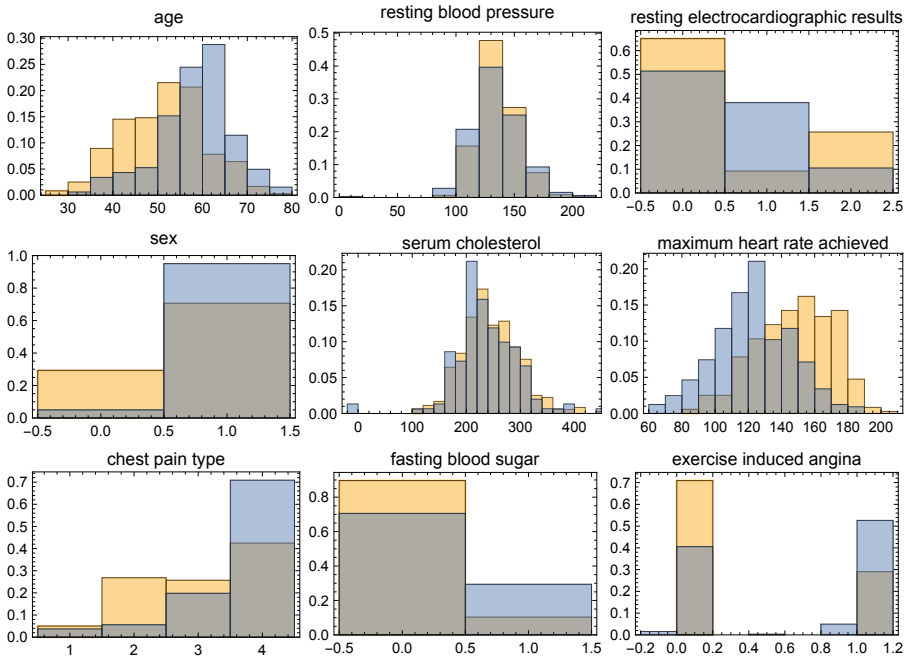

Figure 7: Marginal distributions for each of the nine variables chosen from the UCI Heart Disease dataset for our experiments. The source domain (yellow) is chosen from the Cleveland and Hungary patient databases, while the shifted target domain (blue) is selected from the Swiss and Long Beach databases. Although the distributions are visually similar, a simple neural network classifier that achieves an AUC of 0.85 on the source domain drops to 0.42 on the target domain.

and 1-4 indicates a severity level based on the narrowing of major blood vessels. Following prior work (Chaki et al., 2015) we only consider the simplified binary classification task for differentiating patients with a normal angiographic status (label of 0) from those with abnormal status (label > 0). We select the source domain as the Cleveland and Hungary databases and the target domain as the Switzerland and VA Long Beach databases. A graphical overview of the marginal feature distributions for the source and target domain is shown in Figure 7.

To allow for out-of-the-box training of deep neural networks on the UCI HD dataset, we use the missing value synthesis functional in Wolfram Mathematica (Inc.). The algorithm uses density estimation and mode finding on conditioned distributions to synthesize missing values. See the language guide page titled Synthesize Missing Values in Numeric Data for a more detailed description. To aid in future research, we provide a copy of our processed dataset in `<our github repo>/data/uci_heart.pt`.

A summary of the three datasets used as well as the description of shifts and effects on model performance is provided in Table 2.

F Experimental Details

F.1 Base Classifiers

For each dataset used in our experiments, we begin by training a *base classifier* on the source domain portion of the dataset to use in subsequent experiments and baselines. For a brief description of the datasets used and base classifiers, see Table 2 and for a more detailed description of each dataset as well as what we have considered precisely as the source and shifted domains, see subsection E.2.

CIFAR 10. We use a standard Resnet18 model pre-trained on ImageNet (Deng et al., 2009) made available in the torchvision library (Marcel & Rodriguez, 2010) (`torchvision.models.resnet18(pretrained=True)`) although we reinitialize the last network layer to have an output size of 10. We use stochastic gradient descent (SGD) with a base learning rate

Table 2: **Datasets:** We investigate three different forms of covariate shift. To verify that these shifts are indeed harmful to the models, we report performance in both the shifted and unshifted domains. Examples and further descriptions of unshifted/shifted splits of each dataset are given in Appendix E

Domain/Task	Dataset	Shift	Metric	(Unshifted)	(Shifted)
Natural Images *Object classification*	CIFAR-10/10.1 (Recht et al., 2019)	Data Collection Process	Accuracy	0.87 (Resent18)	0.77 (Resent18)
Histopathological Images *Metastases Detection*	Camelyon-17 (Veeling et al., 2018)	Different Hospitals	Accuracy	0.93 (Resent18)	0.81 (Resent18)
Tabular Medical Data *Angiographic Status*	UCI Heart Disease (Janosi et al., 1988)	Different Countries	AUROC	0.88 (xgboost) 0.85 (MLP)	0.70 (xgboost) 0.42 (MLP)

of 0.1, L_2 regularization of 5×10^{-4}, momentum of 0.9, a batch size of 128 and a cosine annealing learning rate schedule with a maximum 200 iterations stepped once per epoch for a total of 200 epochs. We use the standard CIFAR-10 training split normalized by its mean ($\mu = [0.4914, 0.4822, 0.4465]$) and standard deviation ($\sigma = [0.2023, 0.1994, 0.2010]$). Every epoch, we randomly crop each image to a size of 32×32 after applying a 0 padding of four pixels to each spatial dimension, and we apply a horizontal flip with probability 0.5. This model achieves a test performance of 87%. While this score is far from state-of-the-art on CIFAR-10, our goal is not to construct a perfect model. We wish to create a *reasonably good* model as an example of a model that could realistically be deployed in real-world settings. When training deep ensembles, we only vary the random seed in the range $[0, \ldots, 4]$.

Camelyon 17. We follow a similar approach to CIFAR 10. However, we use two output features (for binary classification of cancerous or benign pathology), a batch size of 512, the ADAM optimizer (Kingma & Ba, 2015) with a base learning rate of 0.001, L_2 regularization of 10^{-5} and a total of 5 training epochs for which we select the model with the best validation accuracy. This model achieves a test accuracy of 0.93. When training deep ensembles, we only vary the random seed in the range $[0, \ldots, 4]$.

UCI Heart Disease. We train both neural networks and gradient boosted trees using the XGboost library (Chen & Guestrin, 2016). For the neural network model, we use a simple MLP with an input dimension of 9, 3 hidden layers of size 16 with ReLU activation followed by a 30% dropout layer and a linear layer to 2 outputs (heart disease present or not). We use 358 samples for training and 120 for validation. We train for a maximum of 1000 epochs and select the model with the highest AUC on the validation set, performing early stopping if the validation AUC has not increased in over 100 epochs. This model achieves a test AUC computed on 119 samples of 0.85. As with CIFAR 10 and Camelyon 17, we only vary the random seed in the range $[0, \ldots, 9]$ when training deep ensembles. Note that we chose a larger ensemble size here as models are fairly cheap to train. Another important trick when using small $\mathbf{Q}$ sizes is to sample all of $\mathbf{Q}$ in each batch filling the best with a random set of samples from $\mathbf{P}$. Of procedure artificially inflates the size of $\mathbf{Q}$ so the hyperparameter λ must account for this by picking up an extra multiplicative factor equal to $(\text{batches per epoch})^{-1}$.

When training gradient boosted trees using XGboost we employ standard library parameters ($\eta = 0.1$, eval_metric=auc, max_depth= 6, subsample= 0.8, colsample_bytree=0.8, min_child_weight= 1, objective=binary:logistic, num_round= 10). This model while taking less then $5s$ to train achieves a test AUC of 0.88.

F.2 Constrained Disagreement Classifiers

We expand on the experimental details for learning constrained disagreement classifiers. When training a CDC $g_{(f,\mathbf{P},\mathbf{Q})}$ we start by creating a new dataset that combines all elements of the labeled set $\mathbf{P}$ and the unlabeled set $\mathbf{Q}$ with pseudo labels inferred by the base classifier f. We store a single bit for each sample in the combined dataset to indicate if a sample was originally drawn from $\mathbf{P}$ or $\mathbf{Q}$. When training CDCs with neural networks, we use the DCE loss (*Equation* 5) under similar semantics as the pseudo-code implementation provided above. When training discrete models, we resort to our generalized approach in subsection C.4. To reduce training time we initialize g using the

exact same architecture/weights as f and apply the exact same optimization algorithm/learning rate used to train f (see subsection F.1). For CIFAR 10, we train each CDC for a maximum of 10 epochs performing early stopping if the model drops in in-distribution validation performance by over 5%.

We enforce the early stopping criteria to help prevent CDCs from overfitting to the disagreement loss when the target dataset has not come from a harmfully shifted domain. The intuition is the following: under the null, if a target dataset $\mathbf{Q}$ comes from the same distribution as a training dataset $\mathbf{P}$, then learning to disagree with f on $\mathbf{Q}$ while constrained to agree on all of $\mathbf{P}$ can only be solved by overfitting to predict with high entropy on the specific examples in $\mathbf{Q}$, versus learning a distinct pattern that distinguishes the distributions. Forcing a model to predict with high entropy on a subset of in-distribution datapoints can only hurt its associated in-distribution generalization, a phenomenon which we can directly assess by measuring validation performance.

The details for training CDCs on Camelyon 17 are the same as those described for CIFAR 10, however due to the large training set size (302436 samples) we simply select a random subset of size $50,000$ as $\mathbf{P}$ at each epoch – a number we experimentally deemed as sufficient to achieve low in-distribution generalization error. When training CDCs on the UCI Heart Disease dataset, we use XGBoost (Chen & Guestrin, 2016) with the same hyperparameters described in subsection F.1.

For the runtime experiment presented in Figure 10 we train each CDC for only one batch, where each batch contains a set of 100 samples $\mathbf{Q}$ and is filled up to a batch size of 512 with random samples from $\mathbf{P}_{\text{train}}$. After every batch we eliminate all samples where the CDC disagrees with the base predictions. We continue this for a maximum of 150 batches, but perform early stopping if 10 batches pass without at least one sample getting disagreed on.

F.3 Description of Baseline Methods

We compare the Detectron against several methods for OOD detection, uncertainty estimation and covariate shift detection found in recent literature.

1. *Deep Ensembles* shown by Ovadia et al. (2019) to provide the most accurate estimates of predictive confidence under covariate shift. To compare directly with Detectron we test both the disagreements rates and the entropy distributions of the ensemble. See Appendix D for more information on how these tests are run.
2. *Black Box Shift Detection (BBSD)* (Lipton et al., 2018) is overall best method across numerous synthetic benchmarks for covariate shift detection evaluated by Rabanser et al.. We follow the same evaluation and perform a univariate KS test on each dimension of the softmax output of the base classifier between $\mathbf{Q}$ and a held out set from the training distribution. Bonferroni correction is used to compute a single p-value as the minimum value divided by the number of tests. We guarantee significance using the same permutation approach described in Appendix D.
3. *Relative Mahalanobis Distance (RMD)* (Ren et al., 2021) (a method designed specifically for identifying near OOD samples) using the penultimate layer of a pretrained model. We test for covariate shift by performing a KS test directly on the distribution of RMD confidence scores derived on $\mathbf{Q}$ and $\mathbf{P}^{\star}$.
4. *Classifier two sample test (CTST)* (Lopez-Paz & Oquab, 2017). Using the same architecture as and initialization as the base classifier we reconfigure the output layer and we train a domain classifier on half the test data with source data labeled as 0 and test data as 1. We then test this models accuracy on the other half of the test data and compare its performance to random chance using a binomial test (see **??** for more details). While this method is technically sound it is not suitable for the low data regime where learning a domain classifier on half the test data is unlikely to generalize beyond random performance on the other half.
5. *Deep Kernel MMD* (Liu et al., 2020). We use the authors original source code available at `https://github.com/fengliu90/DK-for-TST` to perform the deep kernel MMD test.
6. *H-Divergence* (Zhao et al., 2022). Most similar to our approach, this work proposes a two sample test based on the output of a learning model after training on either source or target data. Specifically, the authors fit a model to both the source dataset $\mathcal{P}$, the target dataset $\mathcal{Q}$ and a uniform mixture $(\mathcal{P} + \mathcal{Q})/2$. Under the null hypothesis $\mathcal{P} = \mathcal{Q}$ the loss in each case is equal in expectation. However when $\mathcal{P} \neq \mathcal{Q}$, the generalized entropy of the mixture distribution may be be larger. In practice the authors fit three VAE (Kingma & Welling, 2014) models and compute

the test statistic $\ell((\mathcal{P}+\mathcal{Q})/2) - \min(\ell(\mathcal{Q}), \ell(\mathcal{P}))$, where ℓ is the VAE loss computed as a sum of the binary cross entropy reconstruction loss and the KL divergence regularizer. The perform 100 runs where the null hypothesis (e.g. sample $\mathbf{Q}$ from $\mathcal{P}$) and one where it does not. Significance is determined in the standard way be observing if the true test statistic exceeds the 95^{th} percentile of the test statistic distribution under the null hypothesis. Unfortunately this method, while state of the art on several benchmarks including the MNIST vs Fake MNIST two sample test, demonstrated low utility on more complex tasks with smaller sample sizes. After a discussion with the authors, we attempted to improve the results by first pretraining the VAE to produce valid samples and reconstructions under the source distribution and computing the H-Divergence statistic after finetuning. Despite this effort, we still was low statistical significance with small sample sizes likely due to the noisy nature of training VAE's in the low data regime. We use the authors original source code available here https://github.com/a7b23/H-Divergence

G Extended Discussion of Experimental Results

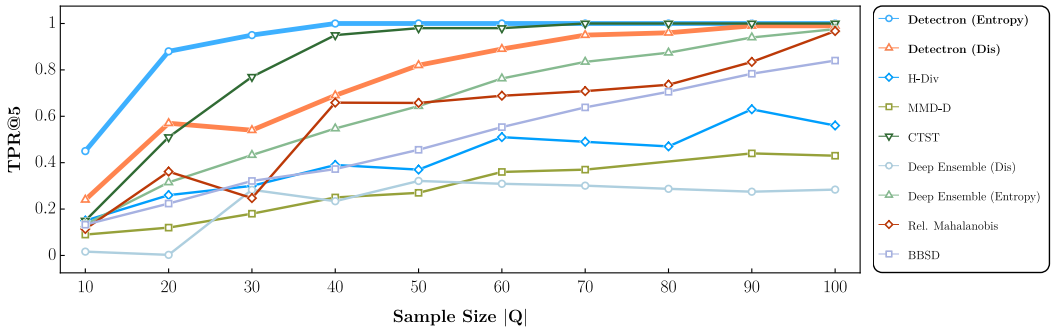

Figure 8: **True positive rate at the 5% significance level** for the Detectron and baseline methods for detection of covariate shift on the UCI heart disease dataset. The Detectron (Entropy) is shown to uniformly outperform baselines. Confidence intervals are excluded for visual clarity but are found in Table 1.

Sample Efficiency. For more significant shifts (Camelyon and UCI), we see in Table 1 the most significant improvements over baselines in the lowest sample regime (10 data points). The fine-grained result in Figure 8 shows that CTST catches up to Detectron at 40 samples while deep ensemble, BBSD, and Mahalanobis catch up at 100.

Disagreement vs Entropy. For the experiments on imaging datasets with deep neural networks Detectron (Disagreement) often performs nearly as well as Detectron (Entropy), while Detectron (Entropy) is strictly superior for the UCI dataset. While we recommend entropy as the method to maximize test power, disagreement is a more interpretable statistic as it is correlates well with the portion of misclassified samples (see Figure 9).

Comparison to baselines. Amongst the baselines, there is no clear best method. Although on average, ensemble entropy is superior on CIFAR, MMD-D on Camelyon, and CTST on UCI. Our method may be thought of as a combination of ensembles, CTST, and H-Divergence. As ensembles, we leverage the variation in outputs between a set of classifiers; as CTST, we learn in a domain adversarial setting; and as H-Divergence, we compute a test statistic based on data that a model was trained on. Lastly, while MMD-D and H-Divergence were shown to be the previous state-of-the-art, their performance was validated only on larger sample sizes (≥ 200).

On Tabular Data. The Detectron shows promise for deployment on tabular datasets (bottom right of Table 1 and Figure 8), where (1) the computational cost of training models is low, (2) the model agnostic nature of the Detectron is beneficial as random forests often outperform neural networks in tabular data (Borisov et al., 2021), and (3) based on our discussions with medical professionals, the ability to detect covariate shift from small test sizes is of particular interest in the healthcare domain where population shift is a constant problem burden for maintaining the reliability of deployed models.

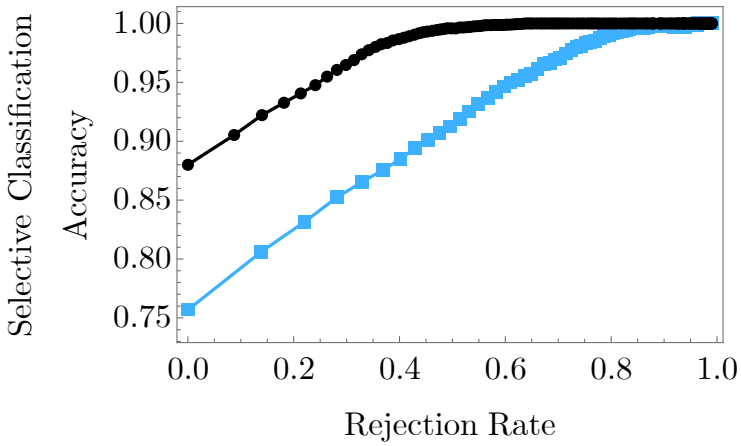

Figure 9: We examine the selective classification accuracy of the Detectron on the orignal CIFAR 10 tests et $\mathbf{P}^\star$ and CIFAR 10.1 $\mathbf{Q}$. Averaged over 100 runs with 100 samples we compute the accuracy on the held out data that the CDC has yet to disagree. We see that in distribution test accuracy reaches near 1 at a rejection rate of $\approx .5$ whereas on CIFAR 10.1 we require a rejection rate of ≈ 0.8.

H Runtime Study

Our method is more computationally expensive than some existing methods for detecting shifts such as BBSD and Mahalanobis Scores, but is similar complexity to other approaches such as Ensembles, MMD-D and H-Divergence which may require training multiple deep models. However, as the Detectron leverages a pretrained model already in deployment, we find in practice that only a small number of training rounds are required to create each CDC. For instance, on CIFAR 10/10.1 a CDC ensemble of 5 models using a ResNet 18 architecture can train in under $\approx$ 2m using an unoptimized PyTorch implementation on 1 GPU. Furthermore, looking at the runtime behavior in Figure 10 we see that while allowing for more computation time increases the fidelity of the Detectron, only a small number of training batches may be required to achieve a desirable level of statistical significance.

In scenarios where the deployed classifier is deemed high-risk (e.g. healthcare, justice system, education) where each data point is a decision that affects a human being, we believe the additional computational expense is justified for an accurate and sensitive assessment of whether the classifier needs updating. Having established the utility, accelerating the Detectron as well as building a deeper understanding of the runtime performance tradeoffs, is fertile ground for future work.

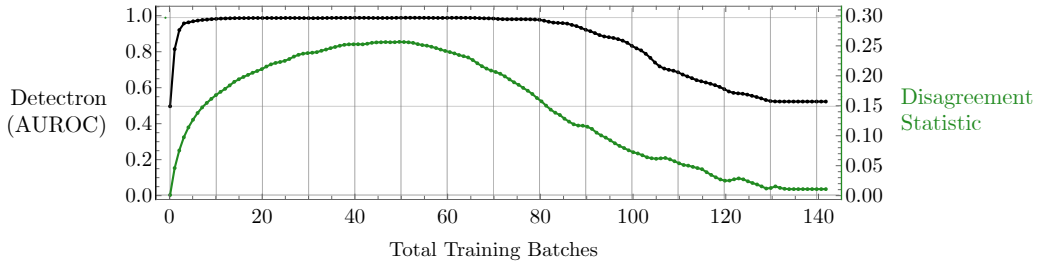

Figure 10: **Runtime Characteristics:** We train 100 random runs of CDCs on 100 samples from CIFAR 10 and 10.1 and compute the *disagreement satistic* as the difference $\psi := \mathbb{E}[\phi_{\mathbf{Q}} - \phi_{\mathbf{P}}]$. While we see that while ψ peaks near 50 training batches, only 10 batches are required for the Detectron disagreement test to reach an area under the TPR vs FPR curve (AUROC) of nearly 1 (i.e., perfect discrimination). Training CDCs for too long eventually lowers ψ as $\mathbb{E}[\phi_{\mathbf{Q}}] \approx \mathbb{E}[\phi_{\mathbf{P}}] \approx 1$ meaning CDCs eventually overfit to disagreeing on all of their data.