Leveraging Unlabeled Data to Predict Out-of-Distribution Performance

Anonymous Author(s)
Affiliation
Address
email

Abstract

Real-world machine learning deployments are characterized by mismatches between the source (training) and target (test) distributions that may cause performance drops. In this work, we investigate methods for predicting the target domain accuracy using only labeled source data and unlabeled target data. We propose Average Thresholded Confidence (ATC), a practical method that learns a *threshold* on the model's confidence, predicting accuracy as the fraction of unlabeled examples for which model confidence exceeds that threshold. ATC outperforms previous methods across several model architectures, types of distribution shifts (e.g., due to synthetic corruptions, dataset reproduction, or novel subpopulations), and datasets (WILDS-FMoW, ImageNet, BREEDS, CIFAR, and MNIST). In our experiments, ATC estimates target performance 2–$4\times$ more accurately than prior methods. We also explore the theoretical foundations of the problem, proving that, in general, identifying the accuracy is just as hard as identifying the optimal predictor and thus, the efficacy of any method rests upon (perhaps unstated) assumptions on the nature of the shift. Finally, analyzing our method on some toy distributions, we provide insights concerning when it works.

1 Introduction

Machine learning models deployed in the real world typically encounter examples from previously unseen distributions. While the IID assumption enables us to evaluate models using held-out data from the *source* distribution (from which training data is sampled), this estimate is no longer valid in presence of a distribution shift. Moreover, under such shifts, model accuracy tends to degrade [56, 49, 27]. Commonly, the only data available to the practitioner are a labeled training set (source) and unlabeled deployment-time data which makes the problem more difficult. In this setting, detecting shifts in the distribution of covariates is known to be possible (but difficult) in theory [47], and in practice [46]. However, producing an optimal predictor using only labeled source and unlabeled target data is well-known to be impossible absent further assumptions [4, 32].

Two vital questions that remain are: (i) the precise conditions under which we can estimate a classifier's target-domain accuracy; and (ii) which methods are most practically useful. To begin, the straightforward way to assess the performance of a model under distribution shift would be to collect labeled (target domain) examples and then to evaluate the model on that data. However, collecting fresh labeled data from the target distribution is prohibitively expensive and time-consuming, especially if the target distribution is non-stationary. Hence, instead of using labeled data, we aim to use unlabeled data from the target distribution, that is comparatively abundant, to predict model performance. Note that in this work, our focus is *not* to improve performance on the target but, rather, to estimate the accuracy on the target for a given classifier.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Recently, numerous methods have been proposed for this purpose [8, 5, 25, 9, 14]. These methods either require calibration on the target domain to yield consistent estimates [25, 14] or additional labeled data from several target domains to learn a linear regression function on a distributional distance that then predicts model performance [9, 8, 14]. However, methods that require calibration on the target domain typically yield poor estimates since deep models trained and calibrated on source data are not, in general, calibrated on a (previously unseen) target domain [42]. Besides, methods that leverage labeled data from target domains rely on the fact that unseen target domains exhibit strong linear correlation with seen target domains on the underlying distance measure and, hence, can be rendered ineffective when such target domains with labeled data are unavailable (in Sec. D we demonstrate such a failure on a real-world distribution shift problem). Therefore, throughout the paper, we assume access to labeled source data and only unlabeled data from target domain(s).

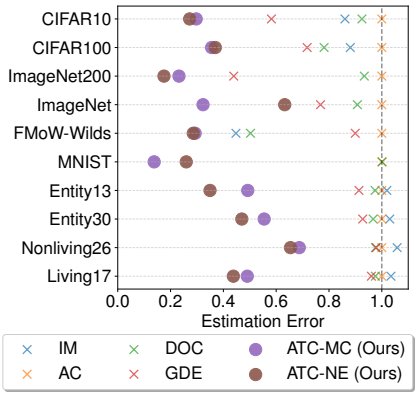


Figure 1: *Efficacy of ATC over previously proposed approaches* on our testbed with a post-hoc calibrated DenseNet121 model. To obtain errors on the same scale, we rescale all errors with Average Confidence (AC) error. Lower estimation error is better. See Table 1 for exact numbers and comparison on various types of distribution shift. See Sec. C for details on our testbed.

In this work, we first show that *absent assumptions on the source classifier or the nature of the shift, no method of estimating accuracy will work generally* (even in non-contrived settings). To estimate accuracy on target domain *perfectly*, we highlight that even given perfect knowledge of the labeled source distribution (i.e., $p_s(x, y)$) and unlabeled target distribution (i.e., $p_t(x)$), we need restrictions on the nature of the shift such that we can uniquely identify the target conditional $p_t(y|x)$. Thus, in general, identifying the accuracy of the classifier is as hard as identifying the optimal predictor.

Second, motivated by the superiority of methods that use maximum softmax probability (or logit) of a model for Out-Of-Distribution (OOD) detection [18, 19], we propose a simple method that leverages softmax probability to predict model performance. Our method, Average Thresholded Confidence (ATC), learns a threshold on a score (e.g., maximum confidence or negative entropy) of model confidence on validation source data and predicts target domain accuracy as the fraction of unlabeled target points that receive a score above that threshold. ATC selects a threshold on validation source data such that the fraction of source examples that receive the score above the threshold match the accuracy of those examples. Our primary contribution in ATC is the proposal of obtaining the threshold and showing its efficacy on (practical) accuracy estimation. Importantly, our work takes a step forward in positively answering the question raised in Deng and Zheng [8], Deng et al. [9] about a practical strategy to select a threshold that enables accuracy prediction with thresholded model confidence.

ATC is simple to implement with existing frameworks, compatible with arbitrary model classes, and dominates other contemporary methods. Across several model architectures on a range of benchmark datasets, we verify that ATC outperforms prior methods on a variety of distribution shifts including shifts due to common corruptions (e.g., ImageNet-C), natural distribution shifts due to dataset reproduction (e.g., ImageNet-v2, ImageNet-R), and shifts due to novel subpopulations (e.g., BREEDS) by at least 2–4$\times$ in predicting target accuracy.

As a starting point for theory development, in App. E, we investigate ATC on a simple toy model that models distribution shift with varying proportions of the population with spurious features, as in Nagarajan et al. [36]. Finally, we note that although ATC achieves superior performance in our empirical evaluation, like all methods, it must fail (returns inconsistent estimates) on certain types of distribution shifts, per our impossibility result.

2 Problem Setup and Impossibility Result

Suppose we have a multi-class classification problem with the input domain $\mathcal{X} \subseteq \mathbb{R}^d$ and label space $\mathcal{Y} = \{1, 2, \ldots, k\}$. For binary classification, we use $\mathcal{Y} = \{0, 1\}$. By $\mathcal{D}^{\text{S}}$ and $\mathcal{D}^{\text{T}}$, we denote source and target distribution over $\mathcal{X} \times \mathcal{Y}$. For distributions $\mathcal{D}^{\text{S}}$ and $\mathcal{D}^{\text{T}}$, we define p_{S} or p_{T} as the corresponding probability density (or mass) functions. A dataset $S := \{(x_i, y_i)\}_{i=1}^n \sim (\mathcal{D}^{\text{S}})^n$ contains n points sampled i.i.d. from $\mathcal{D}^{\text{S}}$. Let $\mathcal{F}$ be a class of hypotheses mapping $\mathcal{X}$ to Δ^{k-1} where Δ^{k-1} is a simplex in k dimensions. Given a classifier $f \in \mathcal{F}$ and datum (x, y), we denote the 0-1 error (i.e., classification error) on that point by $\mathcal{E}(f(x), y) := \mathbb{I}\left[y \notin \arg\max_{j \in \mathcal{Y}} f_j(x)\right]$. Given a model $f \in \mathcal{F}$, our goal in this work is to understand the performance of f on $\mathcal{D}^{\text{T}}$ without access to labeled data from $\mathcal{D}^{\text{T}}$. Note that our goal is not to adapt the model to the target data. Concretely, we aim to predict accuracy of f on $\mathcal{D}^{\text{T}}$. Throughout this paper, we assume we have access to the following: (i) model f; (ii) previously-unseen (validation) data from $\mathcal{D}^{\text{S}}$; and (iii) unlabeled data from target distribution $\mathcal{D}^{\text{T}}$. Throughout the paper, we assume that the classifier f is fixed.

Accuracy Estimation: Possibility and Impossibility Results First, we investigate the question of when it is possible to estimate the target accuracy of an arbitrary classifier, even given knowledge of the full source distribution $p_s(x, y)$ and target marginal $p_t(x)$. Absent assumptions on the nature of shift, estimating target accuracy is impossible. Even given access to $p_s(x, y)$ and $p_t(x)$, the problem is fundamentally unidentifiable because $p_t(y|x)$ can shift arbitrarily. In the following proposition, we show that absent assumptions on the classifier f (i.e., when f can be any classifier in the space of all classifiers on $\mathcal{X}$), we can estimate accuracy on the target data iff assumptions on the nature of the shift, together with $p_s(x, y)$ and $p_t(x)$, uniquely identify the (unknown) target conditional $p_t(y|x)$. We relegate proofs from this section to App. F.

Proposition 1. *Absent further assumptions, accuracy on the target is identifiable iff $p_t(y|x)$ is uniquely identified given $p_s(x, y)$ and $p_t(x)$.*

Proposition 1 states that we need enough constraints on nature of shift such that $p_s(x, y)$ and $p_t(x)$ identifies unique $p_t(y|x)$. It also states that under some assumptions on the nature of the shift, we can hope to estimate the model's accuracy on target data. We will illustrate this on two common assumptions made in domain adaptation literature in App. B. As a corollary of Proposition 1, we now present a simple impossibility result, demonstrating that no single method can work for all families of distribution shift.

Corollary 1. *Absent assumptions on the classifier f, no method of estimating accuracy will work in all scenarios, i.e., for different nature of distribution shifts.*

Intuitively, this result states that every method of estimating accuracy on target data is tied up with some assumption on the nature of the shift and might not be useful for estimating accuracy under a different assumption on the nature of the shift. For illustration, consider a setting where we have access to distribution $p_s(x, y)$ and $p_t(x)$. Additionally, assume that the distribution can shift only due to covariate shift or label shift without any knowledge about which one. Then Corollary 1 says that it is impossible to have a single method that will simultaneously for both label shift and covariate shift as illustrated in Example 1 in App. B.

3 Predicting accuracy with Average Thresholded Confidence

In this section, we present our method ATC that leverages a black box classifier f and (labeled) validation source data to predict accuracy on target domain given access to unlabeled target data.

Before presenting our method, we introduce some terminology. Define a score function $s : \Delta^{k-1} \to \mathbb{R}$ that takes in the softmax prediction of the function f and outputs a scalar. We want a score function such that if the score function takes a high value at a datum (x, y) then f is likely to be correct. In this work, we explore two such score functions: (i) Maximum Confidence (MC), i.e., $s(f(x)) = \max_{j \in \mathcal{Y}} f_j(x)$; and (ii) Negative Entropy (NE), i.e., $s(f(x)) = \sum_j f_j(x) \log(f_j(x))$. Our method identifies a threshold t on source data $\mathcal{D}^{\text{S}}$ such that the expected number of points that obtain a score less than t match the error of f on $\mathcal{D}^{\text{S}}$, i.e.,

$$\mathbb{E}_{x \sim \mathcal{D}^{\text{S}}}\left[\mathbb{I}\left[s(f(x)) < t\right]\right] = \mathbb{E}_{(x,y) \sim \mathcal{D}^{\text{S}}}\left[\mathbb{I}\left[\arg\max_{j \in \mathcal{Y}} f_j(x) \neq y\right]\right], \tag{1}$$

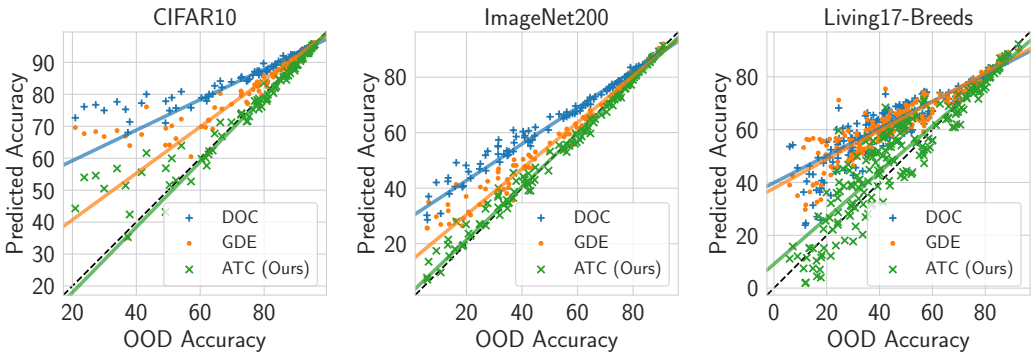


Figure 2: *Scatter plot of predicted accuracy versus (true) OOD accuracy.* Each point denotes a different OOD dataset, all evaluated with the same DenseNet121 model. We only plot the best three methods. With ATC (ours), we refer to ATC-NE. We observe that ATC significantly outperforms other methods and with ATC, we recover the desired line $y = x$ with a robust linear fit. Aggregated estimation error in Table 1 and plots for other datasets and architectures in App. M.

and then our error estimate $\text{ATC}_{D^{\text{T}}}(s)$ on the target domain $\mathcal{D}^{\text{T}}$ is given by the expected number of target points that obtain a score less than t, i.e.,

$$\text{ATC}_{\mathcal{D}^{\text{T}}}(s) = \mathbb{E}_{x \sim \mathcal{D}^{\text{T}}} \left[\mathbb{I} \left[s(f(x)) < t \right] \right] . \tag{2}$$

In short, in (1), ATC selects a threshold on the score function such that the error in the source domain matches the expected number of points that receive a score below t and in (2), ATC predicts error on the target domain as the fraction of unlabeled points that obtain a score below that threshold t. Note that, in principle, there exists a different threshold t' on the target distribution $\mathcal{D}^{\text{T}}$ such that (1) is satisfied on $\mathcal{D}^{\text{T}}$. However, in our experiments, the same threshold performs remarkably well. The main empirical contribution of our work is to show that the threshold obtained with (1) might be used effectively in conduction with modern deep networks in a wide range of settings to estimate error on the target data. In practice, to obtain the threshold with ATC, we minimize the difference between the expression on two sides of (1) using finite samples. In the next section, we show that ATC precisely predicts accuracy on the OOD data on the desired line $y = x$. In App. H, we discuss an alternate interpretation of the method and make connections with OOD detection methods.

Experiments We now empirical evaluate ATC and compare it with existing methods (refer to App. J for details on baselines). We describe the setup in App. L. In each of our main experiment, keeping the underlying model fixed, we vary target datasets and make a prediction of the target accuracy with various methods given access to only unlabeled data from the target. Details of our results in App. D.

Overall, ATC-NE typically achieves the smallest MAE improving by more than $2\times$ on CIFAR and by 3–$4\times$ on ImageNet over GDE (the next best alternative to ATC). Alongside, we also observe that robust linear fit on the scatter plot recovers a line close to $x = y$ for ATC while the line is far away from $x = y$ for other methods (Fig. 2). In Table 1 (in App. D), we report MAE results aggregated by the nature of the shift. We observe that MAE is in the range of 0.4–5.8 with ATC for CIFAR, ImageNet, MNIST, and FMoW which is significantly lower when compared with other methods.

Conclusion and Future Work In this work, we proposed ATC, a simple method for estimating target domain accuracy based on unlabeled target (and labeled source data). ATC achieves remarkably low estimation error on several synthetic and natural shift benchmarks in our empirical evaluation. Notably, our work draws inspiration from recent state-of-the-art methods that use softmax confidences below a certain threshold for OOD detection [18, 19] and takes a step forward in answering questions raised in Deng and Zheng [8], Deng et al. [9] about the practicality of threshold based methods.

Our theory establishes that a classifier's accuracy is not, in general identified, from labeled source and unlabeled target data alone, absent considerable additional constraints on the target conditional $p_t(y|x)$. In light of this finding, we also hope to extend our understanding beyond the simple theoretical model to characterize broader sets of conditions under which ATC might be guaranteed to obtain consistent estimates. Finally, we should note that while ATC outperforms previous approaches, it still suffers from huge estimation error on BREEDS with novel populations. We hope that our findings can lay the groundwork for future work for improving accuracy estimation on such datasets.

References

[1] A. Alexandari, A. Kundaje, and A. Shrikumar. Adapting to label shift with bias-corrected calibration. In *arXiv preprint arXiv:1901.06852*, 2019.

[2] K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations (ICLR)*, 2019.

[3] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in neural information processing systems*, pages 6240–6249, 2017.

[4] S. Ben-David, T. Lu, T. Luu, and D. Pál. Impossibility Theorems for Domain Adaptation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[5] M. Chen, K. Goel, N. S. Sohoni, F. Poms, K. Fatahalian, and C. Ré. Mandoline: Model evaluation under distribution shift. In *International Conference on Machine Learning*, pages 1617–1629. PMLR, 2021.

[6] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[7] C.-Y. Chuang, A. Torralba, and S. Jegelka. Estimating generalization under distribution shifts via domain-invariant representations. *arXiv preprint arXiv:2007.03511*, 2020.

[8] W. Deng and L. 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.

[9] W. Deng, S. Gould, and L. Zheng. What does rotation prediction tell us about classifier accuracy under varying testing environments? *arXiv preprint arXiv:2106.05961*, 2021.

[10] G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

[11] S. Garg, Y. Wu, S. Balakrishnan, and Z. C. Lipton. A unified view of label shift estimation. *arXiv preprint arXiv:2003.07554*, 2020.

[12] S. Garg, S. Balakrishnan, J. Z. Kolter, and Z. C. Lipton. Ratt: Leveraging unlabeled data to guarantee generalization. *arXiv preprint arXiv:2105.00303*, 2021.

[13] Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. *arXiv preprint arXiv:1705.08500*, 2017.

[14] D. Guillory, V. Shankar, S. Ebrahimi, T. Darrell, and L. Schmidt. Predicting with confidence on unseen distributions. *arXiv preprint arXiv:2107.03315*, 2021.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] J. J. Heckman. Sample Selection Bias as a Specification Error (With an Application to the Estimation of Labor Supply Functions), 1977.

[17] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[18] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[19] D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, and D. Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019.

[20] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021.

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[22] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[23] X. Ji, R. Pascanu, D. Hjelm, A. Vedaldi, B. Lakshminarayanan, and Y. Bengio. Predicting unreliable predictions by shattering a neural network. *arXiv preprint arXiv:2106.08365*, 2021.

[24] H. Jiang, B. Kim, M. Y. Guan, and M. R. Gupta. To trust or not to trust a classifier. In *NeurIPS*, pages 5546–5557, 2018.

[25] Y. Jiang, V. Nagarajan, C. Baek, and J. Z. Kolter. Assessing generalization of sgd via disagreement. *arXiv preprint arXiv:2106.13799*, 2021.

[26] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv Preprint arXiv:1412.6980*, 2014.

[27] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. A. Earnshaw, I. S. Haque, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.

[28] A. Krizhevsky and G. Hinton. Learning Multiple Layers of Features from Tiny Images. Technical report, Citeseer, 2009.

[29] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86, 1998.

[31] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[32] Z. C. Lipton, Y.-X. Wang, and A. Smola. Detecting and Correcting for Label Shift with Black Box Predictors. In *International Conference on Machine Learning (ICML)*, 2018.

[33] P. M. Long and H. Sedghi. Generalization bounds for deep convolutional neural networks. *arXiv preprint arXiv:1905.12600*, 2019.

[34] V. Nagarajan and J. Z. Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *arXiv preprint arXiv:1905.13344*, 2019.

[35] V. Nagarajan and J. Z. Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 11615–11626, 2019.

[36] V. Nagarajan, A. Andreassen, and B. Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.

[37] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[38] B. Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.

[39] B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

[40] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

[41] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2018.

[42] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, 2019.

[44] E. A. Platanios, A. Dubey, and T. Mitchell. Estimating accuracy from unlabeled data: A bayesian approach. In *International Conference on Machine Learning*, pages 1416–1425. PMLR, 2016.

[45] E. A. Platanios, H. Poon, T. M. Mitchell, and E. Horvitz. Estimating accuracy from unlabeled data: A probabilistic logic approach. *arXiv preprint arXiv:1705.07086*, 2017.

[46] S. Rabanser, S. Günnemann, and Z. C. Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. *arXiv preprint arXiv:1810.11953*, 2018.

[47] A. Ramdas, S. J. Reddi, B. Póczos, A. Singh, and L. A. Wasserman. On the Decreasing Power of Kernel and Distance Based Nonparametric Hypothesis Tests in High Dimensions. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2015.

[48] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

[49] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

[50] M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

[51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[52] M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure. *Neural Computation*, 2002.

[53] S. Santurkar, D. Tsipras, and A. Madry. Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*, 2020.

[54] H. Shimodaira. Improving Predictive Inference Under Covariate Shift by Weighting the Log-Likelihood Function. *Journal of Statistical Planning and Inference*, 2000.

[55] A. F. Siegel. Robust regression using repeated medians. *Biometrika*, 69(1):242–244, 1982.

[56] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing Properties of Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[57] R. Tachet des Combes, H. Zhao, Y.-X. Wang, and G. J. Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33, 2020.

[58] A. Torralba, R. Fergus, and W. 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.

[59] H. Wang, S. Ge, Z. Lipton, and E. P. Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019.

[60] C. Yadav and L. Bottou. Cold case: The lost mnist digits. In *Advances in Neural Information Processing Systems 32*, 2019.

[61] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[62] H. Zhang, A. Li, J. Guo, and Y. Guo. Hybrid models for open set recognition. In *European Conference on Computer Vision*, pages 102–117. Springer, 2020.

[63] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain Adaptation Under Target and Conditional Shift. In *International Conference on Machine Learning (ICML)*, 2013.

[64] W. Zhou, V. Veitch, M. Austern, R. P. Adams, and P. Orbanz. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. *arXiv preprint arXiv:1804.05862*, 2018.

Ethics Statement This paper studies methods for accuracy estimation on target data by using unlabeled data from underlying target distribution. While this could potentially guide practitioners to improve reliability in deploying machine learning models in the wild, we do not believe that it will fundamentally impact how machine learning is used in a way that could conceivably be socially salient. We take the potential impact of machine learning on society seriously amd we believe that this work, which focuses on estimating performance on out-of-distribution data, does not present any societal concern.

Reproducibility Statement We have been careful to ensure that our results are reproducible. We have stored all models and logged all hyperparameters and seeds to facilitate reproducibility. Note that throughout our work, we do not perform any hyperparameter tuning, instead, using benchmarked hyperparameters and training procedures to make our results easy to reproduce. While, we have not released code yet, the Appendix provides all the necessary details to replicate our experiments and results. Moreover, we plan to release the code with a revised version of the manuscript.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] . See reproducibility statement
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

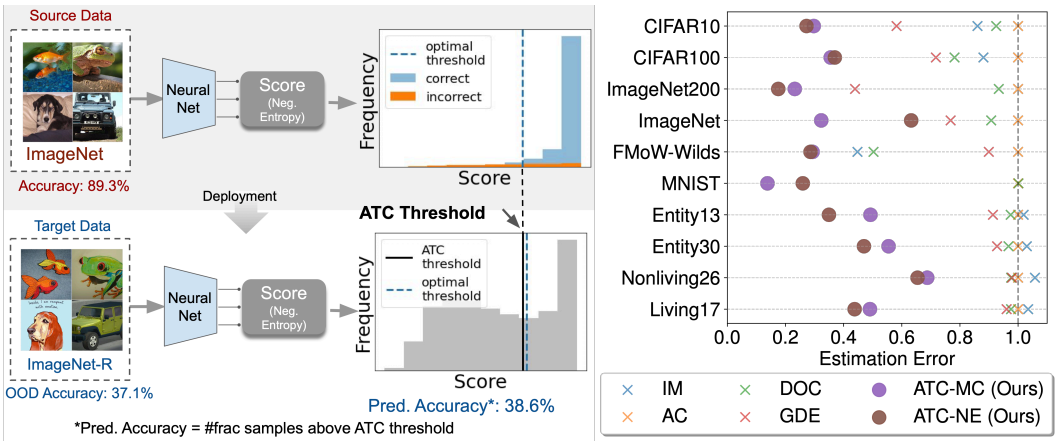


Figure 3: *Illustration of our proposed method ATC.* **Left**: using source domain validation data, we identify a *threshold* on a score (e.g. negative entropy) computed on model confidence such that fraction of examples above the threshold matches the validation set accuracy. ATC estimates accuracy on unlabeled target data as the fraction of examples with the score above the threshold. Interestingly, this threshold yields accurate estimates on a wide set of target distributions resulting from natural and synthetic shifts. **Right**: Efficacy of ATC over previously proposed approaches on our testbed with a post-hoc calibrated DenseNet121 model. To obtain errors on the same scale, we rescale all errors with Average Confidence (AC) error. Lower estimation error is better. See Table 1 for exact numbers and comparison on various types of distribution shift. See Sec. C for details on our testbed.

Appendix

A Prior Work

Out-of-distribution detection. The main goal of OOD detection is to identify previously unseen examples, i.e., samples out of the support of training distribution. To accomplish this, modern methods utilize confidence or features learned by a deep network trained on some source data. Hendrycks and Gimpel [18], Geifman and El-Yaniv [13] used the confidence score of an (already) trained deep model to identify OOD points. Lakshminarayanan et al. [29] use entropy of an ensemble model to evaluate prediction uncertainty on OOD points. To improve OOD detection with model confidence, Liang et al. [31] propose to use temperature scaling and input perturbations. Jiang et al. [24] propose to use scores based on the relative distance of the predicted class to the second class. Recently, residual flow-based methods were used to obtain a density model for OOD detection [62]. Ji et al. [23] proposed a method based on subfunction error bounds to compute unreliability per sample. Refer to Ovadia et al. [42], Ji et al. [23] for an overview and comparison of methods for prediction uncertainty on OOD data.

Predicting model generalization. Understanding generalization capabilities of overparameterized models on in-distribution data using conventional machine learning tools has been a focus of a long line of work; representative research includes Neyshabur et al. [39, 40], Neyshabur [38], Neyshabur et al. [41], Dziugaite and Roy [10], Bartlett et al. [3], Zhou et al. [64], Long and Sedghi [33], Nagarajan and Kolter [34]. At a high level, this line of research bounds the generalization gap directly with complexity measures calculated on the trained model. However, these bounds typically remain numerically loose relative to the true generalization error [61, 35]. On the other hand, another line of research departs from complexity-based approaches to use unseen unlabeled data to predict in-distribution generalization [44, 45, 12, 25].

Relevant to our work are methods for predicting the error of a classifier on OOD data based on unlabeled data from the target (OOD) domain. These methods can be characterized into two broad categories: (i) Methods which explicitly predict correctness of the model on individual unlabeled points [8, 25, 9]; and (ii) Methods which directly obtain an estimate of error with unlabeled OOD data without making a point-wise prediction [5, 14, 7].

To achieve a consistent estimate of the target accuracy, Jiang et al. [25], Guillory et al. [14] require calibration on target domain. However, these methods typically yield poor estimates as deep models trained and calibrated on some source data are seldom calibrated on previously unseen domains [42]. Additionally, Deng and Zheng [8], Guillory et al. [14] derive model-based distribution statistics on unlabeled target set that correlate with the target accuracy and propose to use a subset of *labeled* target domains to learn a (linear) regression function that predicts model performance. However, there are two drawbacks with this approach: (i) the correlation of these distribution statistics can vary substantially as we consider different nature of shifts (refer to Sec. D, where we empirically demonstrate this failure); (ii) even if there exists a (hypothetical) statistic with strong correlations, obtaining labeled target domains (even simulated ones) with strong correlations would require significant *a priori* knowledge about the nature of shift that, in general, might not be available before models are deployed in the wild. Nonetheless, in our work, we only assume access to labeled data from the source domain presuming no access to labeled target domains or information about how to simulate them.

Moreover, unlike the parallel work of Deng et al. [9], we do not focus on methods that alter the training on source data to aid accuracy prediction on the target data. Chen et al. [5] propose an importance re-weighting based approach that leverages (additional) information about the axis along which distribution is shifting in form of "slicing functions". In our work, we make comparisons with importance re-weighting baseline from Chen et al. [5] as we do not have any additional information about the axis along which the distribution is shifting.

B Illustration of Proposition 1

We will illustrate this on two common assumptions made in domain adaptation: (i) covariate shift [16, 54] and (ii) label shift [52, 63, 32]. Under covariate shift assumption, that the target marginal support $\textbf{supp}(p_t(x))$ is a subset of the source marginal support $\textbf{supp}(p_s(x))$ and that the conditional distribution of labels given inputs does not change within support, i.e., $p_s(y|x) = p_t(y|x)$, which, trivially, identifies a unique target conditional $p_t(y|x)$. Under label shift, the reverse holds: the class-conditional distribution does not change ($p_s(x|y) = p_t(x|y)$) and, again, information about $p_t(x)$ uniquely determines the target conditional $p_t(y|x)$ [32, 11]. In these settings, one can estimate an arbitrary classifier's accuracy on the target domain either by using importance re-weighting with the ratio $p_t(x)/p_s(x)$ in case of covariate shift or by using importance re-weighting with the ratio $p_t(y)/p_s(y)$ in case of label shift. While importance ratios in the former case can be obtained directly when $p_t(x)$ and $p_s(x)$ are known, the importance ratios in the latter case can be obtained by using techniques from Saerens et al. [52], Lipton et al. [32], Azizzadenesheli et al. [2], Alexandari et al. [1]. In App. G,we explore accuracy estimation in the setting of these shifts and present extensions to generalized notions of label shift [57] and covariate shift [50].

Intuitively, Corollary 1 states that every method of estimating accuracy on target data is tied up with some assumption on the nature of the shift and might not be useful for estimating accuracy under a different assumption on the nature of the shift. For illustration, consider a setting where we have access to distribution $p_s(x, y)$ and $p_t(x)$. Additionally, assume that the distribution can shift only due to covariate shift or label shift without any knowledge about which one. Then Corollary 1 says that it is impossible to have a single method that will simultaneously for both label shift and covariate shift as illustrated in the following example (we spell out the details in App. F):

Example 1. Assume binary classification with $p_s(x) = \alpha \cdot \phi(\mu_1) + (1 - \alpha) \cdot \phi(\mu_2)$, $p_s(x|y = 0) = \phi(\mu_1)$, $p_s(x|y = 1) = \phi(\mu_2)$, and $p_t(x) = \beta \cdot \phi(\mu_1) + (1 - \beta) \cdot \phi(\mu_2)$ where $\phi(\mu) = \mathcal{N}(\mu, 1)$, $\alpha, \beta \in (0, 1)$, and $\alpha \neq \beta$. Error of a classifier f on target data is given by $\mathcal{E}_1 = \mathbb{E}_{(x,y)\sim p_s(x,y)} \left[\frac{p_t(x)}{p_s(x)} \mathbb{I}\left[f(x) \neq y\right] \right]$ under covariate shift and by $\mathcal{E}_2 = \mathbb{E}_{(x,y)\sim p_s(x,y)} \left[\left(\frac{\beta}{\alpha} \mathbb{I}\left[y = 0\right] + \frac{1-\beta}{1-\alpha} \mathbb{I}\left[y = 1\right] \right) \mathbb{I}\left[f(x) \neq y\right] \right]$ under label shift. In App. F, we show that $\mathcal{E}_1 \neq \mathcal{E}_2$ for all f. Thus, given access to $p_s(x, y)$, and $p_t(x)$, any method that consistently estimates error under covariate shift will give an incorrect estimate under label shift and vice-versa. The reason is that the same $p_t(x)$ and $p_s(x, y)$ can correspond to error $\mathcal{E}_1$ (under covariate shift) or error $\mathcal{E}_2$ (under label shift) and determining which scenario one faces requires further assumptions on the nature of shift.

C Experimental Setup

Unless noted otherwise, all models are trained only on samples from the source distribution with the main exception of pre-training on a different distribution. We use labeled examples from the target distribution to only obtain true error estimates.

Datasets. First, we consider synthetic shifts induced due to different visual corruptions (e.g., shot noise, motion blur etc.) under ImageNet-C [17]. Next, we consider natural shifts due to differences in the data collection process of ImageNet [51], e.g, ImageNetv2 [49]. We also consider images with artistic renditions of object classes, i.e., ImageNet-R [20] and ImageNet-Sketch [59]. Note that renditions dataset only contains a subset 200 classes from ImageNet. To include renditions dataset in our testbed, we include results on ImageNet restricted to these 200 classes (which we call ImageNet-200) along with full ImageNet.

Second, we consider BREEDS [53] to assess robustness to subpopulation shifts, in particular, to understand how accuracy estimation methods behave when novel subpopulations not observed during training are introduced. BREEDS leverages class hierarchy in ImageNet to create 4 datasets ENTITY-13, ENTITY-30, LIVING-17, NON-LIVING-26. We focus on natural and synthetic shifts as in ImageNet on same and different subpopulations in BREEDs. Third, from WILDS benchmark, we consider FMoW-WILDS [27, 6] to consider distribution shifts faced in the wild.

Finally, similar to ImageNet, we consider (i) synthetic shifts (CIFAR-10-C) due to common corruptions; and (ii) natural shift (i.e., CIFARv2 [48]) on CIFAR-10 [28]. On CIFAR-100, we just have synthetic shifts due to common corruptions. For completeness, we also consider natural shifts on MNIST [30] as in the prior work [8]. We use three real shifted datasets, i.e., USPS [22], SVHN [37] and QMNIST [60]. We give a detailed overview of our setup in App. K.

Architectures and Evaluation. For ImageNet, BREEDS, CIFAR, and FMoW-WILDS datasets, we use DenseNet121 [21] and ResNet50 [15] architectures. For MNIST, we train a fully connected multilayer perceptron. We use standard training with benchmarked hyperparameters. To compare methods, we report average absolute difference between the true accuracy on the target data and the estimated accuracy on the same unlabeled examples. We refer to this metric as Mean Absolute estimation Error (MAE). Along with MAE, we also show scatter plots to visualize performance at individual target sets. Refer to App. L for additional details on the setup.

D Main Results

In Table 1, we report MAE results aggregated by the nature of the shift in our testbed. In Fig. 2 and Fig. 3(right), we show scatter plots for predicted accuracy versus OOD accuracy on several datasets. We include scatter plots for all datasets and parallel results with other architectures in App. M. In App. M.1, we also perform ablations on CIFAR using a pre-trained model and observe that pre-training doesn't change the efficacy of ATC.

We predict accuracy on the target data before and after calibration with Temperature Scaling (TS). First, we observe that both ATC-NE and ATC-MC (even without TS) obtain significantly lower MAE when compared with other methods (even with TS). Note that with TS we observe substantial improvements in MAE for all methods.

Overall, ATC-NE (with TS) typically achieves the smallest MAE improving by more than $2\times$ on CIFAR and by 3–$4\times$ on ImageNet over GDE (the next best alternative to ATC). Alongside, we also observe that robust linear fit on the scatter plot recovers a line close to $x = y$ for ATC-NE with TS while the line is far away from $x = y$ for other methods (Fig. 2 and Fig. 3(right)). Remarkably, MAE is in the range of 0.4–5.8 with ATC for CIFAR, ImageNet, MNIST, and FMoW. However, MAE is much higher on BREEDS benchmark with novel subpopulations. While we observe a small MAE (i.e., comparable to our observations on other datasets) on BREEDS with natural and synthetic shifts from the same sub-population, MAE on shifts with novel population is significantly higher with all methods. Note that even on novel populations, ATC continues to dominate all other methods across all datasets in BREEDS.

For different subpopulations in BREEDS setup, we observe a poor linear correlation of the estimated performance with the actual performance as shown in Fig. 4 (left)(we notice a similar gap in the linear fit for all other methods). Hence in such a setting, we would expect methods that fine-tune a

Dataset	Shift	IM		AC		DOC		GDE	ATC-MC (Ours)		ATC-NE (Ours)	
		Pre T	Post T	Pre T	Post T	Pre T	Post T	Post T	Pre T	Post T	Pre T	Post T
CIFAR10	Natural	5.60	5.05	9.89	6.97	7.24	6.07	4.73	3.11	2.79	2.80	**2.35**
	Synthetic	11.12	10.31	16.42	11.97	13.78	11.07	6.96	4.17	3.55	3.53	**3.25**
CIFAR100	Synthetic	12.10	11.50	24.14	13.05	14.82	10.20	9.36	5.30	**4.64**	**4.59**	4.82
ImageNet200	Natural	10.52	9.92	21.96	8.19	14.99	7.31	5.20	3.91	**1.67**	3.35	**1.51**
	Synthetic	19.04	16.58	32.41	13.57	25.09	12.69	5.88	5.85	3.16	5.05	**2.37**
ImageNet	Natural	3.88	2.81	12.61	2.42	3.08	2.17	4.32	1.44	0.93	0.80	**0.43**
	Synthetic	12.15	9.93	24.15	8.22	13.06	7.46	6.24	2.94	**2.65**	**2.71**	5.24
FMoW-WILDS	Natural	4.09	2.37	32.97	5.31	3.89	2.67	4.77	2.18	**1.55**	2.20	**1.52**
MNIST	Natural	27.51	22.99	21.08	12.81	20.11	12.82	23.76	5.36	**1.77**	3.47	3.32
ENTITY-13	Same	17.36	12.07	25.57	11.80	19.83	11.37	10.71	6.68	4.59	5.77	**3.96**
	Novel	29.79	23.04	38.88	22.65	33.14	22.21	20.76	15.40	12.35	14.21	**8.08**
ENTITY-30	Same	17.82	13.68	28.37	13.29	21.12	12.66	12.78	8.61	7.38	7.73	**7.95**
	Novel	31.67	26.17	43.85	25.39	36.58	24.75	23.11	17.60	14.07	15.57	**10.24**
NONLIVING-26	Same	19.37	17.82	27.14	16.79	20.38	16.22	16.70	12.27	**11.28**	**11.10**	11.52
	Novel	34.33	32.32	42.57	30.59	35.79	30.02	29.69	23.14	21.29	20.28	**19.48**
LIVING-17	Same	11.24	9.52	16.77	9.04	12.93	8.68	9.31	3.89	**3.51**	**3.81**	4.63
	Novel	27.86	25.57	34.31	24.87	30.41	24.41	23.27	14.97	13.12	11.57	**10.22**

Table 1: *Mean Absolute estimation Error (MAE) results for different datasets in our setup grouped by the nature of shift.* 'Same' refers to same subpopulation shifts and 'Novel' refers novel subpopulation shifts. Post T denotes use of TS calibration on source. Across all datasets, we observe that ATC achieves superior performance (lower MAE is better). We include results on other architectures in App. M. For GDE post T and pre T estimates match since TS doesn't alter the argmax prediction.

regression model on labeled target examples from shifts with one subpopulation will perform poorly on shifts with different subpopulations. Corroborating this intuition, next, we show that even after fitting a regression model for DOC on natural and synthetic shifts with source subpopulations, ATC without regression model continues to outperform DOC with regression model on shifts with novel subpopulation.

Fitting a regression model on BREEDS with DOC. Using label target data from natural and synthetic shifts for the same subpopulation (same as source), we fit a robust linear regression model [55] to fine-tune DOC as in Guillory et al. [14]. We then evaluate the fine-tuned DOC (i.e., DOC with linear model) on natural and synthetic shifts from novel subpopulations on BREEDS benchmark. Although we observe significant improvements in the performance of fine-tuned DOC when compared with DOC (without any fine-tuning), ATC without any regression model continues to perform better (or similar) to that of fine-tuned DOC on novel subpopulations (Fig. 4 (middle)). Refer to App. M.2 for details and Table 4 for MAE on BREEDS with regression model.

E Investigating ATC on Toy Model

In this section, we propose and analyze a simple theoretical model that distills empirical phenomena from the previous section and highlights efficacy of ATC. Here, our aim is not to obtain a general model that captures complicated real distributions on high dimensional input space as the images in ImageNet. Instead to further our understanding, we focus on an *easy-to-learn* binary classification task from Nagarajan et al. [36] with linear classifiers, that is rich enough to exhibit some of the same phenomena as with deep networks on real data distributions.

Consider a easy-to-learn binary classification problem with two features $x = [x_{\text{inv}}, x_{\text{sp}}] \in \mathbb{R}^2$ where x_{inv} is fully predictive invariant feature with a margin $\gamma > 0$ and $x_{\text{sp}} \in \{-1, 1\}$ is a spurious feature (i.e., a feature that is correlated but not predictive of the true label). Conditional on y, the distribution over x_{inv} is given as follows: $x_{\text{inv}}|(y = 1) \sim U[\gamma, c]$ and $x_{\text{inv}}|(y = 0) \sim U[-c, -\gamma]$, where c is a

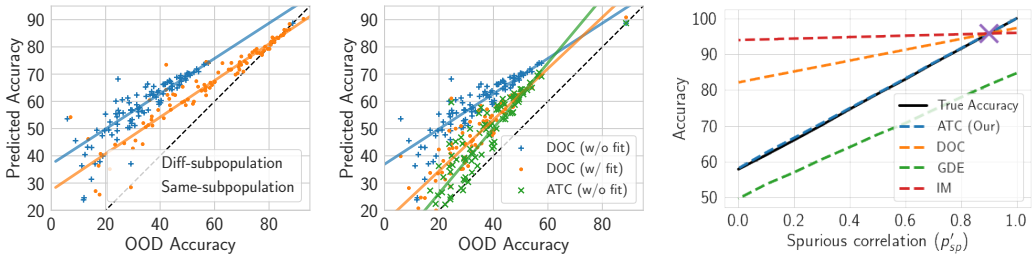


Figure 4: **Left:** Predicted accuracy with DOC on Living17 BREEDS dataset. We observe a substantial gap in the linear fit of same and different subpopulations highlighting poor correlation. **Middle:** After fitting a robust linear model for DOC on same subpopulation, we show predicted accuracy on different subpopulations with fine-tuned DOC (i.e., DOC (w/ fit)) and compare with ATC without any regression model, i.e., ATC (w/o fit). While observe substantial improvements in MAE from 24.41 with DOC (w/o fit) to 13.26 with DOC (w/ fit), ATC (w/o fit) continues to outperform even DOC (w/ fit) with MAE 10.22. We show parallel results with other BREEDS datasets in App. M.2. **Right :** Empirical validation of our toy model. We show that ATC perfectly estimates target performance as we vary the degree of spurious correlation in target. '$\times$' represents accuracy on source.

fixed constant greater than γ. For simplicity, we assume that label distribution on source is uniform on $\{-1, 1\}$. x_{sp} is distributed such that $P_s[x_{\text{sp}} \cdot (2y-1) > 0] = p_{\text{sp}}$, where $p_{\text{sp}} \in (0.5, 1.0)$ controls the degree of spurious correlation. To model distribution shift, we simulate target data with different degree of spurious correlation, i.e., in target distribution $P_t[x_{\text{sp}} \cdot (2y-1) > 0] = p'_{\text{sp}} \in [0, 1]$. Note that here we do not consider shifts in the label distribution but our result extends to arbitrary shifts in the label distribution as well.

In this setup, we examine linear sigmoid classifiers of the form $f(x) = \left[\frac{1}{1+e^{w^T x}}, \frac{e^{w^T x}}{1+e^{w^T x}}\right]$ where $w = [w_{\text{inv}}, w_{\text{sp}}] \in \mathbb{R}^2$. While there exists a linear classifier with $w = [1, 0]$ that correctly classifies all the points with a margin γ, Nagarajan et al. [36] demonstrated that a linear classifier will typically have a dependency on the spurious feature, i.e., $w_{\text{sp}} \neq 0$. They show that due to geometric skews, despite having positive dependencies on the invariant feature, a max-margin classifier trained on finite samples relies on the spurious feature. Refer to App. I for more details on these skews. In our work, we show that given a linear classifier that relies on the spurious feature and achieves a non-trivial performance on the source (i.e., $w_{\text{inv}} > 0$), ATC with maximum confidence score function *consistently* estimates the accuracy on the target distribution.

Theorem 1 (Informal)**.** *Given any classifier with $w_{inv} > 0$ in the above setting, the threshold obtained in* (1) *together with ATC as in* (2) *with maximum confidence score function obtains a consistent estimate of the target accuracy.*

Consider a classifier that depends positively on the spurious feature (i.e., $w_{\text{sp}} > 0$). Then as the spurious correlation decreases in the target data, the classifier accuracy on the target will drop and vice-versa if the spurious correlation increases on the target data. Theorem 1 shows that the threshold identified with ATC as in (1) remains invariant as the distribution shifts and hence ATC as in (2) will correctly estimate the accuracy with shifting distributions. Next, we illustrate Theorem 1 by simulating the setup empirically. First we pick a arbitrary classifier (which can also be obtained by training on source samples), tune the threshold on hold-out source examples and predict accuracy with different methods as we shift the distribution by varying the degree of spurious correlation.

Empirical validation and comparison with other methods. Fig. 4(right) shows that as the degree of spurious correlation varies, our method accurately estimates the target performance where all other methods fail to accurately estimate the target performance. Understandably, due to poor calibration of the sigmoid linear classifier AC, DOC and GDE fail. While in principle IM can perfectly estimate the accuracy on target in this case, we observe that it is highly sensitive to the number bins and choice of histogram binning (i.e., uniform mass or equal width binning). We elaborate more on this in App. I.

Biased estimation with ATC. Now we discuss changes in the above setup where ATC yields inconsistent estimates. We assumed that both in source and target $x_{\text{inv}}|y = 1$ is uniform between $[\gamma, c]$ and $x|y = -1$ is uniform between $[-c, -\gamma]$. Shifting the support of target class conditional $p_t(x_{\text{inv}}|y)$

may introduce a bias in ATC estimates, e.g., shrinking the support to $c_1(< c)$ (while maintaining uniform distribution) in the target will lead to an over-estimation of the target performance with ATC. In App. I.1, we elaborate on this failure and present a general (but less interpretable) classifier dependent distribution shift condition where ATC is guaranteed to yield consistent estimates.

Conclusion Our distribution shift toy model justifies ATC on an easy-to-learn binary classification task. In our experiments, we also observe that calibration significantly improves estimation with ATC. Since in binary classification, post hoc calibration with TS does not change the effective threshold, in future work, we hope to extend our theoretical model to multi-class classification to understand the efficacy of calibration.

F Proofs from Sec. 2

Before proving results from Sec. 2, we introduce some notations. Define $\mathcal{E}(f(x), y) := \mathbb{I}\left[y \notin \arg\max_{j\in\mathcal{Y}} f_j(x)\right]$. We express the *population error* on distribution $\mathcal{D}$ as $\mathcal{E}_{\mathcal{D}}(f) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{E}(f(x), y)\right]$.

Proof of Proposition 1. Consider a binary classification problem. Assume $\mathcal{P}$ be the set of possible target conditional distribution of labels given $p_s(x, y)$ and $p_t(x)$.

The forward direction is simple. If $\mathcal{P} = \{p_t(y|x)\}$ is singleton given $p_s(x, y)$ and $p_t(x)$, then the error of any classifier f on the target domain is identified and is given by

$$\mathcal{E}_{\mathcal{D}^T}(f) = \mathbb{E}_{x\sim p_t(x), y\sim p_t(y|x)}\left[\mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x) \neq y\right]\right] . \tag{3}$$

For the reverse direction assume that given $p_t(x)$ and $p_s(x, y)$, we have two possible distributions $\mathcal{D}^T$ and $\mathcal{D}^{T'}$ with $p_t(y|x), p'_t(y|x) \in \mathcal{P}$ such that on some x with $p_t(x) > 0$, we have $p_t(y|x) \neq p'_t(y|x)$. Consider $\mathcal{X}_M = \{x \in \mathcal{X} | p_t(x) > 0 \text{ and } p_t(y = 1|x) \neq p'_t(y = 1|x)\}$ be the set of all input covariates where the two distributions differ. We will now choose a classifier f such that the error on the two distributions differ. On a subset $\mathcal{X}^1_M = \{x \in \mathcal{X} | p_t(x) > 0 \text{ and } p_t(y = 1|x) > p'_t(y = 1|x)\}$, assume $f(x) = 0$ and on a subset $\mathcal{X}^2_M = \{x \in \mathcal{X} | p_t(x) > 0 \text{ and } p_t(y = 1|x) < p'_t(y = 1|x)\}$, assume $f(x) = 1$. We will show that the error of f on distribution with $p_t(y|x)$ is strictly greater than the error of f on distribution with $p'_t(y|x)$. Formally,

$$\begin{aligned}
&\mathcal{E}_{\mathcal{D}^T}(f) - \mathcal{E}_{\mathcal{D}^{T'}}(f) \\
&= \mathbb{E}_{x\sim p_t(x), y\sim p_t(y|x)}\left[\mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x) \neq y\right]\right] - \mathbb{E}_{x\sim p_t(x), y\sim p'_t(y|x)}\left[\mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x) \neq y\right]\right] \\
&= \int_{x\in\mathcal{X}_M} \mathbb{I}\left[f(x) \neq 0\right]\left(p_t(y = 0|x) - p'_t(y = 0|x)\right) p_t(x)dx \\
&\qquad + \int_{x\in\mathcal{X}_M} \mathbb{I}\left[f(x) \neq 1\right]\left(p_t(y = 1|x) - p'_t(y = 1|x)\right) p_t(x)dx \\
&= \int_{x\in\mathcal{X}^2_M} \left(p_t(y = 0|x) - p'_t(y = 0|x)\right) p_t(x)dx + \int_{x\in\mathcal{X}^1_M} \left(p_t(y = 1|x) - p'_t(y = 1|x)\right) p_t(x)dx \\
&> 0 ,
\end{aligned} \tag{4}$$

where the last step follows by construction of the set $\mathcal{X}^1_M$ and $\mathcal{X}^2_M$. Since $\mathcal{E}_{\mathcal{D}^T}(f) \neq \mathcal{E}_{\mathcal{D}^{T'}}(f)$, given the information of $p_t(x)$ and $p_s(x, y)$ it is impossible to distinguish the two values of the error with classifier f. Thus, we obtain a contradiction on the assumption that $p_t(y|x) \neq p'_t(y|x)$. Hence, we must pose restrictions on the nature of shift such that $\mathcal{P}$ is singleton to to identify accuracy on the target. □

Proof of Corollary 1. The corollary follows directly from Proposition 1. Since two different target conditional distribution can lead to different error estimates without assumptions on the classifier, no method can estimate two different quantities from the same given information. We illustrate this in Example 1 next. □

G Estimating accuracy in covariate shift or label shift

Accuracy estimation under covariate shift assumption Under the assumption that $p_t(y|x) = p_s(y|x)$, accuracy on the target domain can be estimated as follows:

$$\mathcal{E}_{\mathcal{D}^{\mathrm{T}}}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}^{\mathrm{S}}}\left[\frac{p_t(x,y)}{p_s(x,y)}\mathbb{I}\left[f(x)\neq y\right]\right] \tag{5}$$

$$= \mathbb{E}_{(x,y)\sim\mathcal{D}^{\mathrm{S}}}\left[\frac{p_t(x)}{p_s(x)}\mathbb{I}\left[f(x)\neq y\right]\right]. \tag{6}$$

Given access to $p_t(x)$ and $p_s(x)$, one can directly estimate the expression in (6).

Accuracy estimation under label shift assumption Under the assumption that $p_t(x|y) = p_s(x|y)$, accuracy on the target domain can be estimated as follows:

$$\mathcal{E}_{\mathcal{D}^{\mathrm{T}}}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}^{\mathrm{S}}}\left[\frac{p_t(x,y)}{p_s(x,y)}\mathbb{I}\left[f(x)\neq y\right]\right] \tag{7}$$

$$= \mathbb{E}_{(x,y)\sim\mathcal{D}^{\mathrm{S}}}\left[\frac{p_t(y)}{p_s(y)}\mathbb{I}\left[f(x)\neq y\right]\right]. \tag{8}$$

Estimating importance ratios $p_t(x)/p_s(x)$ is straightforward under covariate shift assumption when the distributions $p_t(x)$ and $p_s(x)$ are known. For label shift, one can leverage moment matching approach called BBSE [32] or likelihood minimization approach MLLS [11]. Below we discuss the objective of MLLS:

$$w = \arg\max_{w\in\mathcal{W}} \mathbb{E}_{x\sim p_t(x)}\left[\log p_s(y|x)^T w\right], \tag{9}$$

where $\mathcal{W} = \{w \mid \forall y\,, w_y \geqslant 0 \text{ and } \sum_{y=1}^{k} w_y p_s(y) = 1\}$. MLLS objective is guaranteed to obtain consistent estimates for the importance ratios $w^*(y) = p_t(y)/p_s(y)$ under the following condition.

Theorem 2 (Theorem 1 [11])**.** *If the distributions $\{p(x|y) : y = 1,\ldots,k\}$ are strictly linearly independent, then w^* is the unique maximizer of the MLLS objective* (9)*.*

We refer interested reader to Garg et al. [11] for details.

Above results of accuracy estimation under label shift and covariate shift can be extended to a generalized label shift and covariate shift settings. Assume a function $h : \mathcal{X} \to \mathcal{Z}$ such that y is independent of x given $h(x)$. In other words $h(x)$ contains all the information needed to predict label y. With help of h, we can extend estimation to following settings: (i) *Generalized covariate shift*, i.e., $p_s(y|h(x)) = p_t(y|h(x))$ and $p_s(h(x)) > 0$ for all $x \in \mathcal{X}_t$; (ii) *Generalized label shift*, i.e., $p_s(h(x)|y) = p_t(h(x)|y)$ and $p_s(y) > 0$ for all $y \in \mathcal{Y}_t$. By simply replacing x with $h(x)$ in (6) and (9), we will obtain consistent error estimates under these generalized conditions.

Proof of Example 1. Under covariate shift using (6), we get

$$\begin{aligned}
\mathcal{E}_1 &= \mathbb{E}_{(x,y)\sim p_s(x,y)}\left[\frac{p_t(x)}{p_s(x)}\mathbb{I}\left[f(x)\neq y\right]\right] \\
&= \mathbb{E}_{x\sim p_s(x,y=0)}\left[\frac{p_t(x)}{p_s(x)}\mathbb{I}\left[f(x)\neq 0\right]\right] + \mathbb{E}_{x\sim p_s(x,y=1)}\left[\frac{p_t(x)}{p_s(x)}\mathbb{I}\left[f(x)\neq 1\right]\right] \\
&= \int \mathbb{I}\left[f(x)\neq 0\right] p_t(x)p_s(y=0|x)dx + \int \mathbb{I}\left[f(x)\neq 1\right] p_t(x)p_s(y=1|x)dx
\end{aligned}$$

Under label shift using (8), we get

$$\begin{aligned}
\mathcal{E}_2 &= \mathbb{E}_{(x,y)\sim\mathcal{D}^{\mathrm{S}}}\left[\frac{p_t(y)}{p_s(y)}\mathbb{I}\left[f(x)\neq y\right]\right] \\
&= \mathbb{E}_{x\sim p_s(x,y=0)}\left[\frac{\beta}{\alpha}\mathbb{I}\left[f(x)\neq 0\right]\right] + \mathbb{E}_{x\sim p_s(x,y=1)}\left[\frac{1-\beta}{1-\alpha}\mathbb{I}\left[f(x)\neq 1\right]\right] \\
&= \int \mathbb{I}\left[f(x)\neq 0\right]\frac{\beta}{\alpha}p_s(y=0|x)p_s(x)dx + \int \mathbb{I}\left[f(x)\neq 1\right]\frac{(1-\beta)}{(1-\alpha)}p_s(y=1|x)p_s(x)dx
\end{aligned}$$

Then $\mathcal{E}_1 - \mathcal{E}_2$ is given by

$$\begin{aligned}
\mathcal{E}_1 - \mathcal{E}_2 &= \int \mathbb{I}\left[f(x) \neq 0\right] p_s(y=0|x) \left[p_t(x) - \frac{\beta}{\alpha} p_s(x)\right] dx \\
&+ \int \mathbb{I}\left[f(x) \neq 1\right] p_s(y=1|x) \left[p_t(x) - \frac{(1-\beta)}{(1-\alpha)} p_s(x)\right] dx \\
&= \int \mathbb{I}\left[f(x) \neq 0\right] p_s(y=0|x) \frac{(\alpha-\beta)}{\alpha} \phi(\mu_2) dx \\
&+ \int \mathbb{I}\left[f(x) \neq 1\right] p_s(y=1|x) \frac{(\alpha-\beta)}{1-\alpha} \phi(\mu_1) dx\,. \qquad (10)
\end{aligned}$$

If $\alpha > \beta$, then $\mathcal{E}_1 > \mathcal{E}_2$ and if $\alpha < \beta$, then $\mathcal{E}_1 < \mathcal{E}_2$. Since $\mathcal{E}_1 \neq \mathcal{E}_2$ for arbitrary f, given access to $p_s(x, y)$, and $p_t(x)$, any method that consistently estimates error under covariate shift will give an incorrect estimate under label shift and vice-versa. The reason being that the same $p_t(x)$ and $p_s(x, y)$ can correspond to error $\mathcal{E}_1$ (under covariate shift) or error $\mathcal{E}_2$ (under label shift) either of which is not discernable absent further assumptions on the nature of shift. □

H Alternate interpretation of ATC

Consider the following framework: Given a datum (x, y), define a binary classification problem of whether the model prediction $\arg\max f(x)$ was correct or incorrect. In particular, if the model prediction matches the true label, then we assign a label 1 (positive) and conversely, if the model prediction doesn't match the true label then we assign a label 0 (negative).

Our method can be interpreted as identifying examples for correct and incorrect prediction based on the value of the score function $s(f(x))$, i.e., if the score $s(f(x))$ is greater than or equal to the threshold t than our method predicts that the classifier correctly predicted datum (x, y) and vice-versa if the score is less the than t. A method that can solve this task will perfectly estimate the target performance. However, such an expectation is unrealistic. Instead, ATC expects that *most* of the examples with score above threshold correct and most of the examples below the threshold are incorrect. More importantly, ATC selects a threshold such that the number of falsely identified correct predictions match falsely identified incorrect predictions on source distribution, thereby balancing incorrect predictions.

This interpretation relates our method to the OOD detection literature where Hendrycks and Gimpel [18], Hendrycks et al. [19] highlight that classifiers tend to assign higher confidence to in-distribution examples and leverage maximum softmax confidence (or logit) to perform OOD detection.

I Details on the Toy Model

Skews observed in this toy model In Fig. 5, we illustrate the toy model used in our empirical experiment. In the same setup, we empirically observe that the margin on population with less density is large, i.e., margin is much greater than γ when the number of observed samples is small (in Fig. 5 (d)). Building on this observation, Nagarajan et al. [36] showed in cases when margin decreases with number of samples, a max margin classifier trained on finite samples is bound to depend on the spurious features in such cases. They referred to this skew as *geometric skew*.

Moreover, even when the number of samples are large so that we do not observe geometric skews, Nagarajan et al. [36] showed that training for finite number of epochs, a linear classifier will have a non zero dependency on the spurious feature. They referred to this skew as *statistical skew*. Due both of these skews, we observe that a linear classifier obtained with training for finite steps on training data with finite samples, will have a non-zero dependency on the spurious feature. We refer interested reader to Nagarajan et al. [36] for more details.

Proof of Theorem 1 Define $X_M = \{x | x_{\text{sp}} \cdot (2y-1) < 0\}$ and $X_C = \{x | x_{\text{sp}} \cdot (2y-1) > 0\}$. Notice that in target distributions, we are changing the fraction of examples in X_M and X_C but we are not changing the distribution of examples within individual set.

Theorem 3. *Given any classifier with $w_{inv} > 0$ in the above setting. Say t is the threshold obtained with* (1) *together on source distribution. Then the estimate of accuracy by ATC as in* (2) *satisfies the*

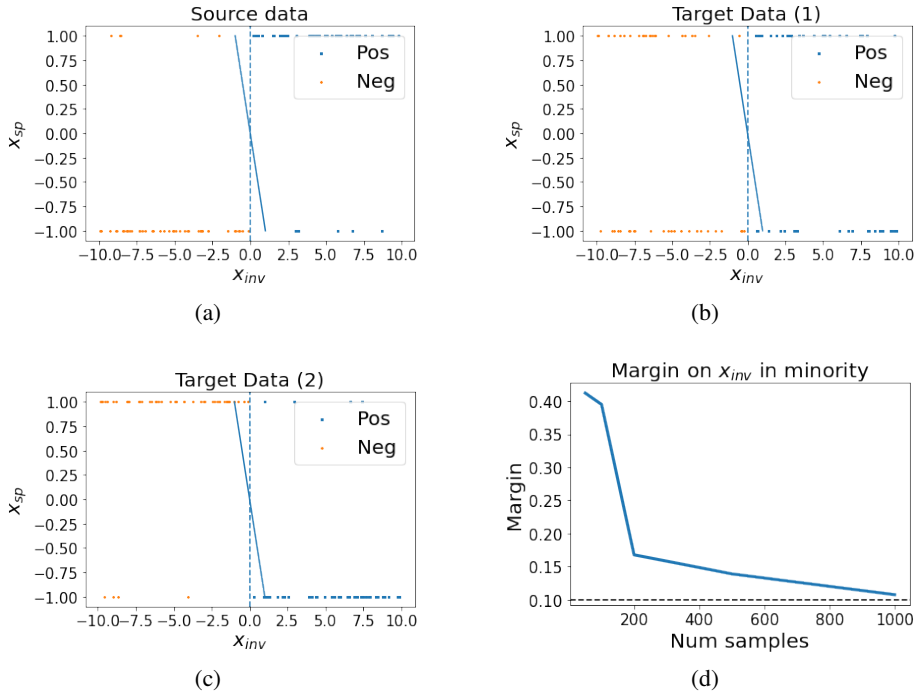


Figure 5: Illustration of toy model. (a) Source data at $n = 100$. (b) Target data with $p'_s = 0.5$. (b) Target data with $p'_s = 0.9$. (c) Margin of x_{inv} in the minority group in source data. As sample size increases the margin saturates to true margin $\gamma = 0.1$.

following:

$$ATC_{\mathcal{D}^T}(s) = \mathbb{E}_{x\sim\mathcal{D}^T}\left[\mathbb{I}\left[s(f(x)) < t\right]\right] = \mathbb{E}_{(x,y)\sim\mathcal{D}^T}\left[\mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x) \neq y\right]\right],$$

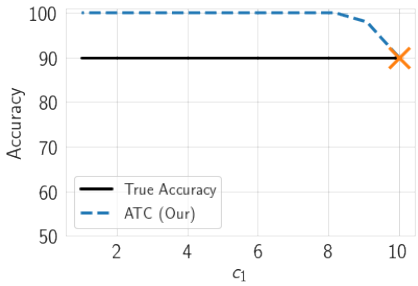
where f is the classifier parameterized by w_{inv}, w_{sp} and $\mathcal{D}^T$ is any target distribution considered in our setting.

Proof. First we consider the case of $w_{\text{sp}} > 0$. The proof follows in two simple steps. First we notice that the classifier will make an error only on some points in X_M and the threshold t will be selected such that the fraction of points in X_M with maximum confidence less than the threshold t will match the error of the classifier on X_M. Classifier with $w_{\text{sp}} > 0$ and $w_{\text{inv}} > 0$ will classify all the points in X_C correctly. Second, since the distribution of points is not changing within X_M and X_C, the same threshold continues to work for arbitrary shift in the fraction of examples in X_M, i.e., p'_{sp}.

Now for the case of $w_{\text{sp}} < 0$, we can use the same arguments on X_C. That is, since now all the error will be on points in X_C and classifier will make no error X_M, we can show that threshold t will be selected such that the fraction of points in X_C with maximum confidence less than the threshold t will match the error of the classifier on X_C. Again, since the distribution of points is not changing within X_M and X_C, the same threshold continues to work for arbitrary shift in the fraction of examples in X_M, i.e., p'_{sp}. □

Issues with IM in toy setting As described in App. J, we observe that IM is sensitive to binning strategy. In the main paper, we include IM result with uniform mass binning with 100 bins. Empirically, we observe that we recover the true performance with IM if we use equal width binning with number of bins greater than 5.

Biased estimation with ATC in our toy model We assumed that both in source and target $x_{\text{inv}}|y = 1$ is uniform between $[\gamma, c]$ and $x|y = -1$ is uniform between $[-c, -\gamma]$. Shifting the support of target class conditional $p_t(x_{\text{inv}}|y)$ may introduce a bias in ATC estimates, e.g., shrinking the support to



(a)

Figure 6: Failure of ATC in our toy model. Shifting the support of target class conditional $p_t(x_{\text{inv}}|y)$ may introduce a bias in ATC estimates, e.g., shrinking the support to $c_1(< c)$ (while maintaining uniform distribution) in the target leads to overestimation bias.

$c_1(< c)$ (while maintaining uniform distribution) in the target will lead to an over-estimation of the target performance with ATC. We show this failure in Fig. 6. The reason being that with the same threshold that we see more examples falsely identified as correct as compared to examples falsely identified as incorrect.

I.1 A More General Result

Recall, for a given threshold t, we categorize an example (x, y) as a falsely identified correct prediction (ficp) if the predicted label $\widehat{y} = \arg\max f(x)$ is not the same as y but the predicted score $f_{\widehat{y}}(x)$ is greater than t. Similarly, an example is falsely identified incorrect prediction (fiip) if the predicted label $\widehat{y}$ is the same as y but the predicted score $f_{\widehat{y}}(x)$ is less than t.

In general, we believe that our method will obtain consistent estimates in scenarios where the relative distribution of covariates doesn't change among examples that are falsely identified as incorrect and examples that are falsely identified as correct. In other words, ATC is expected to work if the distribution shift is such that falsely identified incorrect predictions match falsely identified correct prediction.

I.2 ATC produces consistent estimate on source distribution

Proposition 2. *Given labeled validation data $\{(x_i, y_i)\}_{i=1}^n$ from a distribution $\mathcal{D}^S$ and a model f, choose a threshold t as in* (1)*. Then for $\delta > 0$, with probability at least $1 - \delta$, we have*

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{I}\left[\max_{j\in\mathcal{Y}} f_j(x) < t\right] - \mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x) \neq y\right]\right] \leqslant 2\sqrt{\frac{\log(4/\delta)}{2n}} \tag{11}$$

Proof. The proof uses (i) Hoeffdings' inequality to relate the accuracy with expected accuracy; and (ii) DKW inequality to show the concentration of the estimated accuracy with our proposed method. Finally, we combine (i) and (ii) using the fact that at selected threshold t the number of false positives is equal to the number of false negatives.

Using Hoeffdings' bound, we have with probability at least $1 - \delta/2$

$$\left|\sum_{i=1}^n \left[\mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x_i) \neq y_i\right]\right] - \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x) \neq y\right]\right]\right| \leqslant \sqrt{\frac{\log(4/\delta)}{2n}}\,. \tag{12}$$

With DKW inequality, we have with probability at least $1 - \delta/2$

$$\left|\sum_{i=1}^n \left[\mathbb{I}\left[\max_{j\in\mathcal{Y}} f_j(x_i) < t'\right]\right] - \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{I}\left[\max_{j\in\mathcal{Y}} f_j(x) < t'\right]\right]\right| \leqslant \sqrt{\frac{\log(4/\delta)}{2n}}\,, \tag{13}$$

for all $t' > 0$. Finally by definition, we have

$$\sum_{i=1}^n \left[\mathbb{I}\left[\max_{j\in\mathcal{Y}} f_j(x_i) < t'\right]\right] = \sum_{i=1}^n \left[\mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x_i) \neq y_i\right]\right] \tag{14}$$

Combining (12), (13) at $t' = t$, and (14), we have the desired result. $\square$

J Basline Methods

Importance-re-weighting (IM) If we can estimate the importance-ratios $\frac{p_t(x)}{p_s(x)}$ with just the unlabeled data from the target and validation labeled data from source, then we can estimate the accuracy as on target as follows:

$$\mathcal{E}_{\mathcal{D}^{\mathrm{T}}}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}^{\mathrm{S}}}\left[\frac{p_t(x)}{p_s(x)}\mathbb{I}\left[f(x) \neq y\right]\right] . \tag{15}$$

As previously discussed, this is particularly useful in the setting of covariate shift (within support) where importance ratios estimation has been explored in the literature in the past. Mandolin [5] extends this approach. They estimate importance-weights with use of extra supervision about the axis along which the distribution is shifting.

In our work, we experiment with uniform mass binning and equal width binning with the number of bins in $[5, 10, 50, 100, 500]$. Overall, we observed that uniform mass binning works the best with 100 bins. Hence throughout this paper we perform uniform mass binning with 100 bins to include results with IM.

Average Confidence (AC) If we expect the classifier to be argmax calibrated on the target then average confidence is equal to accuracy of the classifier. Formally, by definition of argmax calibration of f on any distribution $\mathcal{D}$, we have

$$\mathcal{E}_{\mathcal{D}}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{I}\left[y \notin \arg\max_{j\in\mathcal{Y}} f_j(x)\right]\right] = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{j\in\mathcal{Y}} f_j(x)\right] . \tag{16}$$

Difference Of Confidence We estimate the error on target by subtracting difference of confidences on source and target (as a distributional distance [14]) from expected error on source distribution, i.e, $\mathrm{DOC}_{\mathcal{D}^{\mathrm{T}}} = \mathbb{E}_{x\sim\mathcal{D}^{\mathrm{S}}}\left[\mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x) \neq y\right]\right] + \mathbb{E}_{x\sim\mathcal{D}^{\mathrm{T}}}\left[\max_{j\in\mathcal{Y}} f_j(x)\right] - \mathbb{E}_{x\sim\mathcal{D}^{\mathrm{S}}}\left[\max_{j\in\mathcal{Y}} f_j(x)\right]$. This is referred to as DOC-Feat in [14].

Generalized Disagreement Equality (GDE) Jiang et al. [25] proposed average disagreement of two models (trained on the same training set but with different initialization and/or different data ordering) as a approximate measure of accuracy on the underlying data, i.e.,

$$\mathcal{E}_{\mathcal{D}}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{I}\left[f(x) \neq f'(x)\right]\right] . \tag{17}$$

They show that marginal calibration of the model is sufficient to have expected test error equal to the expected of average disagreement of two models where the latter expectation is also taken over the models used to calculate disagreement.

K Details on the Dataset Setup

In our empirical evaluation, we consider both natural and synthetic distribution shifts. We consider shifts on ImageNet [51], CIFAR Krizhevsky and Hinton [28], FMoW-WILDS [27, 6], and MNIST [30] datasets.

ImageNet setup. First, we consider synthetic shifts induced to simulate 19 different visual corruptions (e.g., shot noise, motion blur, pixelation etc.) each with 5 different intensities giving us a total of 95 datasets under ImageNet-C [17]. Next, we consider natural distribution shifts due to differences in the data collection process. In particular, we consider 3 ImageNetv2 [49] datasets each using a different strategy to collect test sets. We also evaluate performance on images with artistic renditions of object classes, i.e., ImageNet-R [20] and ImageNet-Sketch [59] with hand drawn sketch images. Note that renditions dataset only contains 200 classes from ImageNet. Hence, in the main paper we include results on ImageNet restricted to these 200 classes, which we call as ImageNet-200, and relegate results on ImageNet with 1k classes to appendix.

We also consider BREEDS benchmark [53] in our evaluation to assess robustness to subpopulation shifts, in particular, to understand how accuracy estimation methods behave when novel subpopulations not observed during training are introduced. BREEDS leverages class hierarchy in

ImageNet to repurpose original classes to be the subpopulations and defines a classification task on superclasses. Subpopulation shift is induced by directly making the subpopulations present in the training and test distributions disjoint. Overall, BREEDS benchmark contains 4 datasets ENTITY-13, ENTITY-30, LIVING-17, NON-LIVING-26, each focusing on different subtrees in the hierarchy. To generate BREEDS dataset on top of ImageNet, we use the open source library: `https://github.com/MadryLab/BREEDS-Benchmarks`. We focus on natural and synthetic shifts as in ImageNet on same and different subpopulations in BREEDs. Thus for both the subpopulation (same or novel), we obtain a total of 99 target datasets.

CIFAR setup. Similar to the ImageNet setup, we consider (i) synthetic shifts (CIFAR-10-C) due to common corruptions; and (ii) natural distribution shift (i.e., CIFARv2 [48, 58]) due to differences in data collection strategy on on CIFAR-10 [28]. On CIFAR-100, we just have synthetic shifts due to common corruptions.

*FMoW-*WILDS *setup.* In order to consider distribution shifts faced in the wild, we consider FMoW-WILDS [27, 6] from WILDS benchmark, which contains satellite images taken in different geographical regions and at different times. We obtain two different OOD target sets by considering images between years 2013–2016 and 2016–2018.

MNIST setup. For completeness, we also consider distribution shifts on MNIST [30] digit classification as in the prior work [8]. We use three real shifted datasets, i.e., USPS [22], SVHN [37] and QMNIST [60].

Train (Source)	Valid (Source)	Evaluation (Target)
MNIST (train)	MNIST (valid)	USPS, SVHN and Q-MNIST
CIFAR10 (train)	CIFAR10 (valid)	CIFAR10v2, 95 CIFAR10-C datasets (Fog and Motion blur, etc.)
CIFAR100 (train)	CIFAR100 (valid)	95 CIFAR100-C datasets (Fog and Motion blur, etc.)
FMoW (2002-12)	FMoW (2002-12)	FMoW (2013-15), FMoW (2016-17)
ImageNet (train)	ImageNet (valid)	3 ImageNetv2 datasets, ImageNet-Sketch, 95 ImageNet-C datasets
ImageNet-200 (train)	ImageNet-200 (valid)	3 ImageNet-200v2 datasets, ImageNet-R, ImageNet200-Sketch, 95 ImageNet200-C datasets
BREEDS (train)	BREEDS (valid)	Same subpopulations as train but unseen images from natural and synthetic shifts in ImageNet, Novel subpopulations on natural and synthetic shifts

Table 2: Datasets

L More Details on the Experimental Setup

All experiments were run on NVIDIA Tesla V100 GPUs. We used PyTorch [43] for experiments.

Deep nets We consider a 4-layered MLP. The PyTorch code for 4-layer MLP is as follows:

```
nn.Sequential(nn.Flatten(),
    nn.Linear(input_dim, 5000, bias=True),
    nn.ReLU(),
    nn.Linear(5000, 5000, bias=True),
    nn.ReLU(),
    nn.Linear(5000, 50, bias=True),
    nn.ReLU(),
    nn.Linear(50, num_label, bias=True)
    )
```

We mainly experiment convolutional nets. In particular, we use ResNet18 [15], ResNet50, and DenseNet121 [21] architectures with their default implementation in PyTorch. Whenever we initial our models with pre-trained models, we again use default models in PyTorch.

Hyperparameters and Training details As mentioned in the main text we do not alter the standard training procedures and hyperparameters for each task. We present results at final model, however,

we observed that the same results extend to an early stopped model as well. For completeness, we include these details below:

CIFAR10 and CIFAR100 We train DenseNet121 and ResNet18 architectures from scratch. We use SGD training with momentum of 0.9 for 300 epochs. We start with learning rate 0.1 and decay it by multiplying it with 0.1 every 100 epochs. We use a weight decay of 5^-4. We use batch size of 200. For CIFAR10, we also experiment with the same models pre-trained on ImageNet.

ImageNet For training, we use Adam with a batch size of 64 and learning rate 0.0001. Due to huge size of ImageNet, we could only train two models needed for GDE for 10 epochs. Hence, for relatively small scale experiments, we also perform experiments on ImageNet subset with 200 classes, which we call as ImageNet-200 with the same training procedure. These 200 classes are the same classes as in ImageNet-R dataset. This not only allows us to train ImageNet for 50 epochs but also allows us to use ImageNet-R in our testbed. On the both the datasets, we observe a similar superioriy with ATC. Note that all the models trained here were initialized with a pre-trained ImageNet model with the last layer replaced with random weights.

FMoW-WILDS For all experiments, we follow Koh et al. [27] and use two architectures DenseNet121 and ResNet50, both pre-trained on ImageNet. We use the Adam optimizer [26] with an initial learning rate of 10^{-4} that decays by 0.96 per epoch, and train for 50 epochs and with a batch size of 64.

Living17 and Nonliving26 from BREEDS For training, we use SGD with a batch size of 128, weight decay of 10^{-4}, and learning rate 0.1. Models were trained until convergence. Models were trained for a total of 450 epochs, with 10-fold learning rate drops every 150 epochs. Note that since we want to evaluate models for novel subpopulations no pre-training was used. We train two architectures DenseNet121 and ResNet50.

Entity13 and Entity30 from BREEDS For training, we use SGD with a batch size of 128, weight decay of 10^{-4}, and learning rate 0.1. Models were trained until convergence. Models were trained for a total of 300 epochs, with 10-fold learning rate drops every 100 epochs. Note that since we want to evaluate models for novel subpopulations no pre-training was used. We train two architectures DenseNet121 and ResNet50.

MNIST For MNIST, we train a MLP described above with SGD with momentum 0.9 and learning rate 0.01 for 50 epochs. We use weight decay of 10^{-5} and batch size as 200.

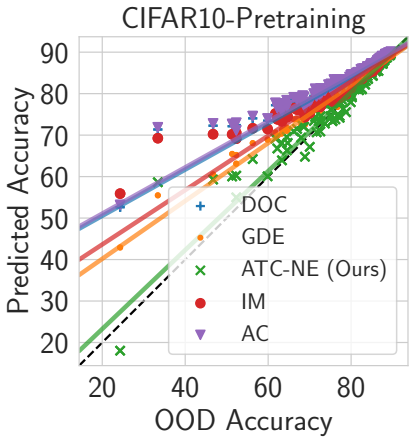
Implementation for Temperature Scaling We use temperature scaling implementation from `https://github.com/kundajelab/abstention`. We use validation set (the same we use to obtain ATC threshold or DOC source error estimate) to tune a single temperature parameter.

M Supplementary Results

Dataset	Shift	IM		AC		DOC		GDE	ATC-MC (Ours)		ATC-NE (Ours)	
		Pre T	Post T	Pre T	Post T	Pre T	Post T	Post T	Pre T	Post T	Pre T	Post T
CIFAR10	Natural	7.31	6.28	10.29	7.05	7.66	6.31	5.80	4.30	4.05	3.97	**3.48**
	Synthetic	13.36	11.21	16.83	12.10	14.20	11.37	8.09	6.01	5.12	5.01	**3.51**
CIFAR100	Synthetic	12.28	11.68	16.67	11.27	10.97	8.93	11.80	5.34	5.23	5.46	**5.65**
ImageNet200	Natural	15.84	9.58	23.46	7.55	15.98	6.69	5.46	5.62	1.98	5.18	**1.61**
	Synthetic	23.09	13.97	33.50	9.81	25.53	8.92	4.30	6.21	2.94	5.41	3.46
ImageNet	Natural	6.13	3.24	15.95	2.33	3.21	1.87	2.68	1.48	0.92	1.05	**0.61**
	Synthetic	13.64	10.98	27.81	7.56	13.64	6.18	7.29	3.00	2.29	**1.71**	4.75
FMoW-WILDS	Natural	9.62	2.92	34.11	5.08	4.59	2.96	4.56	2.09	1.83	2.00	**1.79**
ENTITY-13	Same	18.72	14.80	28.09	14.01	21.11	13.81	14.71	10.43	8.37	8.82	**8.15**
	Novel	28.88	24.14	39.01	23.14	32.01	22.94	23.44	18.04	14.76	15.76	**10.65**
ENTITY-30	Same	15.55	13.85	24.69	13.17	16.62	11.85	14.47	6.48	6.02	**5.86**	6.06
	Novel	27.14	25.10	36.67	23.74	28.57	22.39	23.37	13.59	12.00	11.03	**10.09**
NONLIVING-26	Same	16.07	14.61	26.27	14.50	16.44	13.37	15.07	**10.02**	**10.20**	10.59	11.40
	Novel	28.51	26.68	38.89	26.16	29.06	24.98	25.77	18.59	17.02	16.26	**15.74**
LIVING-17	Same	15.29	13.41	23.70	12.07	16.04	11.92	9.74	5.63	4.75	**4.63**	**4.52**
	Novel	29.81	27.50	38.67	25.76	30.93	25.61	21.19	16.17	13.80	13.40	**11.19**

Table 3: *Mean Absolute estimation Error (MAE) results for different datasets in our setup grouped by the nature of shift.* Result parallel to Table 1 with ResNet model. 'Same' refers to same subpopulation shifts and 'Novel' refers novel subpopulation shifts. Post T denotes use of TS calibration on source. Across all datasets, we observe that ATC achieves superior performance (lower MAE is better). For GDE post T and pre T estimates match since TS doesn't alter the argmax prediction.

M.1 CIFAR pretraining Ablation



(a)

Figure 7: Results with a pretrained DenseNet121 model on CIFAR10. We observe similar behaviour as that with a model trained from scratch.

M.2 BREEDS results with regression model

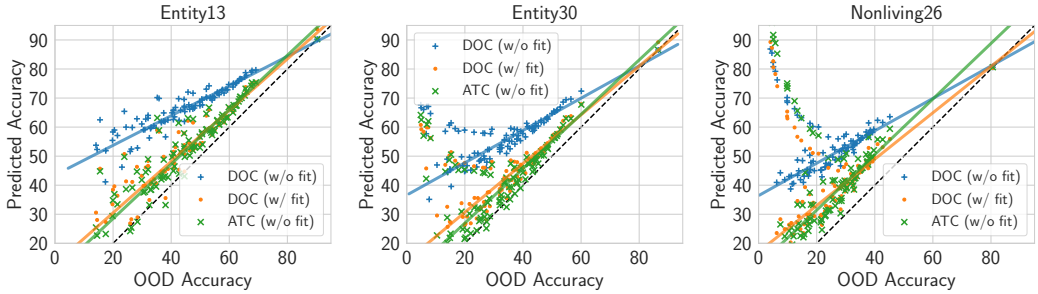


Figure 8: Scatter plots for DOC with linear fit. Results parallel to Fig. 4(Middle) on other BREEDS dataset.

Dataset	DOC (w/o fit)	DOC (w fit)	ATC-MC (Ours) (w/o fit)
LIVING-17	24.32	13.65	**10.07**
NONLIVING-26	29.91	**18.13**	19.37
ENTITY-13	22.18	8.63	8.01
ENTITY-30	24.71	12.28	**10.21**

Table 4: *Mean Absolute estimation Error (MAE) results for BREEDs datasets with novel populations in our setup.* After fitting a robust linear model for DOC on same subpopulation, we show predicted accuracy on different subpopulations with fine-tuned DOC (i.e., DOC (w/ fit)) and compare with ATC without any regression model, i.e., ATC (w/o fit). While observe substantial improvements in MAE from DOC (w/o fit) to DOC (w/ fit), ATC (w/o fit) continues to outperform even DOC (w/ fit).

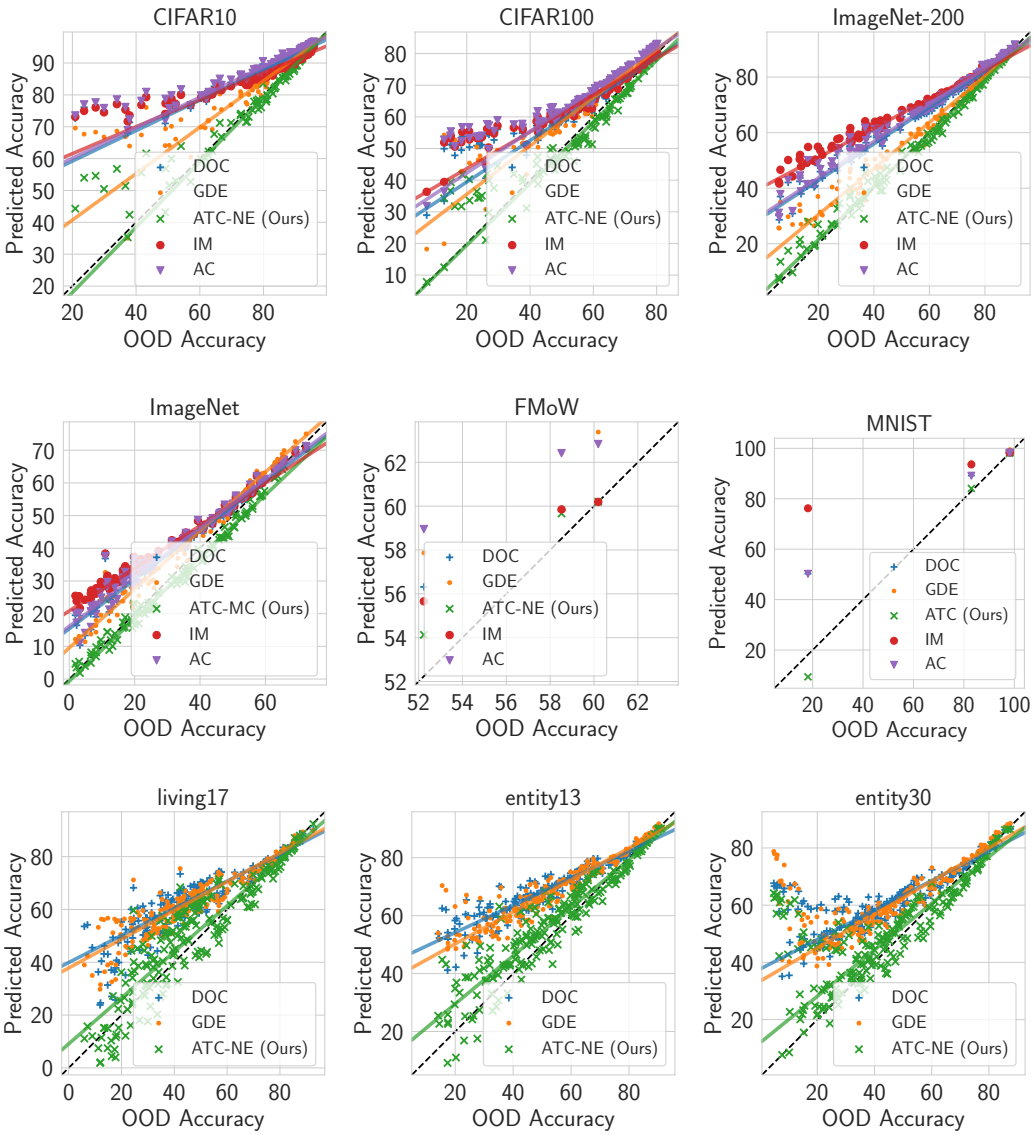


Figure 9: Scatter plot of predicted accuracy versus (true) OOD accuracy for DenseNet121 model.

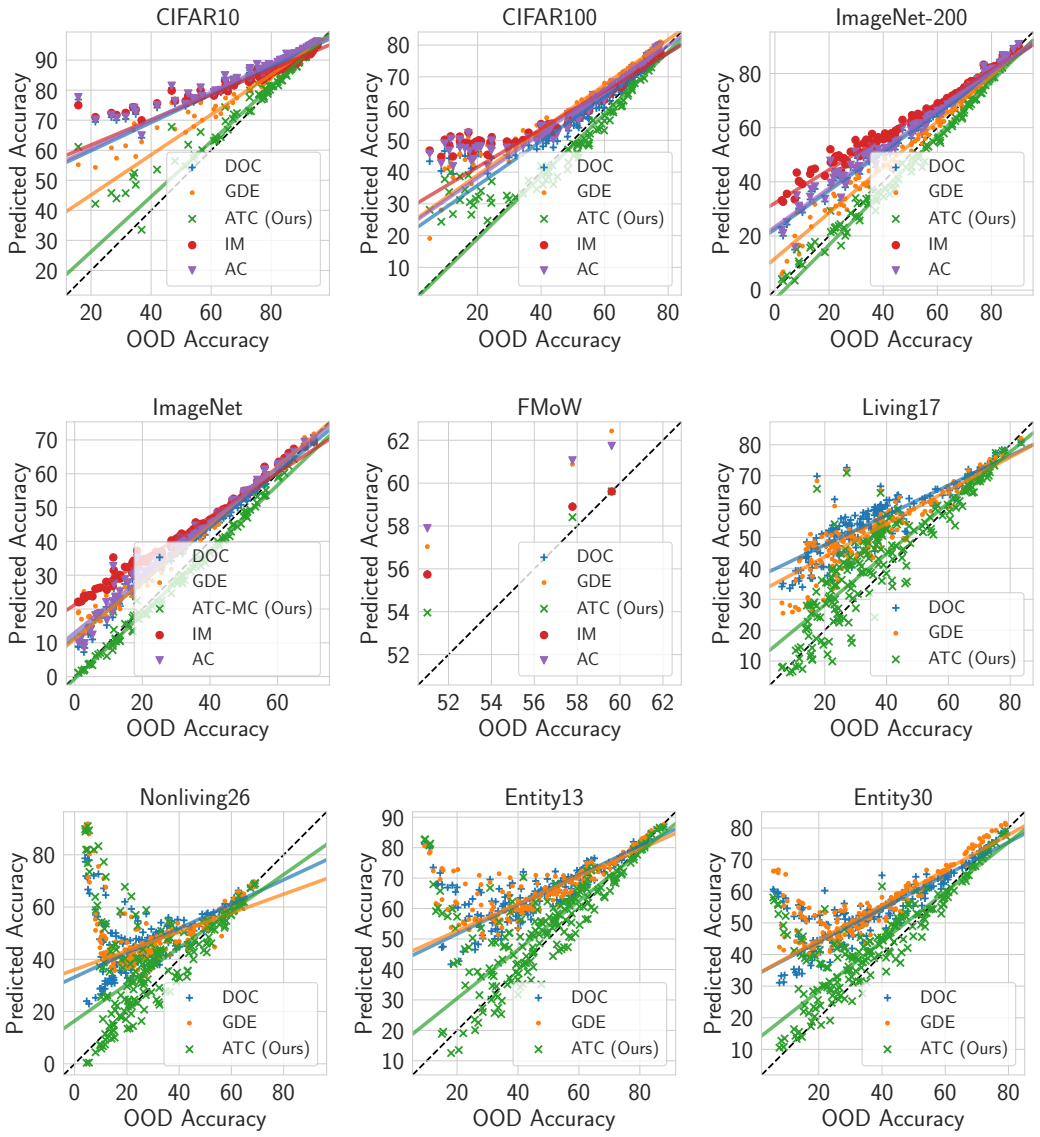


Figure 10: Scatter plot of predicted accuracy versus (true) OOD accuracy for ResNet50 model.