Understanding Threshold-based Auto-labeling: The Good, the Bad, and the Terra Incognita

Harit Vishwakarma HVISHWAKARMA@CS.WISC.EDU
University of Wisconsin-Madison

Heguang Lin HGLIN@SEAS.UPENN.EDU
University of Pennsylvania

Frederic Sala FREDSALA@CS.WISC.EDU
University of Wisconsin-Madison

Ramya Korlakai Vinayak RAMYA@ECE.WISC.EDU
University of Wisconsin-Madison

Abstract

Creating large-scale high-quality labeled datasets is a major bottleneck in supervised machine learning workflows. Threshold-based auto-labeling (TBAL), where validation data obtained from humans is used to find a confidence threshold above which the data is machine-labeled, reduces reliance on manual annotation. TBAL is emerging as a widely-used solution in practice. Given the long shelf-life and diverse usage of the resulting datasets, understanding when the data obtained by such auto-labeling systems can be relied on is crucial. This is the first work to analyze TBAL systems and derive sample complexity bounds on the amount of human-labeled validation data required for guaranteeing the quality of machine-labeled data. Our results provide two crucial insights. First, reasonable chunks of unlabeled data can be automatically and accurately labeled by seemingly bad models. Second, a hidden downside of TBAL systems is potentially prohibitive validation data usage. Together, these insights describe the promise and pitfalls of using such systems. We validate our theoretical guarantees with extensive experiments on synthetic and real datasets. [1]

1. Introduction

Machine learning (ML) models with billions of parameters are used to obtain state-of-the-art performance in many applications, e.g., object identification Redmon and Farhadi (2017), machine translation Vaswani et al. (2017), and fraud detection Zeng and Tang (2021). Such large-scale models require training on large-scale labeled datasets. As an outcome, the typical supervised ML workflow begins with the construction of a large-scale high-quality dataset. Datasets with up to millions of labeled data points, have played a pivotal role in the advancement of computer vision. However, collecting labeled data is an expensive and time consuming process. A common approach is to rely on the services of crowd-sourcing platforms such as Amazon Mechanical Turk (AMT) to get ground-truth labels.

1. A detailed version of the paper will be appearing in NeurIPS 2023.

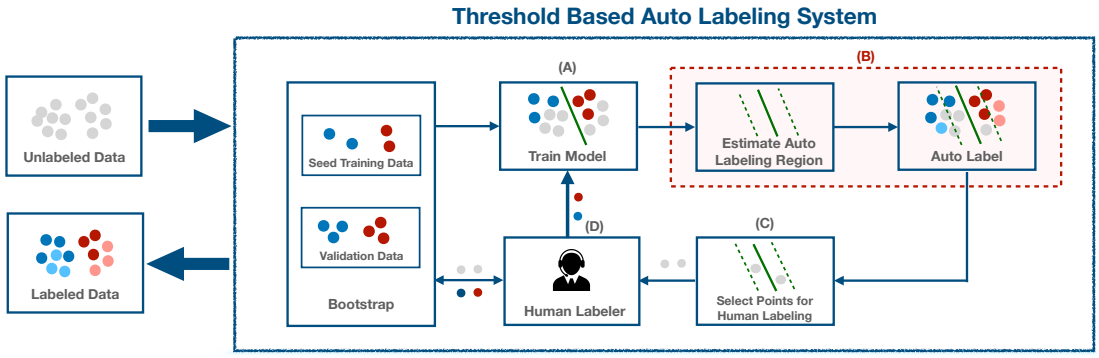


Figure 1: High-level workflow threshold-based auto-labeling (TBAL). Box (B) shows the key component estimating the auto-labeling region using validation data and autolabeling points in it.

Even with crowd-sourcing, obtaining labels for the entire dataset is expensive. To reduce costs, data labeling systems that partially rely on using a model's predictions as labels have been developed. Such systems date back to teacher-less training Fralick (1967). Modern examples include Amazon Sagemaker Ground Truth SGT (2022) and others Superb-AI (2022); Samsung-SDS (2022); Airbus (2022); Venguswamy et al. (2021). These approaches can be broadly termed *auto-labeling*.

Auto-labeling systems aim to label unlabeled data using predictions from ML models that are often trained on small amounts of human-labeled data which can produce incorrect labels. The shelf life of datasets is longer than those of models, e.g., ImageNet continues to be a benchmark for many computer vision tasks Deng et al. (2009) fifteen years after its initial development. As a result, to reliably train new models on auto-labeled datasets and deploy them, we need a thorough understanding of how reliable the datasets output by these auto-labeling systems are. Unfortunately, many widely used commercial auto-labeling systems SGT (2022); Samsung-SDS (2022) are largely opaque with limited public information on their functionality. It is therefore unclear whether the quality of the datasets obtained can be trusted. To address this, we study the high-level workflow of a popular *threshold-based* auto-labeling (TBAL) system (see Figure 1).

We emphasize that our goal is to understand such systems and their performance—not to promote them as a superior alternative to other approaches. Our goal is:

Goal: Develop a fundamental understanding of TBAL systems. This is crucial: there is a lack of theoretical understanding of reliability of these systems despite their wide adoption.

The TBAL systems we study (Figure 1) work iteratively. At high level, in each iteration, the system trains a model on currently available human-labeled data and decides to label certain parts of unlabeled data using the trained model by finding high-accuracy regions using validation data. It then collects human labels on a small portion of unlabeled data

that is deemed helpful for training the current model in the next iteration. The validation data is created by sampling i.i.d. points from the unlabeled pool and querying human labels for them. See section 7 for detailed description of the algorithm. In addition to training data, the validation data is a major driver of the cost and accuracy of auto-labeling, and will be a key component in our study.

Our Contributions: We study TBAL systems (Figure 1) and make the following contributions:

- Provide the **first theoretical characterization of TBAL systems**, developing tradeoffs between the quantity of manually-labeled data and the quantity and quality of auto-labeled data (Section 3).
- Empirical results validating our theoretical understanding on real and synthetic data (Section 9).

Our results reveal **two important insights**. Promisingly, even poor-quality models are capable of reliably labeling at least some data when we have access to sufficient validation data and a good confidence function that can accurately quantify the confidence of a given model on any data point. On the downside, in certain scenarios, the quantity of the validation data required to reach a certain quantity and quality of auto-labeled data can be high.

2. Threshold-Based Auto-Labeling

Notation: Let the instance and label spaces be $\mathcal{X}$ and $\mathcal{Y} = \{1, \ldots, k\}$. We assume that there is some *deterministic* but unknown function $f^* : \mathcal{X} \mapsto \mathcal{Y}$ that assigns true label $y = f^*(\mathbf{x})$ to any $\mathbf{x} \in \mathcal{X}$. We also assume that there is a *noiseless oracle* $\mathcal{O}$ that can provide the true label $y \in \mathcal{Y}$ for any given $\mathbf{x} \in \mathcal{X}$. Let $X_{pool} \subseteq \mathcal{X}$ denote a sufficiently large pool of unlabeled data to be labeled.

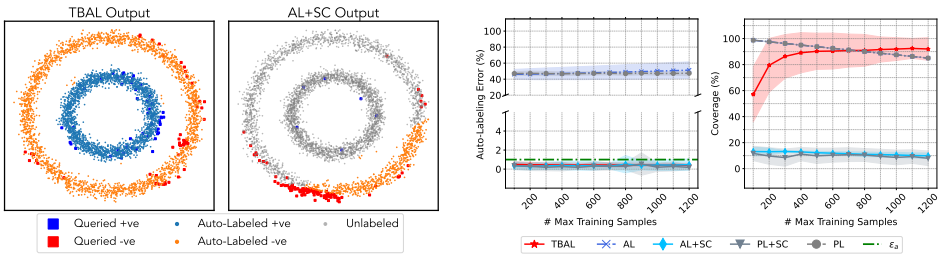
The goal of an auto-labeling algorithm is to produce accurate labels $\tilde{y}_i \in \mathcal{Y}$ for points $\mathbf{x}_i \in X_{pool}$ while minimizing the number of queries to the oracle. Let $[m] := \{1, 2, \ldots, m\}$, $A \subseteq [N]$ be the set of indices of auto-labeled points, and $X_{pool}(A)$ be these points. The *auto-labeling error* $\widehat{\mathcal{E}}(X_{pool}(A))$ and the *coverage* $\widehat{\mathcal{P}}(X_{pool}(A))$ are defined as

$$\widehat{\mathcal{E}}(X_{pool}(A)) := \frac{1}{N_a} \sum_{i \in A} \mathbb{1}(\tilde{y}_i \neq f^*(\mathbf{x}_i)) \quad \text{and} \quad \widehat{\mathcal{P}}(X_{pool}(A)) := \frac{|A|}{N} = \frac{N_a}{N}, \tag{1}$$

where N_a denotes the size of auto-labeled set A. TBAL algorithm aims to auto-label the dataset so that $\widehat{\mathcal{E}}(X_{pool}(A)) \leq \epsilon_a$ while maximizing coverage $\widehat{\mathcal{P}}(X_{pool}(A))$ for any given $\epsilon_a \in (0, 1)$.

Hypothesis Class and Confidence Function: A TBAL algorithm is given a fixed *hypothesis space* $\mathcal{H}$ and a *confidence function* $g : \mathcal{H} \times \mathcal{X} \mapsto T \subseteq \mathbb{R}^+$ that quantifies the confidence of $h \in \mathcal{H}$ on any data point $\mathbf{x} \in \mathcal{X}$. Confidence functions include prediction probabilities and margin scores. For example, when $\mathcal{H}$ is a set of unit-norm homogeneous linear classifiers, i.e. $h_{\mathbf{w}}(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x})$ with $\mathbf{w} \in \{\mathbf{w} \in \mathbb{R}^d : ||\mathbf{w}||_2 = 1\}$, a reasonable confidence function is $g(h_{\mathbf{w}}, \mathbf{x}) = |\mathbf{w}^T\mathbf{x}|$.

Note that the target f^* might not be in the hypothesis space $\mathcal{H}$. Our analysis (Section 3) shows that the TBAL algorithm can work well, i.e., accurately label a reasonable fraction of



(a) TBAL and AL+SC on Circles dataset

(b) Auto labeling performance of various methods

Figure 2: Comparison of TBAL and Active Learning followed by Selective Classification (AL+SC) on the Circles dataset (Sec. 2.1) using linear classifiers and confidence functions. (a) Samples auto-labeled, queried, and left unlabeled. (b) The auto-labeling error and coverage achieved. (50 trials.)

unlabeled data with simpler hypothesis classes that do not contain the target hypothesis f^*. We illustrate this with a simple example in Section 2.1 and Figure 2. Note as well that the features $\mathbf{x}$ could be raw features or representations from self-supervised techniques, pre-trained models etc. We analyze TBAL in settings (i) with no assumptions on the features and (ii) when the the features are linearly separable. We refer the reader to section 7 for detailed description of the algorithm.

2.1 Comparison between Auto-Labeling, Active Learning and Selective Classification

Active learning: The goal of active learning Settles (2009) (AL) is to find the best model in hypothesis class $\mathcal{H}$ by training with less labeled data compared to passive learning. This is usually achieved by obtaining labels for the most informative points. Note that the *end goal is to output a model* from the function class whose predictions on new data as good as the best model in the function class could.

Selective Classification: The goal of selective classification (SL) El-Yaniv and Wiener (2010) is to find the best combination of the hypothesis and selection functions to minimize error and maximize coverage of selection regions.

Auto-Labeling: The output of an auto-labeling procedure is a labeled dataset (not a model). When the hypothesis class is of lower complexity, it is often not possible to find a good classifier. The goal of auto-labeling is to label as much of the unlabeled data as accurately as possible with a given function class and with limited labeled data from humans.

Is active learning alone enough to auto-label data? AL has been found to be effective in reducing the amount of labels needed to learn versus passive learning, particularly in low-noise cases Hanneke (2014). Using auto-labeling using AL followed by SC may be effective in such settings. However, in real-world scenarios, noise levels may be higher and the hypothesis class could be misspecified. In these instances, using the model learned through active learning to automatically label all data may result in a high number of errors.

We illustrate this difference between AL, SL, and auto-labeling through an example. Suppose the data consists of two concentric circles, one for each class, with the same number of points per class (Figure 2(a)). This setting is not linearly separable. We run TBAL, AL, and AL followed by SL with an error tolerance of $\epsilon_a = 1\%$ and linear classifiers and confidence functions. The results are shown in Figure 2. Note that the multiple optimal linear classifiers will all incur an error of 50%. AL algorithms can only output models that make at least 50% error. If we naively use the output model for auto-labeling, we can obtain near full coverage but incur around 50% auto-labeling error. If we use the model output by AL with threshold-based SC, labeling error is reduced. However, it can only label $\approx 25\%$ of the unlabeled data. In contrast, TBAL is able to label almost all of the data accurately (close to 100% coverage) with less than 1% auto-labeling error.

3. Theoretical Analysis

The performance of the TBAL algorithm 1 depends on many factors including the hypothesis class, the confidence function, the data sampling strategy, and the size of the training and validation data. In particular, the amount of validation data plays a critical role in determining the auto-labeling threshold, which in turn affects the accuracy and coverage.

We derive bounds on the auto-labeling error and the coverage for Algorithm 1 in terms of the size of the validation data, the number of auto-labeled points $N_a^{(k)}$, and the Rademacher complexity of the extended hypothesis class $\mathcal{H}^{T,g}$ induced by the confidence function g. Our first result, Theorem (3.1), applies to general settings and makes no assumptions on the particular form of the hypothesis class, the data distribution, and the confidence function. We then instantiate and specialize the results for a specific setting in Section 8.5. We provide the following guarantees on the auto-labeling error and the coverage.

Theorem 3.1. *(Overall Auto-Labeling Error and Coverage) Let k denote the number of rounds of the TBAL Algorithm 1. Let $n_v^{(i)}, n_a^{(i)}$ denote the number of validation and auto-labeled points at epoch i and $n^{(i)} = |X^{(i)}|$. Let $X_{pool}(A_k)$ be the set of auto-labeled points at the end of round k. $N_a^{(k)} = \sum_{i=1}^{k} n_a^{(i)}$ denote the total number of auto-labled points. Then, with probability at least $1 - \delta/2$,*

$$\widehat{\mathcal{E}}\Big(X_{pool}(A_k)\Big) \leq \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \Bigg(\underbrace{\widehat{\mathcal{E}}_a\Big(\hat{h}_i, \hat{t}_i | X_v^{(i)}\Big)}_{(a)} + \frac{4}{p_0} \underbrace{\Big(\mathfrak{R}_{n_v^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\sqrt{\frac{1}{n_v^{(i)}} \log(\frac{8k}{\delta})} \Big)}_{(b)} \Bigg)$$
$$+ \frac{4}{p_0} \underbrace{\Bigg(\sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \mathfrak{R}_{n_a^{(i)}}\big(\mathcal{H}^{T,g}\big) + \sqrt{\frac{k}{N_a^{(k)}} \log(\frac{8k}{\delta})} \Bigg)}_{(c)}$$

and with probability at least $1 - \delta/2$,

$$\widehat{\mathcal{P}}(X_{pool}(A_k)) \geq \sum_{i=1}^{k} \mathbb{P}\big(\mathcal{X}^{(i)}(\hat{h}_i, \hat{t}_i)\big) - 2\mathfrak{R}_{n^{(i)}}\big(\mathcal{H}^{T,g}\big) - \sqrt{\frac{2k^2}{N} \log\Big(\frac{8k}{\delta}\Big)}. \tag{2}$$

Discussion. We interpret the result, starting with the auto-labeling error $\widehat{\mathcal{E}}(X_{pool}(A_k))$. The term (a) $\widehat{\mathcal{E}}_a(\hat{h}_i, \hat{t}_i | X_v^{(i)})$ is the empirical conditional error in the auto-labeled region computed on the validation data in i-th round, which is at most ϵ_a. Thus, summing term (a) over all the rounds is at most ϵ_a. The term (b) provides an upper bound on the excess error over the empirical estimate term (a) as a function of the Rademacher complexity of $\mathcal{H}^{T,g}$ and the validation data used in each round. The last term (c) captures the variance in the overall estimate as a function of the total number of auto-labeled points and the Rademacher complexity of $\mathcal{H}^{T,g}$. If we let $n_v^{(i)} \geq n_v$ i.e. the minimum validation points ensured in each round, then we can see the second term is $\mathcal{O}(\mathfrak{R}_{n_v}(\mathcal{H}^{T,g}))$ and the third term is $\mathcal{O}\left(\sqrt{1/n_v}\right)$. Therefore, validation data of size $\mathcal{O}\left(1/\epsilon_a^2\right)$ in each round is sufficient to get a $\mathcal{O}(\epsilon_a)$ bound on the excess auto-labeling error. The terms with Rademacher complexities suggest that it is better to use a hypothesis class and confidence function such that the induced hypothesis class has low Rademacher complexity. While such a hypothesis class might not be rich enough to include the target function, it would still be helpful for efficient and accurate auto-labeling of the dataset which can then be used for training richer models in the downstream task. The coverage term provides a lower bound on the empirical coverage in terms of the true coverage of the sequence of estimated hypotheses $\hat{h}_i$ and threshold $\hat{t}_i$.

We note that the size of the validation data needed to guarantee the auto-labeling error in each round by Algorithm 1 is optimal up to log factors. This follows by applying a result on the tail probability of the sum of independent random variables due to Feller (1943):

Lemma 3.2. *Let c_1, c_2 and $\sigma > 0$. Let $\mathbf{x}_i \in X$ be a set of n i.i.d. points from $\mathcal{X}$ with corresponding true labels y_i. Given $(h,t) \in \mathcal{H}^{T,g}$, let $\mathbb{E}\big[\big(\ell_{0-1}^{\perp}(h,t,\mathbf{x}_i,y_i) - \mathcal{E}(h,t|\mathcal{X})\big)^2\big] = \sigma_i^2 > \sigma^2$ for every $\mathbf{x}_i$ for $\sigma_i > 0$ and let $\sum_i^n \sigma_i^2 \geq c_1$ then for every $\epsilon \in [0, \frac{\sum_{i=1}^n \sigma_i^2}{\sqrt{c_1}}]$ with $n_v < \frac{12\sigma^2}{\epsilon^2}\log(4c_2)$ the following holds w.p. at least $1/4$, $\mathcal{E}_a(h,t|\mathcal{X}) > \widehat{\mathcal{E}}_a(h,t|X) + \epsilon$.*

Therefore, if a sufficiently large validation set is not used in each round, there is a constant probability of erroneously deciding on a threshold for auto-labeling. Such a requirement on validation data also applies to active learning if we seek to validate the output model. Bypassing this requirement demands the use of approaches that are different from threshold-based auto-labeling and traditional validation techniques. We note the possibility of using recently proposed *active testing* techniques Kossen et al. (2021), a nascent approach to reducing validation data usage. The technical details and proofs are deferred to section 8. We validate our theoretical analysis through experiments on real and synthetic datasets. Please see section 9 for the details of experiments.

4. Conclusion and Future Work

In this work, we analyzed threshold-based auto-labeling systems and derived sample complexity bounds on the amount of human-labeled validation data required for guaranteeing the quality of machine-labeled data. Our study shows that these methods can accurately label a reasonable size of data using seemingly bad models when good confidence functions are available. Our analysis points to the hidden downside of these systems in terms of a large amount of validation data usage and calls for more sample-efficient methods including active testing.

References

Tiny imagenet dataset. `http://cs231n.stanford.edu/tiny-imagenet-200.zip`. Accessed: May 16, 2023.

Airbus. Airbus active labeling blog. https://acubed.airbus.com/blog/wayfinder/automatic-data-labeling-strategies-for-vision-based-machine-learning-and-ai/ , 2022. Accessed: 2022-11-18.

Maria-Florina Balcan and Phil Long. Active and passive learning of linear separators under log-concave distributions. In *Conference on Learning Theory*, pages 288–316. PMLR, 2013.

Maria-Florina Balcan, Andrei Z. Broder, and Tong Zhang. Margin based active learning. In *COLT*, 2007.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 49–56, 2009.

Kamalika Chaudhuri, Sham M. Kakade, Praneeth Netrapalli, and Sujay Sanghavi. Convergence rates of active learning for maximum likelihood estimation. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015.

C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46, 1970.

Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. In *Advances in Neural Information Processing Systems*, volume 34, pages 11933–11944, 2021.

David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pages 67–82. Springer, 2016.

Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006.

Sanjoy Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 412(19): 1767–1781, 2011. ISSN 0304-3975. Algorithmic Learning Theory (ALT 2009).

Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *International conference on computational learning theory*, pages 249–263. Springer, 2005.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Giulia DeSalvo, Mehryar Mohri, and Umar Syed. Learning with deep cascades. In *Proceedings of the Twenty-Sixth International Conference on Algorithmic Learning Theory (ALT 2015)*, 2015.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *JMLR*, 11:1605–1641, aug 2010. ISSN 1532-4435.

Ran El-Yaniv and Yair Wiener. Active learning via perfect selective classification. *Journal of Machine Learning Research*, 13(2), 2012.

William Feller. Generalization of a probability limit theorem of cramér. In *Transactions of the American Mathematical Society*, pages 361–372,, 1943.

S. Fralick. Learning to recognize patterns without a teacher. *IEEE Transactions on Information Theory*, 13(1):57–64, 1967.

Daniel Y. Fu, Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher Ré. Fast and three-rious: Speeding up weak supervision with triplet methods. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, 2020.

Olivier Gascuel and Gilles Caraux. Distribution-free performance bounds with the resubstitution error estimate. *Pattern Recognition Letters*, 13(11):757–764, 1992. ISSN 0167-8655.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Roei Gelbhart and Ran El-Yaniv. The relationship between agnostic selective classification, active learning and the disagreement coefficient. *The Journal of Machine Learning Research*, 20(1):1136–1173, 2019.

Steve Hanneke. A bound on the label complexity of agnostic active learning. ICML, 2007.

Steve Hanneke. Theory of disagreement-based active learning. *Found. Trends Mach. Learn.*, 7(2–3):131–309, jun 2014. ISSN 1935-8237.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. 2018. URL `https://arxiv.org/abs/1812.05720`.

Daniel Joseph Hsu. *Algorithms for active learning*. PhD thesis, UC San Diego, 2010.

Matti Kääriäinen. Active learning in the non-realizable case. In *International Conference on Algorithmic Learning Theory*, pages 63–77. Springer, 2006.

Julian Katz-Samuels, Jifan Zhang, Lalit Jain, and Kevin Jamieson. Improved algorithms for agnostic pool-based active classification. In *International Conference on Machine Learning*, pages 5334–5344. PMLR, 2021.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. *International Conference on Machine Learning*, 2021.

Akshay Krishnamurthy, Alekh Agarwal, Tzu-Kuo Huang, Hal Daumé III, and John Langford. Active learning for cost-sensitive classification. In *International Conference on Machine Learning*, pages 1915–1924. PMLR, 2017.

Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 33: 18237–18248, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Michel Ledoux and Michel Talagrand. Probability in banach spaces. 1991.

Chunwei Ma, Ziyun Huang, Jiayi Xian, Mingchen Gao, and Jinhui Xu. Improving uncertainty calibration of deep neural networks via truth discovery and geometric optimization. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161, pages 75–85. PMLR, 27–30 Jul 2021.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, June 2011.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 15682–15694, 2021.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN 026201825X.

Stephen Mussmann and Percy Liang. On the relationship between data efficiency and error for uncertainty sampling. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018.

John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Nikita Puchkin and Nikita Zhivotovskiy. Exponential savings in agnostic active learning through abstention. Proceedings of Machine Learning Research, pages 3806–3832. PMLR, 2021.

PyTorch. Cifar-10 pytorch tutorial. `https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html`, 2022. Accessed: 2022-11-18.

Hang Qiu, Krishna Chintalapudi, and Ramesh Govindan. Minimum cost active labeling. *arXiv preprint arXiv:2006.13999*, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

A. J. Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, 2016.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the 44th International Conference on Very Large Data Bases (VLDB)*, Rio de Janeiro, Brazil, 2018.

Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.

Nils Reimers and Iryna Gurevych. Sentence-Bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983, 2019.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning, 2020. URL `https://arxiv.org/abs/2009.00236`.

Samsung-SDS. Samsung sds auto-labeling service. https://www.samsungsds.com/en/insights/TechToolkit_2021_Auto_Labeling.html , 2022. Accessed: 2022-11-18.

Nabeel Seedat and Christopher Kanan. Towards calibrated and scalable uncertainty representations for neural networks. *arXiv preprint arXiv:1911.00104*, 2019.

Burr Settles. Active learning literature survey. 2009.

SGT. Aws sagemaker ground truth. `https://aws.amazon.com/sagemaker/data-labeling/`, 2022. Accessed: 2022-11-18.

Kulin Shah and Naresh Manwani. Online active learning of reject option classifiers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5652–5659, 2020.

Shubhanshu Shekhar, Mohammad Ghavamzadeh, and Tara Javidi. Active learning for classification with abstention. *IEEE Journal on Selected Areas in Information Theory*, 2 (2):705–719, 2021.

Superb-AI. Superb ai automated data labeling service. `https://www.superb-ai.com/product/automate`, 2022. Accessed: 2022-11-18.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

Rudy Venguswamy, Mike Levy, Anirudh Koul, Satyarth Praveen, Tarun Narayanan, Ajay Krishnan, Jenessa Peterson, Siddha Ganju, and Meher Kasam. Curator: A No-Code Self-Supervised Learning and Active Labeling Tool to Create Labeled Image Datasets from Petabyte-Scale Imagery. In *EGU General Assembly Conference Abstracts*, pages EGU21–6853, April 2021.

Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. In *Advances in Neural Information Processing Systems*, volume 34, pages 11809–11820, 2021a.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *Journal of Machine Learning Research*, 22 (201):1–73, 2021b. URL `http://jmlr.org/papers/v22/20-1061.html`.

Yair Wiener and Ran El-Yaniv. Agnostic selective classification. *Advances in neural information processing systems*, 24, 2011.

Yair Wiener and Ran El-Yaniv. Agnostic pointwise-competitive selective classification. *Journal of Artificial Intelligence Research*, 52:171–201, 2015.

Ting-Fan Wu, Chih-Jen Lin, and Ruby Weng. Probability estimates for multi-class classification by pairwise coupling. *Advances in Neural Information Processing Systems*, 16, 2003.

Yufan Zeng and Jiashan Tang. Rlc-gnn: An improved deep architecture for spatial-based graph neural network with application to fraud detection. *Applied Sciences*, 11, 06 2021.

Yinglun Zhu and Robert Nowak. Efficient active learning with abstention. *arXiv preprint arXiv:2204.00043*, 2022.

5. Appendix Organization

The appendix is organized as follows. We begin with discussing the related works in section 6, then provide the details of the auto-labeling algorithm in section 7. We summarize the notation in Table 1 and section 8.2. Then we give the proof of the main theorem (Theorem 3.1) followed by proofs of supporting lemmas. We provide details of its instantiation for finite VC-dimension hypothesis classes and the homogeneous linear separators case. Then, we provide the technical details of the lower bound (Lemma 3.2). Then we provide details of experiments in section 9 and 10. In section 10.3 we provide additional insights into auto-labeling using PaCMAP Wang et al. (2021b) visualizations of auto-labeled regions in each round.

6. Related Work

We review the related literature in this section.

There is a rich body of work on active learning on empirical and theoretical fronts Settles (2009); Dasgupta (2011); Hsu (2010); Hanneke (2014); Citovsky et al. (2021); Ren et al. (2020). In active learning, the goal is to learn the best model in the given function class with fewer labeled data than in classical passive learning. To this end, various active learning algorithms have been developed and analyzed, e.g., uncertainty sampling Tong and Koller (2001); Mussmann and Liang (2018), disagreement region based Cohn et al. (1994); Hanneke (2007), margin based Balcan et al. (2007); Balcan and Long (2013), importance sampling based Beygelzimer et al. (2009) and others Chaudhuri et al. (2015). Active learning has been shown to achieve exponentially smaller label complexity than passive learning in noiseless and low-noise settings Dasgupta et al. (2005); Balcan et al. (2007); Hanneke (2007, 2014); Balcan and Long (2013); Dasgupta (2006); Hsu (2010); Chaudhuri et al. (2015); Krishnamurthy et al. (2017); Katz-Samuels et al. (2021). This suggests, in these settings auto-labeling using active learning followed by selective classification is expected to work well. However, in practice we do not have favorable noise conditions and the hypothesis class could be misspecified i.e. it may not contain the Bayes optimal classifier. In such cases, Kääriäinen (2006) proved lower bounds on the label complexity of active learning that are order wise same as passive learning. These findings have motivated more refined goals for active learning – abstain on hard to classify points and do well on the rest of the points. This idea is captured by the Chow's excess risk Chow (1970) and some of the recent works Shekhar et al. (2021); Shah and Manwani (2020); Puchkin and Zhivotovskiy (2021); Zhu and Nowak (2022) have proved exponential savings in label complexity for active learning when the goal is to minimize Chow's excess risk. The classifier learned by these methods is equipped with the abstain option and hence it can be readily applied for auto-labeling. However, the problem of misspecification of the hypothesis class still remains. Nevertheless, it would be interesting future work to explore the connections between auto-labeling and active learning with abstention. We also note that similar works on learning with abstention are done in the context of passive learning Cortes et al. (2016).

Another closely related line of work is the selective classification where the goal is to equip a given classifier with the option to abstain from the prediction in order to guarantee prediction

quality. The foundations for selective classification are laid down in El-Yaniv and Wiener (2010); Wiener and El-Yaniv (2011); El-Yaniv and Wiener (2012); Wiener and El-Yaniv (2015) where they give results on the error rate in the prediction region and the coverage of a given classifier. However, they lack practical algorithms to find the prediction region. A recent work Gelbhart and El-Yaniv (2019) proposes a new disagreement-based active learning strategy to learn a selective classifier.

Recent work studies a practical algorithm for threshold-based selective classification on deep neural networks Geifman and El-Yaniv (2017). The algorithm estimates the prediction threshold using training samples and they bound the error rate of the selective classifier using Gascuel and Caraux (1992). We note that their result is applicable to a specific setting of a given classifier. In contrast, in the TBAL algorithm analyzed in this paper, selective classification is done in each round and the classifiers are not given a priori but instead learned via ERM on training data which is adaptively sampled in each round.

Another related work Qiu et al. (2020) studies an algorithm similar to TBAL for auto-labeling. Their emphasis is on the cost of training incurred when these systems use large-scale model classes for auto-labeling. They propose an algorithm to predict the training set size that minimizes the overall cost and provides an empirical evaluation.

Well-calibrated uncertainty scores are essential to the success of threshold-based auto-labeling. However, in practice, such scores are often hard to get. Moreover, neural networks can produce overconfident (unreliable) scores Hein et al. (2018). Fortunately, there are plenty of methods in the literature to deal with this problem Platt (1999); Wu et al. (2003). More recently, various approaches have been proposed for uncertainty calibration for neural networks Gawlikowski et al. (2021); Minderer et al. (2021); Wang et al. (2021a); Krishnan and Tickoo (2020); Ma et al. (2021); Seedat and Kanan (2019). A detailed study of calibration methods and their impact on auto-labeling is beyond the scope of this work and left as future work.

There is another line of work emerging towards auto-labeling that does not rely on getting human labels but instead uses potentially noisy but cheaply available sources to infer labels Ratner et al. (2016, 2018); Fu et al. (2020). The focus of this paper, however, is on analyzing the performance of TBAL algorithms SGT (2022); Airbus (2022) that have emerged recently as auto-labeling solutions in systems.

7. Description of the algorithm

The TBAL algorithm is given in Algorithm 1. It starts with an unlabeled pool X_{pool} and an auto-labeling error threshold ϵ. For ease of exposition, the algorithm is given the labeled validation set D_{val} of size N_v separately. In practice, it is created by selecting points at random from X_{pool}.

The algorithm starts with an initial batch of n_s random data points and obtains oracle labels for these. The algorithm works in an iterative manner using the following steps.

Algorithm 1 Threshold-based Auto-Labeling (TBAL)

Input: Unlabeled pool X_{pool}, Auto labeling error threshold ϵ_a, Seed data size n_s, Batch size for active query n_b, Labeled validation data pool D_{val}.

Output: $D_{out} = \{(\mathbf{x}_i, \tilde{y}_i) : \forall \mathbf{x}_i \in X_{pool}\}$

$X_u^{(1)} = X_{pool}; D_{val}^{(1)} = D_{val}$
$D_{query}^{(1)} = \texttt{randomly\_query\_batch}(X_u^{(1)}, n_s)$
Remove queried points from $X_u^{(1)}$
$D_{train}^{(0)} = \phi; i = 1; D_{out} = D_{out}^{(1)} = D_{query}^{(1)}$
while $X_u^{(i)} \neq \phi$ **do**
    $D_{train}^{(i)} = D_{train}^{(i-1)} \cup D_{query}^{(i)}$
    $\hat{h}_i = \texttt{empirical\_risk\_min}(\mathcal{H}, D_{train}^{(i)})$
    $\hat{t}_i = \text{Estimate Threshold}(X_u^{(i)}, \epsilon_a, \hat{h}_i, D_{val}^{(i)})$
    $D_{auto}^{(i)} = \{(\mathbf{x}, \hat{h}_i(\mathbf{x})) : \mathbf{x} \in X_u^{(i)}, g(\hat{h}_i, \mathbf{x}) \geq \hat{t}_i\}$
    Remove auto-labeled points from $X_u^{(i)}$
    $D_{val}^{(i+1)} = D_{val}^{(i)} \setminus \{\mathbf{x} \in D_{val}^{(i)} : g(\hat{h}_i, \mathbf{x}) \geq \hat{t}_i\}$
    $D_{query}^{(i+1)} = \texttt{active\_query\_batch}(\hat{h}_i, X_u^{(i)}, n_b)$
    Remove queried points from $X_u^{(i)}$
    $D_{out} = D_{out} \cup D_{auto}^{(i)} \cup D_{query}^{(i+1)}$
    $i = i + 1$
end while

Algorithm 2 Estimate Threshold

Input: $X_u^{(i)}, X_v^{(i)}, \epsilon_a, \hat{h}_i, n_0$

Output: Threshold $\hat{t}_i$

$T = \{g(\hat{h}_i, \mathbf{x}) : \mathbf{x} \in X_v^{(i)}\}$
$\hat{T}_i = \{t \in T : |X_v^{(i)}(\hat{h}_i, t)| \geq n_0\} \cup \{\infty\}$
$\hat{t}_i = \min\{t \in \hat{T}_i : \widehat{\mathcal{E}}_a(\hat{h}_i, t|X_v^{(i)}) + C_1\hat{\sigma}_i \leq \epsilon_a\}$

1. Data obtained in each iteration i is added to the training pool $D_{train}^{(i)}$. It is used to train a model $\hat{h}_i$ by performing empirical risk minimization (ERM).
2. *Finding the region to auto-label*: where $\hat{h}_i$ can auto-label accurately. The algorithm estimates a threshold $\hat{t}_i$ on the confidence score above which it can auto-label with the desired auto-labeling accuracy on the validation data (see Algorithm 2). Thresholds that have too little validation data are discarded, as they produce large errors. The minimum threshold is found such that the sum of the estimated error and an upper confidence bound, e.g., using the standard deviation of the estimated error $\widehat{\mathcal{E}}_a(\hat{h}_i, t|X_v^{(i)})$, is at most the given auto-labeling error threshold.
3. Auto-label the points in the pool, $X_u^{(i)}$, which have confidence $g(\hat{h}_i, x) > \hat{t}_i$. These are added to the set D_{out} and removed from the unlabeled pool. The validation points that fall in the auto-labeled region are also removed from the validation set so that in the next round the validation set and the unlabeled pool are from the same region and the same

distribution. Removing the auto-labeled points from X_{pool} is a crucial step in the TBAL algorithm that enables it to focus only on the remaining unlabeled regions in the next iteration.

4. If there are points left in X_{pool}, the algorithm selects points using some active querying strategy, obtains human labels for them, and adds them to the training pool.

This process continues until there are no data points left to be labeled. The algorithm then outputs the labeled dataset, which is a mixture of human- and machine-labeled points.

8. Proofs

8.1 Glossary

The notation is summarized in Table 1 below. More detailed notation is in section 8.2.

8.2 Basic Definitions and Setup

Let $\mathcal{X}$ be the instance space and $p(\mathbf{x})$ be a density function supported on $\mathcal{X}$. For any $\mathbf{x}_i \in \mathcal{X}$ let y_i be its true label. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a set of N i.i.d samples drawn from $\mathcal{X}$. Let set $\mathcal{S} \subseteq \mathcal{X}$ denote a non-empty sub-region of $\mathcal{X}$ and $S \subseteq X \cap \mathcal{S}$ be a set of $n > 0$ i.i.d. samples.

Definition 8.1. (Hypothesis Class with Abstain) We can think of the function g along with set T as inducing an extended hypothesis class $\mathcal{H}^{(T,g)}$. Let $\mathcal{H}^{T,g} = \mathcal{H} \times T$. For any function $(h, t) \in \mathcal{H}^{(T,g)}$ is defined as

$$(h,t)(\mathbf{x}) := \begin{cases} h(\mathbf{x}) & \text{if } g(h,\mathbf{x}) \geq t \\ \perp & \text{o.w.} \end{cases} \tag{3}$$

Here $(h, t)(\mathbf{x}) = \perp$ means the hypothesis (h, t) abstains in classifying the point $\mathbf{x}$. Otherwise, it is equal to $h(\mathbf{x})$.

The subset $\mathcal{S}(h, t) \subseteq \mathcal{S}$ where (h, t) does not abstain and its complement $\bar{\mathcal{S}}(h, t)$ where (h, t) abstains, are defined as follows,

$$\mathcal{S}(h,t) := \{\mathbf{x} \in \mathcal{S} : (h,t)(\mathbf{x}) \neq \perp\}, \qquad \bar{\mathcal{S}}(h,t) := \{\mathbf{x} \in \mathcal{S} : (h,t)(\mathbf{x}) = \perp\}$$

Probability Definitions: The probability $\mathbb{P}(\mathcal{S})$ of subset $\mathcal{S} \subseteq \mathcal{X}$ and the conditional probability of any subset $\mathcal{S}' \subseteq \mathcal{S}$ are given as follows,

$$\mathbb{P}(\mathcal{S}) := \mathbb{P}(\mathcal{S}|\mathcal{X}) := \int_{\mathbf{x}\in\mathcal{S}} p(\mathbf{x})d\mathbf{x}, \qquad \mathbb{P}(\mathcal{S}'|\mathcal{S}) := \frac{\mathbb{P}(\mathcal{S}'|\mathcal{X})}{\mathbb{P}(\mathcal{S}|\mathcal{X})}, \qquad \mathbb{P}(h,t|\mathcal{S}) := \mathbb{P}(\mathcal{S}(h,t)|\mathcal{S})$$

The empirical probabilities of S and $S' \subseteq S$ are defined as follows,

$$\widehat{\mathbb{P}}(S) := \frac{|S|}{|X|}, \qquad \widehat{\mathbb{P}}(S'|S) := \frac{|S'|}{|S|}, \qquad \widehat{\mathbb{P}}(h,t|S) := \frac{|S(h,t)|}{|S|}$$

Symbol	Definition
$\mathcal{X}$	feature space.
$\mathcal{Y}$	label space.
$\mathcal{H}$	hypothesis space.
h	a hypothesis in $\mathcal{H}$.
$\mathbf{x}, y$	$\mathbf{x}$ is an element in $\mathcal{X}$ and y is its true label.
$\mathcal{S}, S$	$\mathcal{S} \subseteq \mathcal{X}$ is a sub-region in $\mathcal{X}$, $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ i.i.d. samples in $\mathcal{S}$.
X_{pool}	unlabeled pool of data points.
$X_v^{(i)}, n_v^{(i)}$	set of validation points at the beginning of ith round and $n_v^{(i)} = \lvert X_v^{(i)} \rvert$.
$X_a^{(i)}, n_a^{(i)}$	set of auto-labeled points in ith round and $n_a^{(i)} = \lvert X_a^{(i)} \rvert$.
$\hat{h}_i, \hat{t}_i$	ERM solution and auto-labeling thresholds respectively in ith round.
$\mathcal{X}^{(i)}$	unlabeled region left at the beginning of ith round i.e. $\{\mathbf{x} \in \mathcal{X} : g(\hat{h}_j, \mathbf{x}) < \hat{t}_j \ \forall j < i\}$.
$X^{(i)}$	unlabeled pool left at the beginning of ith round i.e. $\{\mathbf{x} \in X_{pool} : g(\hat{h}_j, \mathbf{x}) < \hat{t}_j \ \forall j < i\}$.
$m_a^{(i)}$	number of auto-labeling mistakes in ith round.
k	number of rounds of the TBAL algorithm.
$X_{pool}(A_k)$	set of all auto-labeled points till the end of round k.
g	confidence function $g : \mathcal{H} \times \mathcal{X} \mapsto T$. Where $T \subseteq \mathbb{R}^+$, usually $T = [0, 1]$
$\mathcal{H}^{T,g}$	Cartesian product of $\mathcal{H}$ and T the range of g.
$N_a^{(k)}$	$\sum_{i=1}^{k} n_a^{(i)}$.
$\ell_{0-1}(h, \mathbf{x}, y)$	$\mathbb{1}(h(\mathbf{x}) \neq y)$.
$\ell_{\perp}(h, t, \mathbf{x})$	$\mathbb{1}(g(h, \mathbf{x}) \geq t)$.
$\ell_{0-1}^{\perp}(h, t, \mathbf{x}, y)$	$\ell_{0-1}(h, \mathbf{x}, y) \cdot \ell_{\perp}((h, t), \mathbf{x})$.
$\mathfrak{R}_n(\mathcal{H}, \ell_{0-1})$	$\mathbb{E}_{\sigma,S}\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell_{0-1}(h, \mathbf{x}_i, y_i)\right]$.
$\mathfrak{R}_n(\mathcal{H}^{T,g}, \ell_{\perp})$	$\mathbb{E}_{\sigma,S}\left[\sup_{(h,t) \in \mathcal{H}^{T,g}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell_{\perp}(h, t, \mathbf{x}_i)\right]$.
$\mathfrak{R}_n(\mathcal{H}^{T,g}, \ell_{0-1}^{\perp})$	$\mathbb{E}_{\sigma,S}\left[\sup_{(h,t) \in \mathcal{H}^{T,g}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell_{0-1}^{\perp}(h, t, \mathbf{x}_i, y_i)\right]$.
$\mathfrak{R}_n(\mathcal{H}^{T,g})$	$\mathfrak{R}_n(\mathcal{H}, \ell_{0-1}) + \mathfrak{R}_n(\mathcal{H}^{T,g}, \ell_{\perp})$.
$\mathcal{E}(h, t \lvert \mathcal{S})$	$\mathbb{E}_{\mathbf{x} \lvert \mathcal{S}}[\ell_{0-1}^{\perp}(h, t, \mathbf{x}, y)]$.
$\widehat{\mathcal{E}}(h, t \lvert S)$	$\frac{1}{\lvert S \rvert} \sum_{i=1}^{\lvert S \rvert} \ell_{0-1}^{\perp}(h, t, \mathbf{x}_i, y_i)$.
$\mathbb{P}(h, t \lvert \mathcal{S})$	$\mathbb{E}_{\mathbf{x} \lvert \mathcal{S}}[\ell_{\perp}(h, t, \mathbf{x})]$.
$\widehat{\mathbb{P}}(h, t \lvert S)$	$\frac{1}{\lvert S \rvert} \sum_{i=1}^{\lvert S \rvert} \ell_{\perp}(h, t, \mathbf{x}_i, y_i)$.
$\mathcal{E}_a(h, t \lvert \mathcal{S})$	$\mathcal{E}(h, t \lvert \mathcal{S}) / \mathbb{P}(h, t \lvert \mathcal{S})$.
$\widehat{\mathcal{E}}_a(h, t \lvert S)$	$\widehat{\mathcal{E}}(h, t \lvert S) / \widehat{\mathbb{P}}(h, t \lvert S)$.

Table 1: Glossary of variables and symbols used in this paper.

Loss Functions: The loss functions and the corresponding Rademacher complexities are defined as follows,

$$\ell_{0-1}(h, \mathbf{x}, y) := \mathbb{1}(h(\mathbf{x}) \neq y), \quad \ell_{\perp}(h, t, \mathbf{x}) := \mathbb{1}(g(h, \mathbf{x}) \geq t), \quad \ell_{0-1}^{\perp}(h, t, \mathbf{x}, y) \quad := \ell_{0-1}(h, \mathbf{x}, y) \cdot \ell_{\perp}(h, t, \mathbf{x}).$$

Error Definitions: Define the conditional error in set $\mathcal{S} \subseteq \mathcal{X}$ as follows,

$$\mathcal{E}(h,t|\mathcal{S}) := \mathbb{E}_{\mathbf{x}|\mathcal{S}}[\ell_{0-1}^{\perp}(h,t,\mathbf{x},y)] = \int_{\mathbf{x}\in\mathcal{S}} \ell_{0-1}^{\perp}(h,t,\mathbf{x},y)\cdot\frac{p(\mathbf{x})}{\mathbb{P}(\mathcal{S})}d\mathbf{x}$$

Then, the conditional error in set $\mathcal{S}(h,t)$ i.e. the subset of $\mathcal{S}$ on which (h,t) does not abstain,

$$\mathcal{E}_a(h,t|\mathcal{S}) := \mathcal{E}(h,t|\mathcal{S}(h,t)) := \mathbb{E}_{\mathbf{x}|\mathcal{S}(h,t)}[\ell_{0-1}^{\perp}(h,t,\mathbf{x},y)] = \frac{\mathcal{E}(h,t|\mathcal{S})}{\mathbb{P}(h,t|\mathcal{S})}$$

Similarly, define their empirical counterparts as follows,

$$\widehat{\mathcal{E}}(h,t|S) := \frac{1}{|S|}\sum_{\mathbf{x}_i\in S}\ell_{0-1}^{\perp}(h,t,\mathbf{x}_i,y_i), \qquad \widehat{\mathcal{E}}_a(h,t|S) := \widehat{\mathcal{E}}(h,t|S(h,t)) := \frac{1}{|S(h,t)|}\sum_{\mathbf{x}_i\in S(h,t)}\ell_{0-1}^{\perp}(h,t,\mathbf{x}_i,y_i),$$

Note that,

$$\sum_{\mathbf{x}_i\in S}\ell_{0-1}^{\perp}(h,t,\mathbf{x}_i,y_i) = \sum_{\mathbf{x}_i\in S(h,t)}\ell_{0-1}^{\perp}(h,t,\mathbf{x}_i,y_i) = \sum_{\mathbf{x}_i\in S(h,t)}\ell_{0-1}(h,\mathbf{x}_i,y_i)$$

Rademacher Complexity: The Rademacher complexities for the function classes induced by the $\mathcal{H}, T, g$ and the loss functions are defined as follows,

$$\begin{aligned}
\mathfrak{R}_n\big(\mathcal{H},\ell_{0-1}\big) &:= \mathbb{E}_{\sigma,S}\Big[\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell_{0-1}(h,\mathbf{x}_i,y_i)\Big].\\
\mathfrak{R}_n\big(\mathcal{H}^{T,g},\ell_{\perp}\big) &:= \mathbb{E}_{\sigma,S}\Big[\sup_{(h,t)\in\mathcal{H}^{T,g}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell_{\perp}(h,t,\mathbf{x}_i)\Big].\\
\mathfrak{R}_n\big(\mathcal{H}^{T,g},\ell_{0-1}^{\perp}\big) &:= \mathbb{E}_{\sigma,S}\Big[\sup_{(h,t)\in\mathcal{H}^{T,g}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell_{0-1}^{\perp}(h,t,\mathbf{x}_i,y_i)\Big].\\
\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) &:= \mathfrak{R}_n\big(\mathcal{H},\ell_{0-1}\big) + \mathfrak{R}_n\big(\mathcal{H}^{T,g},\ell_{\perp}\big)
\end{aligned}$$

Definition 8.2. (Hypothesis Class with Abstain) The function g (along with set T) induces an extended hypothesis class $\mathcal{H}^{T,g}$. Let $\mathcal{H}^{T,g} = \mathcal{H}\times T$. For any function $(h,t)\in\mathcal{H}^{T,g}$ it defined as $(h,t)(\mathbf{x}) = h(\mathbf{x})$ if $g(h,\mathbf{x})\geq t$ and $\perp$ otherwise.

Here $(h,t)(\mathbf{x}) = \perp$ means (h,t) abstains in classifying the point $\mathbf{x}$. Otherwise, it is equal to $h(\mathbf{x})$. Let $\mathcal{S}\subseteq\mathcal{X}$ denote a non-empty sub-region of $\mathcal{X}$ and $S\subseteq\mathcal{S}$ be a finite set of i.i.d. samples from $\mathcal{S}$. The subset $\mathcal{S}(h,t)\subseteq\mathcal{S}$ denotes the regions where (h,t) does not abstain. Its probability is $\mathcal{S}(h,t) := \{\mathbf{x}\in\mathcal{S} : (h,t)(\mathbf{x})\neq\perp\}, \mathbb{P}(h,t|\mathcal{S}) := \mathbb{P}(\mathcal{S}(h,t)|\mathcal{S}), \quad \widehat{\mathbb{P}}(h,t|S) := \frac{|S(h,t)|}{|S|}$. In general we use $\mathbb{P}(\mathcal{S})$ to denote the probability mass of set $\mathcal{S}$ and $\mathbb{P}(\mathcal{S}'|\mathcal{S})$ for the conditional probability of $\mathcal{S}'\subseteq\mathcal{S}$ given $\mathcal{S}$. Their empirical counterparts are $\widehat{\mathbb{P}}(S)$ and $\widehat{\mathbb{P}}(S'|S)$, respectively.

Define the conditional error in set $\mathcal{S}\subseteq\mathcal{X}$ as $\mathcal{E}(h,t|\mathcal{S}) := \mathbb{E}_{\mathbf{x}|\mathcal{S}}[\ell_{0-1}^{\perp}(h,t,\mathbf{x},y)]$ and the conditional error in set $\mathcal{S}(h,t)$ i.e. the subset of $\mathcal{S}$ on which (h,t) does not abstain

denoted by $\mathcal{E}(h,t|\mathcal{S}(h,t))$ as follows: $\mathcal{E}(h,t|\mathcal{S}) := \mathbb{E}_{\mathbf{x}|\mathcal{S}}[\ell^{\perp}_{0-1}(h,t,\mathbf{x},y)]$ and $\mathcal{E}_a(h,t|\mathcal{S}) := \mathcal{E}(h,t|\mathcal{S}(h,t)) = \frac{\mathcal{E}(h,t|\mathcal{S})}{\mathbb{P}(h,t|\mathcal{S})}$. Similarly, define their empirical counterparts as follows, $\widehat{\mathcal{E}}(h,t|S) := \frac{1}{|S|}\sum_{\mathbf{x}_i\in S}\ell^{\perp}_{0-1}(h,t,\mathbf{x}_i,y_i), \widehat{\mathcal{E}}_a(h,t|S) := \widehat{\mathcal{E}}(h,t|S(h,t)) := \frac{1}{|S(h,t)|}\sum_{\mathbf{x}_i\in S(h,t)}\ell^{\perp}_{0-1}(h,t,\mathbf{x}_i,y_i)$ In this notation, the auto-labeling error in i-th epoch is given by, $\widehat{\mathcal{E}}_a(\hat{h}_i,\hat{t}_i|X_u^{(i)}) = m_a^{(i)}/n_a^{(i)}$ where $m_a^{(i)} = \sum_{\mathbf{x}_j\in X_u^{(i)}(\hat{h}_i,\hat{t}_i)}\ell^{\perp}_{0-1}(\hat{h}_i,\hat{t}_i,\mathbf{x}_j,y_j)$ is the number of auto-labeling mistakes in i-th epoch and $n_a^{(i)} = |X_u^{(i)}(\hat{h}_i,\hat{t}_i)|$ is the number of auto-labeled points in that epoch.

Rademacher Complexity: The Rademacher complexities for the function classes induced by the $\mathcal{H}, T, g$ and the loss functions are defined as $\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) := \mathfrak{R}_n\big(\mathcal{H},\ell_{0-1}\big) + \mathfrak{R}_n\big(\mathcal{H}^{T,g},\ell_{\perp}\big)$. Let $\hat{h}_i$ and $\hat{t}_i$ be the ERM solution and the auto-labeling threshold at epoch i. Let $p_0 \in (0,1)$ be a constant such that $\mathbb{P}(\hat{h}_i,\hat{t}_i|\mathcal{X}^{(i)}) \geq p_0$ for all i. Let $X_v^{(i)}$ denote the validation set, and $n_v^{(i)}$ and $n_a^{(i)}$ the number of validation and auto-labeled points at epoch i. Let $\widehat{\mathcal{E}}_a(\hat{h}_i,\hat{t}_i|X_v^{(i)})$ be the empirical conditional risk of $\hat{h}_i$ in the region where $g(\hat{h}_i,\mathbf{x}) \geq \hat{t}_i$ evaluated on the validation data $X_v^{(i)}$.

8.3 Proofs for the General Setup

We begin by restating the theorem here and then give the proof.

Theorem 3.1. *(Overall Auto-Labeling Error and Coverage) Let k denote the number of rounds of the TBAL Algorithm 1. Let $n_v^{(i)}, n_a^{(i)}$ denote the number of validation and auto-labeled points at epoch i and $n^{(i)} = |X^{(i)}|$. Let $X_{pool}(A_k)$ be the set of auto-labeled points at the end of round k. $N_a^{(k)} = \sum_{i=1}^k n_a^{(i)}$ denote the total number of auto-labled points. Then, with probability at least $1-\delta/2$,*

$$\widehat{\mathcal{E}}\Big(X_{pool}(A_k)\Big) \leq \sum_{i=1}^{k}\frac{n_a^{(i)}}{N_a^{(k)}}\Bigg(\underbrace{\widehat{\mathcal{E}}_a\Big(\hat{h}_i,\hat{t}_i|X_v^{(i)}\Big)}_{(a)} + \frac{4}{p_0}\Big(\underbrace{\mathfrak{R}_{n_v^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\sqrt{\frac{1}{n_v^{(i)}}\log(\frac{8k}{\delta})}}_{(b)}\Big)$$
$$+\frac{4}{p_0}\Bigg(\underbrace{\sum_{i=1}^{k}\frac{n_a^{(i)}}{N_a^{(k)}}\mathfrak{R}_{n_a^{(i)}}\big(\mathcal{H}^{T,g}\big) + \sqrt{\frac{k}{N_a^{(k)}}\log(\frac{8k}{\delta})}}_{(c)}\Bigg)$$

and w.p. at least $1-\delta/2$

$$\widehat{\mathcal{P}}(X_{pool}(A_k)) \geq \sum_{i=1}^{k}\mathbb{P}\big(\mathcal{X}^{(i)}(\hat{h}_i,\hat{t}_i)\big) - 2\mathfrak{R}_{n^{(i)}}\big(\mathcal{H}^{T,g}\big) - \sqrt{\frac{2k^2}{N}\log\Big(\frac{8k}{\delta}\Big)}.$$

Proof. Recall the definition of auto-labeling error,

$$\widehat{\mathcal{E}}\Big(X_{pool}(A_k)\Big) = \sum_{i=1}^{k}\frac{m_a^{(i)}}{N_a^{(k)}}, \qquad m_a^{(i)} = n_a^{(i)}\cdot\widehat{\mathcal{E}}_a\big(\hat{h}_i,\hat{t}_i|X^{(i)}\big).$$

Here, $m_a^{(i)}$ is the number of auto-labeling mistakes made by the Algorithm in the ith round and $\widehat{\mathcal{E}}_a\big(\hat{h}_i, \hat{t}_i | X^{(i)}\big)$ is the auto-labeling error in that round. Note that we cannot observe these quantities since the true labels for the auto-labeled points are not available. To estimate the auto-labeling error of each round we make use of validation data. Using the validation data we first get an upper bound on the true error rate of the auto-labeling region i.e. $\mathcal{E}_a\big(\hat{h}_i, \hat{t}_i | \mathcal{X}^{(i)}\big)$ in terms of the auto-labeling error on the validation data $\widehat{\mathcal{E}}_a\big(\hat{h}_i, \hat{t}_i | X_v^{(i)}\big)$ and then get an upper bound on empirical auto-labeling error rate $\widehat{\mathcal{E}}_a\big(\hat{h}_i, \hat{t}_i | X^{(i)}\big)$ using the true error rate of the auto-labeling region.

We get these bounds by application of Lemma 8.3 with $\delta_3 = \delta/4k$ for each round and then apply union bound over all k epochs. Note that we have to apply the lemma twice, first to get the concentration bound w.r.t the validation data and second to get the concentration w.r.t to the auto-labeled points.

$$\mathcal{E}_a\big(\hat{h}_i, \hat{t}_i | \mathcal{X}^{(i)}\big) \leq \widehat{\mathcal{E}}_a\big(\hat{h}_i, \hat{t}_i | X_v^{(i)}\big) + \frac{2}{p_0}\mathfrak{R}_{n_v^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\sqrt{\frac{1}{n_v^{(i)}}\log\Big(\frac{8k}{\delta}\Big)}.$$

$$\widehat{\mathcal{E}}_a\big(\hat{h}_i, \hat{t}_i | X^{(i)}\big) \leq \mathcal{E}_a\big(\hat{h}_i, \hat{t}_i | \mathcal{X}^{(i)}\big) + \frac{2}{p_0}\mathfrak{R}_{n_a^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\sqrt{\frac{1}{n_a^{(i)}}\log\Big(\frac{8k}{\delta}\Big)}.$$

Substituting $\mathcal{E}_a\big(\hat{h}_i, \hat{t}_i | \mathcal{X}^{(i)}\big)$ by its upper confidence bound on the validation data.

$$\widehat{\mathcal{E}}_a\big(\hat{h}_i, \hat{t}_i | X^{(i)}\big) \leq \widehat{\mathcal{E}}_a\big(\hat{h}_i, \hat{t}_i | X_v^{(i)}\big) + \frac{2}{p_0}\mathfrak{R}_{n_v^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\mathfrak{R}_{n_a^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\sqrt{\frac{1}{n_v^{(i)}}\log\Big(\frac{8k}{\delta}\Big)} + \frac{2}{p_0}\sqrt{\frac{1}{n_a^{(i)}}\log\Big(\frac{8k}{\delta}\Big)}.$$

Having an upper bound on the empirical auto-labeling error for i^{th} round gives us an upper bound on the number of auto-labeling mistakes $m_a^{(i)}$ made in that round. It allows us to upper bound the total auto-labeling mistakes in all k rounds and thus the overall auto-labeling error as detailed below,

$$\widehat{\mathcal{E}}\Big(X_{pool}(A_k)\Big) = \sum_{i=1}^{k} \frac{m_a^{(i)}}{N_a^{(k)}}, \qquad m_a^{(i)} = n_a^{(i)} \cdot \widehat{\mathcal{E}}_a\big(\hat{h}_i, \hat{t}_i | X^{(i)}\big).$$

Since we have an upper bound on the empirical auto-labeling error in each round, we have an upper bound for each $m_a^{(i)}$, which are used as follows to get the bound on the auto-labeling error,

$$\widehat{\mathcal{E}}(X_{pool}(A_k)) = \sum_{i=1}^{k} \frac{m_a^{(i)}}{N_a^{(k)}}$$

$$
\begin{aligned}
&= \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \cdot \frac{m_a^{(i)}}{n_a^{(i)}} \\
&= \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \cdot \widehat{\mathcal{E}}_a\Big(\hat{h}_i, \hat{t}_i \big| X^{(i)}\Big) \\
&\leq \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \cdot \left(\mathcal{E}_a\big(\hat{h}_i, \hat{t}_i \big| \mathcal{X}^{(i)}\big) + \frac{4}{p_0}\mathfrak{R}_{n_a^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\sqrt{\frac{1}{n_a^{(i)}}\log\Big(\frac{8k}{\delta}\Big)}\right) \\
&\leq \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \cdot \left(\widehat{\mathcal{E}}_a\big(\hat{h}_i, \hat{t}_i \big| X_v^{(i)}\big) + \frac{4}{p_0}\mathfrak{R}_{n_v^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\sqrt{\frac{1}{n_v^{(i)}}\log\Big(\frac{8k}{\delta}\Big)}\right. \\
&\qquad\qquad \left. + \frac{4}{p_0}\mathfrak{R}_{n_a^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\sqrt{\frac{1}{n_a^{(i)}}\log\Big(\frac{8k}{\delta}\Big)}\right) \\
&\leq \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \cdot \left(\widehat{\mathcal{E}}_a\big(\hat{h}_i, \hat{t}_i \big| X_v^{(i)}\big) + \frac{4}{p_0}\mathfrak{R}_{n_v^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{4}{p_0}\mathfrak{R}_{n_a^{(i)}}\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\sqrt{\frac{1}{n_v^{(i)}}\log\Big(\frac{8k}{\delta}\Big)}\right) \\
&\qquad\qquad + \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \cdot \left(\frac{2}{p_0}\sqrt{\frac{1}{n_a^{(i)}}\log\Big(\frac{8k}{\delta}\Big)}\right)
\end{aligned}
$$

The last term is simplified as follows,

$$
\begin{aligned}
\sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \cdot \left(\frac{2}{p_0}\sqrt{\frac{1}{n_a^{(i)}}\log\Big(\frac{8k}{\delta}\Big)}\right) &= \frac{2}{p_0} \cdot \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}}\sqrt{\frac{1}{n_a^{(i)}}\log\Big(\frac{8k}{\delta}\Big)} \\
&= \frac{2}{p_0} \cdot \frac{1}{N_a^{(k)}} \sum_{i=1}^{k} \sqrt{n_a^{(i)}\log\Big(\frac{8k}{\delta}\Big)} \\
&= \frac{2}{p_0} \cdot \sqrt{\log\Big(\frac{8k}{\delta}\Big)} \cdot \sum_{i=1}^{k} \frac{\sqrt{n_a^{(i)}}}{N_a^{(k)}} \\
&\leq \frac{2}{p_0} \cdot \sqrt{\log\Big(\frac{8k}{\delta}\Big)} \cdot \sqrt{\frac{k}{N_a^{(k)}}} \\
&= \frac{2}{p_0} \cdot \sqrt{\frac{k}{N_a^{(k)}}\log\Big(\frac{8k}{\delta}\Big)}
\end{aligned}
$$

The last inequality follows from the application of the inequality $||\mathbf{u}||_1 \leq \sqrt{k}||\mathbf{u}||_2$ for any vector $\mathbf{u} \in \mathbb{R}^k$. Here we let $\mathbf{u} = [\sqrt{n}_a^{(1)}, \ldots, \sqrt{n}_a^{(k)}]$, and since $\forall i \ \sqrt{n_a^{(i)}} > 0$ so, $\sum_{i=1}^{k} \sqrt{n_a}^{(i)} = ||\mathbf{u}||_1$ and $N_a^{(k)} = ||\mathbf{u}||_2^2$.

$$
\frac{\sum_{i=1}^{k}\sqrt{n_a^{(i)}}}{N_a^{(k)}} = \frac{||\mathbf{u}||_1}{||\mathbf{u}||_2^2} \leq \frac{\sqrt{k}||\mathbf{u}||_2}{||\mathbf{u}||_2^2} = \frac{\sqrt{k}}{||\mathbf{u}||_2} = \sqrt{\frac{k}{N_a^{(k)}}}
$$

To get the bound on coverage we follow the same steps except that we can use all the unlabeled pool of size $n^(i)$ to estimate the coverage in each round which gives us the bound in terms of $n^{(i)}$ and N as follows,

$$\begin{aligned}
\widehat{\mathcal{P}}(X_{pool}(A_k)) &= \frac{1}{N}\sum_{i=1}^{k} n_a^{(i)} \\
&= \frac{1}{N}\sum_{i=1}^{k} n^{(i)} \cdot \frac{n_a^{(i)}}{n^{(i)}} \\
&= \frac{1}{N}\sum_{i=1}^{k} n^{(i)} \cdot \widehat{\mathbb{P}}(X_a^{(i)} \big| X^{(i)}) \\
&= \frac{1}{N}\sum_{i=1}^{k} n^{(i)} \cdot \widehat{\mathbb{P}}(\hat{h}_i, \hat{t}_i \big| X^{(i)}) \\
&\geq \sum_{i=1}^{k} \frac{n^{(i)}}{N}\left(\mathbb{P}\big(\hat{h}_i, \hat{t}_i \big| \mathcal{X}^{(i)}\big) - 2\mathfrak{R}_{n^{(i)}}\big(\mathcal{H}^{T,g}\big) - \sqrt{\frac{1}{n^{(i)}}\log\Big(\frac{8k}{\delta}\Big)}\right) \\
&\geq \sum_{i=1}^{k} \frac{n^{(i)}}{N}\left(\mathbb{P}\big(\hat{h}_i, \hat{t}_i \big| \mathcal{X}^{(i)}\big) - 2\mathfrak{R}_{n^{(i)}}\big(\mathcal{H}^{T,g}\big)\right) - \sqrt{\frac{k}{N}\log\Big(\frac{8k}{\delta}\Big)}
\end{aligned}$$

We bound the first term as follows,

$$\begin{aligned}
\sum_{i=1}^{k} \frac{n^{(i)}}{N}\mathbb{P}\Big(\hat{h}_i, \hat{t}_i \big| \mathcal{X}^{(i)}\Big) &= \sum_{i=1}^{k} \frac{n^{(i)}}{N} \cdot \frac{\mathbb{P}\big(\mathcal{X}^{(i)}(\hat{h}_i, \hat{t}_i)\big)}{\mathbb{P}\big(\mathcal{X}^{(i)}\big)} \\
&\geq \sum_{i=1}^{k} \left(\mathbb{P}\big(\mathcal{X}^{(i)}\big) - \sqrt{\frac{1}{N}\log\Big(\frac{8k}{\delta}\Big)}\right) \cdot \left(\frac{\mathbb{P}\big(\mathcal{X}(\hat{h}_i, \hat{t}_i)\big)}{\mathbb{P}\big(\mathcal{X}^{(i)}\big)}\right) \\
&\geq \sum_{i=1}^{k} \mathbb{P}\big(\mathcal{X}^{(i)}(\hat{h}_i, \hat{t}_i)\big) - \sqrt{\frac{1}{N}\log\Big(\frac{8k}{\delta}\Big)}
\end{aligned}$$

Substituting it back we get,

$$\begin{aligned}
\widehat{\mathcal{P}}\big(X_{pool}(A_k)\big) &\geq \sum_{i=1}^{k} \mathbb{P}\big(\mathcal{X}^{(i)}(\hat{h}_i, \hat{t}_i)\big) - 2\mathfrak{R}_{n^{(i)}}\big(\mathcal{H}^{T,g}\big) - k\sqrt{\frac{1}{N}\log\Big(\frac{8k}{\delta}\Big)} - \sqrt{\frac{k}{N}\log\Big(\frac{8k}{\delta}\Big)} \\
&\geq \sum_{i=1}^{k} \mathbb{P}\big(\mathcal{X}^{(i)}(\hat{h}_i, \hat{t}_i)\big) - 2\mathfrak{R}_{n^{(i)}}\big(\mathcal{H}^{T,g}\big) - \sqrt{\frac{4k^2}{N}\log\Big(\frac{8k}{\delta}\Big)}
\end{aligned}$$

For the last step we use the inequality $\sqrt{a} + \sqrt{b} \leq \sqrt{2(a+b)}$ for any $a, b \in \mathbb{R}^+$. ■

Next we state the result for uniform convergence between $\mathcal{E}_a\big(h,t\big|\mathcal{S}\big), \widehat{\mathcal{E}}_a\big(h,t\big|S\big)$ and give its proof.

Lemma 8.3. *For any $\delta_3, p_0 \in (0,1)$, let $\mathcal{S}$ and S be defined as above. Let $\mathbb{P}(h,t|\mathcal{S}) \geq p_0$ and $\hat{\mathbb{P}}(h,t|S) \geq p_0 \forall (h,t) \in \mathcal{H}^{T,g}$, the following holds w.p. at least $1-\delta_3/2$*

$$\big|\mathcal{E}_a\big(h,t\big|\mathcal{S}\big) - \widehat{\mathcal{E}}_a\big(h,t\big|S\big)\big| \leq +\frac{4}{p_0}\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) + \frac{2}{p_0}\sqrt{\frac{1}{n}\log(\frac{2}{\delta_3})} \qquad \forall (h,t) \in \mathcal{H}^{T,g}. \tag{4}$$

Proof. We begin with proving one side if the inequality and the other side is shown by following the same steps. The proof is based on applying the uniform convergence results for $\widehat{\mathcal{E}}(h,t|S)$ and $\widehat{\mathbb{P}}(h,t|S)$ from Lemma 8.4. The main difficulty here is that $\mathbb{E}_S[\widehat{\mathcal{E}}_a\big(h,t\big|S\big)] \neq \mathcal{E}_a\big(h,t\big|\mathcal{S}\big)$, so we cannot directly get the above result from standard uniform convergence bounds.

We prove it, by using the results from the Lemma 8.4 and restricting the region $\mathcal{S}$ such that it has probability mass at least p_0.

By definitions of $\mathcal{E}_a\big(h,t\big|\mathcal{S}\big)$ and $\widehat{\mathcal{E}}_a\big(h,t\big|S\big)$ we have,

$$\mathcal{E}(h,t|\mathcal{S}) = \mathbb{P}(h,t|\mathcal{S}) \cdot \mathcal{E}_a\big(h,t\big|\mathcal{S}\big) \qquad \text{and} \qquad \widehat{\mathcal{E}}(h,t|S) = \hat{\mathbb{P}}(h,t|S) \cdot \widehat{\mathcal{E}}_a\big(h,t\big|S\big).$$

Let $\xi_1 = \sqrt{(1/n)\log(2/\delta_1)}, \xi_2 = \sqrt{(1/n)\log(2/\delta_2)}$. From lemma 8.4 we have,

$$\mathcal{E}(h,t|\mathcal{S}) \leq \widehat{\mathcal{E}}(h,t|S) + 2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) + \xi_1 \quad \forall (h,t) \in \mathcal{H}^{T,g} \quad \text{w.p. } 1-\delta_1/2. \tag{5}$$

$$\hat{\mathbb{P}}(h,t|S) \leq \mathbb{P}(h,t|\mathcal{S}) + 2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) + \xi_2 \quad \forall (h,t) \in \mathcal{H}^{T,g} \quad \text{w.p. } 1-\delta_2/2. \tag{6}$$

Plugging in the above definitions of errors in equation (5) we get,

$$\mathbb{P}(h,t|\mathcal{S}) \cdot \mathcal{E}_a\big(h,t\big|\mathcal{S}\big) \leq \widehat{\mathbb{P}}(h,t|S) \cdot \widehat{\mathcal{E}}_a\big(h,t\big|S\big) + 2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) + \xi_1. \tag{7}$$

$$\mathcal{E}_a\big(h,t\big|\mathcal{S}\big) \leq \frac{\widehat{\mathbb{P}}(h,t|S)}{\mathbb{P}(h,t|\mathcal{S})}\widehat{\mathcal{E}}_a\big(h,t\big|S\big) + 2\frac{\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big)}{\mathbb{P}(h,t|\mathcal{S})} + \frac{\xi_1}{\mathbb{P}(h,t|\mathcal{S})}. \tag{8}$$

Substituting $\widehat{\mathbb{P}}(h,t|S)$ from equation 6 in the above equation, we get the following w.p. $(1-\delta_1/2)(1-\delta_2/2)$,

$$\begin{aligned}
\mathcal{E}_a\big(h,t\big|\mathcal{S}\big) &\leq \Big(\frac{\mathbb{P}(h,t|\mathcal{S}) + 2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) + \xi_2}{\mathbb{P}(h,t|\mathcal{S})}\Big)\widehat{\mathcal{E}}_a\big(h,t\big|S\big) + \frac{2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big)}{\mathbb{P}(h,t|\mathcal{S})} + \frac{\xi_1}{\mathbb{P}(h,t|\mathcal{S})} \quad \forall (h,t) \in \mathcal{H}^{T,g}. \\
&= \Big(1 + \frac{2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big)}{\mathbb{P}(h,t|\mathcal{S})} + \frac{\xi_2}{\mathbb{P}(h,t|\mathcal{S})}\Big)\widehat{\mathcal{E}}_a\big(h,t\big|S\big) + \frac{2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big)}{\mathbb{P}(h,t|\mathcal{S})} + \frac{\xi_1}{\mathbb{P}(h,t|\mathcal{S})}. \\
&= \widehat{\mathcal{E}}_a\big(h,t\big|S\big) + \frac{2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big)}{\mathbb{P}(h,t|\mathcal{S})} \cdot \widehat{\mathcal{E}}_a\big(h,t\big|S\big) + \frac{\xi_2}{\mathbb{P}(h,t|\mathcal{S})}\widehat{\mathcal{E}}_a\big(h,t\big|S\big) + \frac{2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big)}{\mathbb{P}(h,t|\mathcal{S})} + \frac{\xi_1}{\mathbb{P}(h,t|\mathcal{S})}.
\end{aligned}$$

Using upper bound $\widehat{\mathcal{E}}_a\big(h,t\big|S\big) \leq 1$ in the second and third terms,

$$\mathcal{E}_a\big(h,t\big|\mathcal{S}\big) \leq \widehat{\mathcal{E}}_a\big(h,t\big|S\big) + 4\frac{\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big)}{\mathbb{P}(h,t|\mathcal{S})} + \frac{\xi_1+\xi_2}{\mathbb{P}(h,t|\mathcal{S})} \qquad \forall (h,t) \in \mathcal{H}^{T,g} \; w.p. \geq 1-(\delta_1+\delta_2)/2.$$

Using $\mathbb{P}(h,t|\mathcal{S}) \geq p_0$

$$\mathcal{E}_a\big(h,t\big|\mathcal{S}\big) \leq \widehat{\mathcal{E}}_a\big(h,t\big|S\big) + \frac{4}{p_0}\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) + \frac{\xi_1+\xi_2}{p_0} \qquad \forall (h,t) \in \mathcal{H}^{T,g} \; w.p. \geq 1-(\delta_1+\delta_2)/2.$$

Letting $\delta_1 = \delta_2 = \delta_3$ and $\xi_1 = \xi_2 = \frac{\xi \cdot p_0}{2}$ gives $\xi = \sqrt{\frac{4}{p_0^2 n}\log\left(\frac{2}{\delta_3}\right)}$ and

$$\mathcal{E}_a\big(h,t\big|\mathcal{S}\big) \leq \widehat{\mathcal{E}}_a\big(h,t\big|S\big) + \frac{4}{p_0}\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) + \xi \qquad \forall (h,t) \in \mathcal{H}^{T,g} \; w.p. \geq 1-\delta_3.$$

This proves one side of the result, the other side of results follows similarly. ■

Lemma 8.4. *Let $\mathcal{S} \subseteq \mathcal{X}$ be a sub-region of $\mathcal{X}$ and $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of n i.i.d samples in $\mathcal{S}$ drawn from distribution $P_{\mathbf{x}}$. Let $\{y_1, \ldots y_n\}$ be the corresponding true labels, let $\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big)$ be the rademacher complexity of class $\mathcal{H}^{T,g}$ then for any $\delta_1, \delta_2 \in (0,1)$ we have,*

$$|\mathcal{E}(h,t|\mathcal{S}) - \widehat{\mathcal{E}}(h,t|S)| \leq 2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) + \sqrt{\frac{1}{n}\log(\frac{2}{\delta_1})} \quad \forall (h,t) \in \mathcal{H}^{T,g} \quad w.p.\ 1-\delta_1/2. \quad (9)$$

$$|\mathbb{P}(h,t|\mathcal{S}) - \hat{\mathbb{P}}(h,t|S)| \leq 2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) + \sqrt{\frac{1}{n}\log(\frac{2}{\delta_2})} \quad \forall (h,t) \in \mathcal{H}^{T,g} \quad w.p.\ 1-\delta_2/2. \quad (10)$$

Proof. The proof is similar to the standard proofs for Rademacher complexity based generalization error bound. Since we work with the modified loss function and hypothesis class to include the abstain option, thus for completeness we give the proof here. The proofs for error and probability bounds are very much the same except for the change in the loss function. We give the proof for the error bound here.

The result follows by applying McDiarmid's inequality on the function $\phi(S)$ defined as below,

$$\phi(S) := \sup_{(h,t)\in\mathcal{H}^{T,g}} \mathcal{E}(h,t|\mathcal{S}) - \widehat{\mathcal{E}}_S(h,t|S).$$

To apply McDiarmid's inequality we first show that $\phi(S)$ satisfies the bounded difference property (Lemma 8.6). This gives us,

$$\mathcal{E}(h,t|\mathcal{S}) - \widehat{\mathcal{E}}_S(h,t|S) \leq \phi(S) \leq \mathbb{E}_S[\phi(S)] + \sqrt{\frac{1}{n}\log(\frac{2}{\delta_1})} \quad \forall (h,t) \in \mathcal{H}^{T,g} \quad \text{w.p. } 1-\frac{\delta_1}{2}.$$

Using the bound on $\mathbb{E}_S[\phi(S)]$ from Lemma 8.5 we get,

$$\mathcal{E}(h,t|\mathcal{S}) \leq \widehat{\mathcal{E}}(h,t|S) + 2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big) + \sqrt{\frac{1}{n}\log(\frac{2}{\delta_1})} \quad \forall (h,t) \in \mathcal{H}^{T,g} \quad \text{w.p. } 1 - \frac{\delta_1}{2}.$$

Similarly, the bound for other side is obtained which holds w.p. $1 - \delta_1/2$ and combining both we get eq. (9).
The bound of probabilities is obtained by following the same steps as above but with a different loss function, $\ell_\perp$, since $\mathbb{P}(h,t|\mathcal{S})$ is the probability mass of the region where (h,t) does not abstain.

■

Lemma 8.5. *Let $\mathcal{S} \subseteq \mathcal{X}$ be a sub-region of $\mathcal{X}$ and $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of n i.i.d samples in $\mathcal{S}$ drawn from distribution $P_\mathbf{x}$. Let $\{y_1, \ldots y_n\}$ be the corresponding true labels and let $\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big)$ be the Rademacher complexity of the function class $\mathcal{H}^{T,g}$ defined over n i.i.d. samples. Then we have,*

$$\mathbb{E}_S\Big[\sup_{(h,t)\in\mathcal{H}^{T,g}} \mathcal{E}(h,t|\mathcal{S}) - \widehat{\mathcal{E}}(h,t|S)\Big] \leq 2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big). \tag{11}$$

Proof. Let $\tilde{S} = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots \tilde{\mathbf{x}}_n\}$ be another set of independent draws from the same distribution as of S and let the corresponding labels be $\{\tilde{y}_1, \ldots \tilde{y}_n\}$. These samples are usually termed as *ghost samples* and do not need to be counted in the sample complexity.

$$\begin{aligned}
\mathbb{E}_S\Big[\sup_{(h,t)\in\mathcal{H}^{T,g}} \mathcal{E}(h,t|\mathcal{S}) - \widehat{\mathcal{E}}(h,t|S)\Big] &= \mathbb{E}_S\Big[\sup_{(h,t)\in\mathcal{H}^{T,g}} \mathbb{E}_{\tilde{S}}\big[\widehat{\mathcal{E}}(h,t|\tilde{S})\big] - \widehat{\mathcal{E}}(h,t|S)\Big]. \\
&= \mathbb{E}_S\Big[\sup_{(h,t)\in\mathcal{H}^{T,g}} \mathbb{E}_{\tilde{S}}\big[\widehat{\mathcal{E}}(h,t|\tilde{S}) - \widehat{\mathcal{E}}(h,t|S)\big]\Big]. \\
&\leq \mathbb{E}_S\Big[\mathbb{E}_{\tilde{S}}\Big[\sup_{(h,t)\in\mathcal{H}^{T,g}} \big[\widehat{\mathcal{E}}(h,t|\tilde{S}) - \widehat{\mathcal{E}}(h,t|S)\big]\Big]\Big]. \\
&= \mathbb{E}_{S,\tilde{S}}\Big[\sup_{h\in\mathcal{H}^{T,g}} \big[\widehat{\mathcal{E}}(h,t|\tilde{S}) - \widehat{\mathcal{E}}(h,t|S)\big]\Big]. \\
&= \mathbb{E}_{S,\tilde{S}}\Big[\sup_{(h,t)\in\mathcal{H}^{T,g}} \Big[\frac{1}{n}\sum_{i=1}^n \ell_{0-1}^\perp(h,t,\tilde{\mathbf{x}}_i,\tilde{y}_i) - \frac{1}{n}\sum_{i=1}^n \ell_{0-1}^\perp(h,t,\mathbf{x}_i,y_i)\Big]\Big]. \\
&= \mathbb{E}_{\sigma,S,\tilde{S}}\Big[\sup_{h\in\mathcal{H}^{T,g}} \Big[\frac{1}{n}\sum_{i=1}^n \sigma_i\ell_{0-1}^\perp(h,t,\tilde{\mathbf{x}}_i,\tilde{y}_i) - \frac{1}{n}\sum_{i=1}^n \sigma_i\ell_{0-1}^\perp(h,t,\mathbf{x}_i,y_i)\Big]\Big]. \\
&\leq \mathbb{E}_{\sigma,\tilde{S}}\Big[\sup_{(h,t)\in\mathcal{H}^{T,g}} \frac{1}{n}\sum_{i=1}^n \sigma_i\ell_{0-1}^\perp(h,t,\tilde{\mathbf{x}}_i,\tilde{y}_i)\Big] + \\
&\qquad \mathbb{E}_{\sigma,S}\Big[\sup_{(h,t)\in\mathcal{H}^{T,g}} \frac{1}{n}\sum_{i=1}^n \sigma_i\ell_{0-1}^\perp(h,t,\mathbf{x}_i,y_i)\Big]. \\
&= 2\mathfrak{R}_n\big(\mathcal{H}^{T,g}, \ell_{0-1}^\perp\big).
\end{aligned}$$

$$\leq 2\mathfrak{R}_n\big(\mathcal{H}^{T,g}\big).$$

In the last step we used the upper bound on the Rademacher complexity from Lemma 8.7. ■

Lemma 8.6. *(Bounded Difference) Let S be a set of i.i.d samples from $P_{\mathbf{x}}$ then for $\phi(S) := \sup_{(h,t)\in\mathcal{H}^{T,g}} \mathcal{E}(h,t|\mathcal{S}) - \widehat{\mathcal{E}}(h,t|S)$, with probability at least $1-\delta$,*

$$\phi(S) \leq \mathbb{E}_S[\phi(S)] + \sqrt{\frac{1}{|S|}\log(\frac{1}{\delta})} \tag{12}$$

Proof. It is proved by showing that $\phi(S)$ satisfies the conditions (in particular the bounded difference assumption) needed for the application of McDiarmid Inequality. To see this, Let $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_i, \dots, \mathbf{x}_n\}$ and let $S' = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}'_i, \dots, \mathbf{x}_n\}$, i.e. S and S' may differ only on the i^{th} sample.

$$\begin{aligned}
|\phi(S) - \phi(S')| &= \Big| \sup_{(h,t)\in\mathcal{H}^{T,g}} \mathcal{E}(h,t|\mathcal{S}) - \widehat{\mathcal{E}}(h,t|S) - \sup_{(h,t)\in\mathcal{H}^{T,g}} \mathcal{E}(h,t|\mathcal{S}) - \widehat{\mathcal{E}}(h,t|S')\Big|. \\
&\leq \Big| \sup_{(h,t)\in\mathcal{H}^{T,g}} \Big(\mathcal{E}(h,t|\mathcal{S}) - \widehat{\mathcal{E}}(h,t|S) - \mathcal{E}(h,t|\mathcal{S}) + \widehat{\mathcal{E}}(h,t|S')\Big)\Big|. \\
&= \Big| \sup_{(h,t)\in\mathcal{H}^{T,g}} \Big(\widehat{\mathcal{E}}(h,t|S) - \widehat{\mathcal{E}}(h,t|S')\Big)\Big|. \\
&= \Big| \sup_{(h,t)\in\mathcal{H}^{T,g}} \Big(\frac{1}{|S|}\sum_{\mathbf{z}_j\in S} \ell^{\perp}_{0-1}(h,t,\mathbf{x}_j,y_j) - \frac{1}{|S'|}\sum_{\mathbf{z}_j\in S'} \ell^{\perp}_{0-1}(h,t,\mathbf{x}_j,y_j)\Big)\Big|. \\
&= \Big| \sup_{(h,t)\in\mathcal{H}^{T,g}} \Big(\frac{1}{n}\sum_{j\neq i} \big(\ell^{\perp}_{0-1}(h,t,\mathbf{x}_j,y_j) - \ell^{\perp}_{0-1}(h,t,\mathbf{x}_j,y_j)\big) \quad + \\
&\qquad \frac{1}{n}\big(\ell^{\perp}_{0-1}(h,t,\mathbf{x}_i,y_i) - \ell^{\perp}_{0-1}(h,t,\mathbf{x}'_i,y'_i)\big)\Big)\Big|. \\
&= \Big| \sup_{(h,t)\in\mathcal{H}^{T,g}} \Big(\frac{1}{n}\big(\ell^{\perp}_{0-1}(h,t,\mathbf{x}_i,y_i) - \ell^{\perp}_{0-1}(h,t,\mathbf{x}'_i,y'_i)\big)\Big)\Big|. \\
&\leq \frac{1}{n}
\end{aligned}$$

The last step follows since $\ell^{\perp}_{0-1}$ is a 0-1 loss function so letting $\ell^{\perp}_{0-1}(h,t,\mathbf{x}_i,y_i) = 1$ and $\ell^{\perp}_{0-1}(h,t,\mathbf{x}'_i,y'_i) = 0$ gives an upper bound on the difference. Thus we can apply McDiarmid Inequality here and get the bound.

■

The relationship between the Rademacher complexities is obtained using the following Lemma 8.7 due to DeSalvo et al. (2015).

Lemma 8.7. *(DeSalvo et al. (2015)) Let $\ell_{0-1}, \ell_{\perp}, \ell^{\perp}_{0-1}$ be the loss functions defined as above and the Rademacher complexities on n i.i.d. samples S be $\mathfrak{R}_n\big(\mathcal{H}, \ell_{0-1}\big), \mathfrak{R}_n\big(\mathcal{H}^{T,g}, \ell_{\perp}\big), \mathfrak{R}_n\big(\mathcal{H}^{T,g}, \ell^{\perp}_{0-1}\big)$*

respectively. Then,

$$\mathfrak{R}_n\big(\mathcal{H}^{T,g}, \ell_{0-1}^{\perp}\big) \leq \mathfrak{R}_n\big(\mathcal{H}, \ell_{0-1}\big) + \mathfrak{R}_n\big(\mathcal{H}^{T,g}, \ell_{\perp}\big) =: \mathfrak{R}_n\big(\mathcal{H}^{T,g}\big). \tag{13}$$

Detailed proof of this lemma can be found in DeSalvo et al. (2015). The result follows by expressing $\ell_{0-1} \cdot \ell_{\perp}$ as $(\ell_{0-1} + \ell_{\perp} - 1)_+$ and then applying Talagrand's contraction lemma Ledoux and Talagrand (1991).

8.4 Bounds for Finite VC-Dimension Classes

Here we specialize the auto-labeling error and coverage bounds to the setting of finite VC-dimension classes and then instantiate for a specific setting of homogeneous linear classifiers and uniform distribution.

Lemma 8.8. *Mohri et al. (2012) (Corollary 3.8 and 3.18). Let the VC-dimension of function class induced by $\mathcal{F}$ be any class of functions from $\mathcal{X} \mapsto \mathcal{Y} \cup \{\perp\}$, and $\ell : \mathcal{Y} \cup \{\perp\} \mapsto \{0, 1\}$ be a 0-1 function. Then,*

$$\mathfrak{R}_n(\mathcal{F}, \ell) \leq \sqrt{\frac{2\mathcal{V}(\mathcal{F}, \ell)}{n} \log\Big(\frac{en}{\mathcal{V}(\mathcal{F}, \ell)}\Big)}. \tag{14}$$

Corollary 8.9. *(Auto-Labeling Error and Coverage for Finite VC-dimension Classes) Let k denote the number of rounds of TBAL algorithm 1. Let $\mathcal{V}(\mathcal{H}^{T,g}) = d$ Let $X_{pool}(A_k)$ be the set of auto-labeled points at the end of round k. $N_a^{(k)} = \sum_{i=1}^{k} n_a^{(i)}$ denote the total number of auto-labeled points. With probability at least $1 - \delta$,*

$$\widehat{\mathcal{E}}(X_{pool}(A_k)) \leq \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \Bigg(\underbrace{\widehat{\mathcal{E}}_a(\hat{h}_i, \hat{t}_i | X_v^{(i)})}_{(a)} + \underbrace{\frac{4}{p_0} \sqrt{\frac{2}{n_v^{(i)}} \Big(2d \log\Big(\frac{en_v^{(i)}}{d}\Big) + \log\Big(\frac{8k}{\delta}\Big)\Big)}}_{(b)} \Bigg)$$
$$+ \underbrace{\frac{4}{p_0} \Bigg(\sqrt{\frac{2k}{N_a^{(k)}} \Big(2d \log\Big(\frac{eN_a^{(k)}}{d}\Big) + \log\Big(\frac{8k}{\delta}\Big)\Big)} \Bigg)}_{c}$$

and

$$\widehat{\mathcal{P}}(X_{pool}(A_k)) \geq \sum_{i=1}^{k} \mathbb{P}\big(\mathcal{X}^{(i)}(\hat{h}_i, \hat{t}_i)\big) - 2k \sqrt{\frac{2}{N} \Big(2d \log\Big(\frac{eN}{d}\Big) + \log\Big(\frac{8k}{\delta}\Big)\Big)}.$$

Proof. The proof follows by substituting the Rademacher complexity bounds for finite VC dimension function classes from Lemma 8.8 in the general result from Theorem 3.1.

$$\widehat{\mathcal{E}}(X_{pool}(A_k)) \leq \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \Bigg(\underbrace{\widehat{\mathcal{E}}_a(\hat{h}_i, \hat{t}_i | X_v^{(i)})}_{(a)} + \frac{4}{p_0} \underbrace{\big(\mathfrak{R}_{n_v^{(i)}}(\mathcal{H}^{T,g}) + \mathfrak{R}_{n_a^{(i)}}(\mathcal{H}^{T,g}) \big)}_{(b)} + \frac{4}{p_0} \underbrace{\sqrt{\frac{1}{n_v^{(i)}} \log\Big(\frac{8k}{\delta}\Big)}}_{(c)} \Bigg)$$

$$+ \underbrace{\frac{4}{p_0}\sqrt{\frac{k}{N_a^{(k)}}\log\Big(\frac{8k}{\delta}\Big)}}_{(d)}$$

We first simplify the terms dependent on $n_v^{(i)}$ as follows. Here we use the inequality $\sqrt{a}+\sqrt{b} \leq \sqrt{2(a+b)}$ for any $a, b \in \mathbb{R}^+$.

$$\begin{aligned}\mathfrak{R}_{n_v^{(i)}}\big(\mathcal{H}^{T,g}\big) + \sqrt{\frac{1}{n_v^{(i)}}\log\Big(\frac{8k}{\delta}\Big)} &\leq \sqrt{\frac{2d}{n_v^{(i)}}\log\Big(\frac{en_v^{(i)}}{d}\Big)} + \sqrt{\frac{1}{n_v^{(i)}}\log\Big(\frac{4k}{\delta}\Big)}, \\ &\leq \sqrt{\frac{2}{n_v^{(i)}}\bigg(2d\log\Big(\frac{en_v^{(i)}}{d}\Big) + \log\Big(\frac{8k}{\delta}\Big)\bigg)}.\end{aligned}$$

Next, we simplify the terms dependent on $n_a^{(i)}$ as follows. First, we substitute the Rademacher complexity using the bound in Lemma 8.8 and then apply the same steps as in the proof of Theorem 3.1 to bound $\sum_{i=1}^k \sqrt{n_a^{(i)}}/N_a^{(k)}$ by $\sqrt{k/N_a^{(k)}}$ followed by the application of $\sqrt{a}+\sqrt{b} \leq \sqrt{2(a+b)}$ to get the final term.

$$\begin{aligned}\sum_{i=1}^k \frac{n_a^{(i)}}{N_a^{(k)}}\mathfrak{R}_{n_a^{(i)}}\big(\mathcal{H}^{T,g}\big) + \sqrt{\frac{k}{N_a^{(k)}}\log(\frac{8k}{\delta})} &\leq \sum_{i=1}^k \frac{n_a^{(i)}}{N_a^{(k)}}\sqrt{\frac{2d}{n_a^{(i)}}\log\Big(\frac{en_a^{(i)}}{d}\Big)} + \sqrt{\frac{k}{N_a^{(k)}}\log\Big(\frac{8k}{\delta}\Big)} \\ &= \sum_{i=1}^k \frac{\sqrt{n_a^{(i)}}}{N_a^{(k)}}\sqrt{2d\log\Big(\frac{en_a^{(i)}}{d}\Big)} + \sqrt{\frac{k}{N_a^{(k)}}\log\Big(\frac{8k}{\delta}\Big)} \\ &\leq \sum_{i=1}^k \frac{\sqrt{n_a^{(i)}}}{N_a^{(k)}}\sqrt{2d\log\Big(\frac{eN_a^{(k)}}{d}\Big)} + \sqrt{\frac{k}{N_a^{(k)}}\log\Big(\frac{8k}{\delta}\Big)} \\ &\leq \sqrt{\frac{2dk}{N_a^{(k)}}\log\Big(\frac{eN_a^{(k)}}{d}\Big)} + \sqrt{\frac{k}{N_a^{(k)}}\log\Big(\frac{8k}{\delta}\Big)} \\ &\leq \sqrt{\frac{2k}{N_a^{(k)}}\bigg(2d\log\Big(\frac{eN_a^{(k)}}{d}\Big) + \log\Big(\frac{8k}{\delta}\Big)\bigg)}.\end{aligned}$$

■

8.5 Linear Classifier Setting

Next, we consider a simple setting where active learning is known to be optimal to see if TBAL can offer similar performance guarantees. To do so, we instantiate results from 3.1 to homogeneous linear separators under the uniform distribution in the realizable setting. Let $P_{\mathbf{x}}$ be supported on the unit ball in $\mathbb{R}^d$, $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : ||\mathbf{x}|| \leq 1\}$. Let

$\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : ||\mathbf{w}||_2 = 1\} = \mathbb{S}_d$, $\mathcal{H} = \{\mathbf{x} \mapsto \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) \, \forall \mathbf{w} \in \mathcal{W}\}$, the score function be given by $g(h, \mathbf{x}) = g(\mathbf{w}, \mathbf{x}) = |\langle \mathbf{w}, \mathbf{x} \rangle|$, and set $T = [0, 1]$. For simplicity, we will use $\mathcal{W}$ in place of $\mathcal{H}$.

Corollary 8.10. *(Overall Auto-Labeling Error and Coverage) Let $\hat{\mathbf{w}}_i, \hat{t}_i$ be the ERM solution and the auto-labeling margin threshold respectively at epoch i. Let $n_v^{(i)}, n_a^{(i)}$ denote the number of validation and auto-labeled points at epoch i. Let the auto-labeling algorithm run for k-epochs. Then, w.p. at least $1 - \delta/2$,*

$$\widehat{\mathcal{E}}(X_{pool}(A_k)) \leq \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \Bigg(\underbrace{\widehat{\mathcal{E}}_a(\hat{h}_i, \hat{t}_i | X_v^{(i)})}_{(a)} + \underbrace{\frac{4}{p_0} \sqrt{\frac{2}{n_v^{(i)}} \left(2d \log\left(\frac{e n_v^{(i)}}{d}\right) + \log\left(\frac{8k}{\delta}\right)\right)}}_{(b)} \Bigg)$$
$$+ \underbrace{\frac{4}{p_0} \left(\sqrt{\frac{2k}{N_a^{(k)}} \left(2d \log\left(\frac{e N_a^{(k)}}{d}\right) + \log\left(\frac{8k}{\delta}\right)\right)}\right)}_{c}$$

and w.p. at least $1 - \delta/2$,

$$\widehat{\mathcal{P}}(X_{pool}(A_k)) \geq 1 - \min_i \hat{t}_i \sqrt{4d/\pi} - 2k \sqrt{\frac{2}{N} \left(2d \log\left(\frac{eN}{d}\right) + \log\left(\frac{8k}{\delta}\right)\right)}.$$

These results imply that by ensuring the sum of the empirical validation error term (a) and the upper confidence interval to be less than ϵ_a in each round of the algorithm we can ensure that the overall auto-labeling error remains below ϵ_a. Furthermore, by applying standard VC theory to the first round, we obtain that $\hat{t}_1 \leq 1/2$. Therefore, right after the first round, we are guaranteed to label at least half of the unlabeled pool. We empirically observe that TBAL has coverage at par with active learning while respecting the auto-labeling error constraint (See Figure 4(a)).

Here we instantiate Theorem 3.1 for the case of homogeneous linear separators under the uniform distribution in the realizable setting. Formally, let $P_{\mathbf{x}}$ be a uniform distribution supported on the unit ball in $\mathbb{R}^d$, $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : ||\mathbf{x}|| \leq 1\}$. Let $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : ||\mathbf{w}||_2 = 1\} = \mathbb{S}_d$ and $\mathcal{H} = \{\mathbf{x} \mapsto \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) \, \forall \mathbf{w} \in \mathcal{W}\}$, the score function is given by $g(h, \mathbf{x}) = g(\mathbf{w}, \mathbf{x}) = |\langle \mathbf{w}, \mathbf{x} \rangle|$ and set $T = [0, 1]$. For simplicity, we will use $\mathcal{W}$ in place of $\mathcal{H}$.

Corollary 8.10. *(Overall Auto-Labeling Error and Coverage) Let $\hat{\mathbf{w}}_i, \hat{t}_i$ be the ERM solution and the auto-labeling margin threshold respectively at epoch i. Let $n_v^{(i)}, n_a^{(i)}$ denote the number of validation and auto-labeled points at epoch i. Let the auto-labeling algorithm run for k-epochs. Then, w.p. at least $1 - \delta/2$,*

$$\widehat{\mathcal{E}}(X_{pool}(A_k)) \leq \sum_{i=1}^{k} \frac{n_a^{(i)}}{N_a^{(k)}} \Bigg(\underbrace{\widehat{\mathcal{E}}_a(\hat{\mathbf{w}}_i, \hat{t}_i | X_v^{(i)})}_{(a)} + \underbrace{\frac{4}{p_0} \sqrt{\frac{2}{n_v^{(i)}} \left(2d \log\left(\frac{e n_v^{(i)}}{d}\right) + \log\left(\frac{8k}{\delta}\right)\right)}}_{(b)} \Bigg)$$

$$+\underbrace{\frac{4}{p_0}\left(\sqrt{\frac{2k}{N_a^{(k)}}\left(2d\log\left(\frac{eN_a^{(k)}}{d}\right)+\log\left(\frac{8k}{\delta}\right)\right)}\right)}_{c}$$

and w.p. at least $1-\delta/2$

$$\widehat{\mathcal{P}}(X_{pool}(A_k)) \geq 1-\min_i \hat{t}_i\sqrt{4d/\pi}-2k\sqrt{\frac{2}{N}\left(2d\log\left(\frac{eN}{d}\right)+\log\left(\frac{8k}{\delta}\right)\right)}.$$

Proof. The bound on auto-labeling error follows directly from Theorem 8.9 as the VC dimension for this setting is d. For the coverage bound, we utilize the fact that the distribution $P_{\mathbf{x}}$ is the uniform distribution over the unit ball. This enables us to obtain explicit lower bounds on the coverage. The details are given in Lemma 8.11 and Lemma 8.12. ■

Lemma 8.11. *Let the auto-labeling algorithm run for k-epochs and let $\hat{\mathbf{w}}_i, \hat{t}_i$ be the ERM solution and the auto-labeling margin threshold respectively at epoch i. Let $\mathcal{X}^{(i)}$ be the unlabeled region at the beginning of epoch i, then we have,*

$$\sum_{i=1}^{k}\mathbb{P}\big(\mathcal{X}^{(i)}(\hat{\mathbf{w}}_i,\hat{t}_i)\big) \geq 1-\min_i \hat{t}_i\sqrt{4d/\pi}. \tag{15}$$

Proof. Let $\mathcal{X}(\hat{\mathbf{w}}_i,t_i)=\{\mathbf{x}\in\mathcal{X}:|\langle\hat{\mathbf{w}}_i,\mathbf{x}\rangle|\geq\hat{t}_i\}$ denote the region that can be auto-labeled by $\hat{\mathbf{w}}_i,\hat{t}_i$. However, since in each round the remaining region is $\mathcal{X}^{(i)}$ the actual auto-labeled region of epoch i is $\mathcal{X}_a^{(i)}=\{\mathbf{x}\in\mathcal{X}^{(i)}:|\langle\hat{\mathbf{w}}_i,\mathbf{x}\rangle|\geq\hat{t}_i\}$. Let $\bar{\mathcal{X}}(\hat{\mathbf{w}}_i,t_i)$ denote the complement of set $\mathcal{X}(\hat{\mathbf{w}}_i,t_i)$.
Now observe that $\mathcal{X}_a=\cup_{i=1}^{k}\mathcal{X}_a^{(i)}$ and $\mathcal{X}(\hat{\mathbf{w}}_k,\hat{t}_k)\subseteq\mathcal{X}_a$ because any $\mathbf{x}\in\mathcal{X}(\hat{\mathbf{w}}_k,\hat{t}_k)$ is either auto-labeled in previous rounds $i<k$ or if not then it will be auto-labeled in the k^{th} round. More specifically, any $\mathbf{x}\in\mathcal{X}(\hat{\mathbf{w}}_k,\hat{t}_k)$ is either in $\cup_{i=1}^{k-1}\mathcal{X}_a^{(i)}$ and if not then it must be in $\mathcal{X}_a^{(k)}$. Thus the sum of probabilities,

$$\begin{aligned}
\sum_{i=1}^{k}\mathbb{P}\big(\mathcal{X}^{(i)}(\hat{\mathbf{w}}_i,\hat{t}_i)\big) &= \sum_{i=1}^{k}\mathbb{P}(\mathcal{X}_a^{(i)}) \\
&= \mathbb{P}(\mathcal{X}_a) \\
&\geq \min_i \mathbb{P}\big(\mathcal{X}(\hat{\mathbf{w}}_i,\hat{t}_i)\big) \\
&= 1-\max_i \mathbb{P}\big(\bar{\mathcal{X}}(\hat{\mathbf{w}}_i,\hat{t}_i)\big) \\
&\geq 1-\min_i \hat{t}_i\sqrt{4d/\pi}
\end{aligned}$$

The last step used Lemma 4 from Balcan et al. (2007)) with $\gamma_1=\hat{t}_i$ and $\gamma_2=0$ to upper bound $\mathbb{P}(\bar{\mathcal{X}}(\hat{\mathbf{w}}_i,\hat{t}_i))$ by $\hat{t}_i\sqrt{4d/\pi}$. The lemma is stated as follows in Lemma 8.12, ■

Lemma 8.12. *(Balcan et al. (2007) (Lemma 4)) Let $d \geq 2$ and let $\mathbf{x} = [x_1, \dots x_d]$ be uniformly distributed in the d-dimensional unit ball. Given $\gamma_1 \in [0,1], \gamma_2 \in [0,1]$, we have:*

$$\mathbb{P}\big((x_1, x_2) \in [0, \gamma_1] \times [\gamma_2, 1]\big) \leq \frac{\gamma_1 \sqrt{d}}{2\sqrt{\pi}} \exp\Big(-\frac{(d-2)\gamma_2^2}{2}\Big)$$

8.6 Lower Bound

Lemma 3.2. *Let c_1, c_2 and $\sigma > 0$. Let $\mathbf{x}_i \in X$ be a set of n i.i.d. points from $\mathcal{X}$ with corresponding true labels y_i. Given $(h,t) \in \mathcal{H}^{T,g}$, let $\mathbb{E}\big[\big(\ell_{0-1}^{\perp}(h,t,\mathbf{x}_i,y_i) - \mathcal{E}(h,t|\mathcal{X})\big)^2\big] = \sigma_i^2 > \sigma^2$ for every $\mathbf{x}_i$ for $\sigma_i > 0$ and let $\sum_i^n \sigma_i^2 \geq c_1$ then for every $\epsilon \in [0, \frac{\sum_{i=1}^n \sigma_i^2}{\sqrt{c_1}}]$ with $n_v < \frac{12\sigma^2}{\epsilon^2}\log(4c_2)$ the following holds w.p. at least 1/4, $\mathcal{E}_a(h,t|\mathcal{X}) > \widehat{\mathcal{E}}_a(h,t|X) + \epsilon$.*

Proof. It follows by application of Feller's result stated in lemma 8.13. ■

Lemma 8.13. *(Feller, Lower Bound on Tail Probability of Sum of Independent Random Variables) There exists positive universal constants c_1 and c_2 such that for any set of independent random variables $X_1, \dots, X_m$ satisfying $E[X_i] = 0$ and $|X_i| \leq M$, for every $i \in \{1, \dots, m\}$, if $\sum_{i=1}^m \mathbb{E}[(X)_i^2] \geq c_1$, then for every $\epsilon \in [0, \frac{\sum_{i=1}^m \mathbb{E}[(X_i)^2]\}}{M\sqrt{c_1}}]$*

$$\mathbb{P}(\sum_{i=1}^{m} X_i > \epsilon) \geq c_2 \exp\Big(\frac{-\epsilon^2}{12\sum_{i=1}^m \mathbb{E}[(X_i)^2]}\Big). \tag{16}$$

9. Experiments

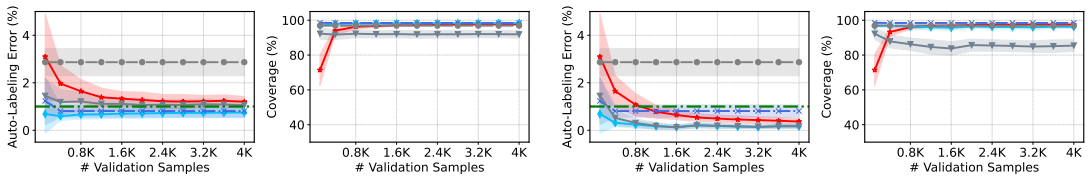
We study the effectiveness of TBAL on synthetic and real datasets. We validate our theoretical results and aim to understand the amount of labeled validation and training data required to achieve a certain auto-labeling error and coverage. We also seek to understand whether our findings apply to real data—where labels may be noisy—along with how TBAL performs compared to common baselines.

Baselines: We compare TBAL to the following methods:

a) *Passive Learning (PL)* queries a subset of the points randomly to train a model from a given model class and then uses it to predict the labels for the remaining unlabeled pool.

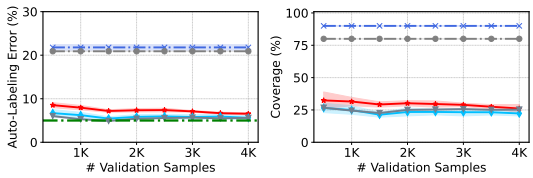
b) *Active Learning (AL)* (using margin-random query strategy, described below) trains a model from a model class and uses it to predict the labels for the remaining unlabeled pool.

c) *Passive Labeling + Selective Classification (PL+SC)* first performs passive learning to train a model from a given model class. Then it performs auto-labeling on the unlabeled data using threshold-based selective classification with the model output by passive learning. Only those unlabeled points that are deemed as fit to be labeled by the selection function are auto-labeled.

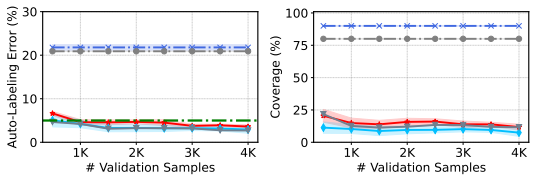


(a) Unit-Ball: with No-UCB ($C_1 = 0$), $\epsilon_a = 1\%$.

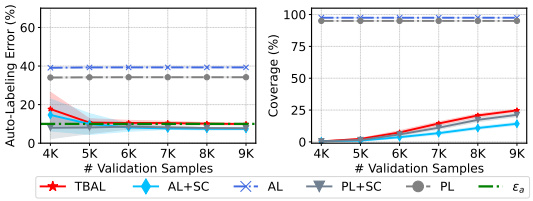
(b) Unit-Ball: with UCB ($C_1 = 0.25$), $\epsilon_a = 1\%$.



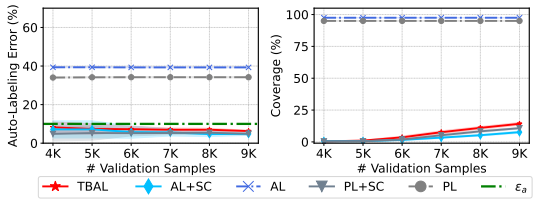
(c) IMDB with No-UCB ($C_1 = 0$), $\epsilon_a = 5\%$



(d) IMDB with UCB ($C_1 = 0.25$), $\epsilon_a = 5\%$



(e) Tiny-Imagenet with No-UCB ($C_1 = 0$), $\epsilon_a = 10\%$



(f) Tiny-Imagenet: with UCB ($C_1 = 0.25$), $\epsilon_a = 10\%$

Figure 3: Effect of variation of validation data size with and without using a UCB on error estimates. We keep the maximum number of training samples fixed at 500, 4000, 5000 for Unit-ball, IMDB and Tiny-Imagenet respectively.

d) *Active Learning + Selective Classification (AL+SC)* first performs active learning (using margin-random query strategy) to train a model from a given model class. It then performs auto-labeling using threshold-based selective classification with the model output by active learning.

For selective classification in the above methods, we use Algorithm 2 to estimate the threshold and use it to perform auto-labeling. In experiments, we adapt Algorithm 2 slightly—instead of estimating a single threshold for all classes, we estimate thresholds for each class separately.

Active Querying Strategy: We use the *margin-random* query strategy for querying the next batch of training data. In this strategy, the algorithm first sorts the points based on the margin (uncertainty) score and then selects the top Cn_b ($C > 1$) points from which n_b points are picked at random. This is a simple and computationally efficient method that balances the exploration and exploitation trade-off. We note that other active-querying strategies exist; we use margin-random as our standard querying strategy to keep the focus on comparing auto-labeling—not active learning approaches.

Datasets: We use five datasets, three synthetic and two real. For each dataset, we split the data into two sufficiently large pools. One is used as X_{pool} on which auto-labeling algorithms are run and the other is used as X_{val} from which the algorithms subsample validation data.

a) *Unit-Ball:* is a synthetic dataset of uniformly sampled points from the d-dimensional unit ball. The true labels are generated using a homogeneous linear separator with

$\mathbf{w} = [1/\sqrt{d}, \ldots, 1/\sqrt{d}]$. We use $d = 30$ and generate N=20k samples, out of which 16k are in X_{pool} and 4k are in X_{val}.

b) *Tiny-Imagenet* tin is a subset of the larger ImageNet Deng et al. (2009) dataset, designed for image classification tasks. It consists of 200 classes, each with 500 training images and 50 validation and test images. With a total of 100,000 images, Tiny ImageNet provides a diverse and challenging dataset. We use precomputed embeddings of the images using CLIP Radford et al. (2021).

c) *IMDB Reviews* Maas et al. (2011) is a comprehensive collection of movie reviews, consisting of 50,000 individual reviews. It is a balanced dataset of positive and negative labels. We use the standard train set of size 25K and split into X_{pool} and X_{val} of sizes 20K and 5K respectively. We compute embeddings of reviews using a pretrained transformer model Reimers and Gurevych (2019).

d) *CIFAR-10* Krizhevsky et al. (2009) is an image dataset. We randomly split the standard training set into X_{pool} of size 40k and the validation pool of size 10k. We use the raw features for model training.

Models and Training: For the linear models, we use SVM with the usual hinge loss and train it to loss tolerance 10^{-5}. To train a multi-layer perceptron (MLP) on the precomputed embeddings of IMDB and Tiny-Imagenet we we use SGD with a learning rate of 0.05 and batch sizes 64, 256 respectively. To train the medium CNN we use SGD with a learning rate of 10^{-2}, batch size 256, and momentum of 0.9. More details on model training are in the Appendix.

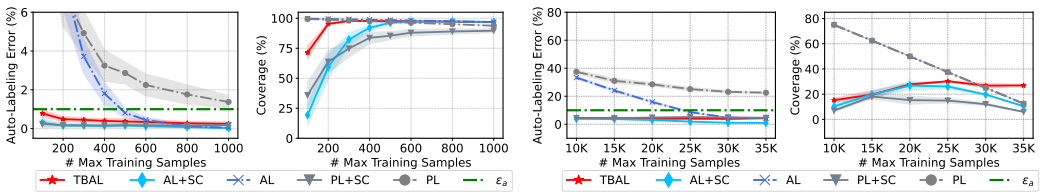
The score function g: For SVMs we use the standard implementations of Wu et al. (2003); Platt (1999) in `sklearn` to get the prediction probabilities and use them as the score function. Neural networks use softmax output.

9.1 Role of Validation Data

The TBAL algorithm uses validation data to estimate the auto-labeling errors at various thresholds to determine the threshold for automatically labeling points accurately. Thus, it is crucial to have accurate estimates of the auto-labeling errors. Our analysis shows that to get such good estimates, large amounts of validation data are needed. In this section we study the effect of varying the amount of validation data on auto-labeling performance .

Setup: We fix the maximum training data size N_q and run the algorithm with different amounts of validation data. We also consider the two cases where the algorithm uses an upper confidence bound on the error estimate and where it does not. We use the Unit-Ball, Tiny-Imagenet and IMDB datasets for this study with $N_q = 500, 2000$ and 5000, respectively, and the auto-labeling error thresholds $\epsilon_a = 1\%$, 5%, 10%, respectively. Initial seed data of size n_s is 20% of N_q and query batch size n_b is 5% of N_q; $C = 2$ for both AL and TBAL for both datasets. We give the same initial seed samples of size n_s to all the methods to ensure they have the same starting point.

Results: Figure 3 demonstrates the impact of validation data on the performance of TBAL and other algorithms. The auto-labeling error and coverage of TBAL and other methods are



(a) Unit ball: varying training samples size, validation samples size=4K, $\epsilon_a = 1\%, C_1 = 0.25$

(b) CIFAR-10: varying training samples size,validation size=10K, $\epsilon_a = 10\%, C_1 = 0.25$

Figure 4: Results for varying the maximum number of samples algorithm can use for training while providing sufficient validation samples.

affected by the amount of validation data provided. When the validation data is insufficient, the auto-labeling error of TBAL increases. However, as more validation data is used, the auto-labeling error and coverage of TBAL improves. Providing too little X_{val} can lead to incorrect estimates of the auto-labeling error, which in turn results in poor auto-labeling performance. This is further highlighted in our theoretical analysis as seen in Theorem 3.1. We also take a more nuanced look at the performance when the algorithm uses an upper confidence bound (with $C_1 = 0.25$) on the estimates and when it does not. We see the effects of not using any upper confidence bound (i.e. $C_1 = 0$). Figures 3(a)3(c), 3(e) show the results when $C_1 = 0$ and Figures 3(b), 3(d), 3(f) show the results when $C_1 = 0.25$. These show that not using UCB leads to high auto-labeling error (i.e. not meeting the guarantees) even when there is a sufficient amount of validation data. This can happen with high coverage as well—yielding a dataset with large error. On the other hand using UCB, i.e. $C_1 = 0.25$, the algorithm can keep the auto-labeling error below the given threshold but suffers in coverage.

9.2 Role of Training Data Size

The labels queried for model training also play an important role in the process while incurring costs to obtain. The next experiment focuses on the impact of training data on auto-labeling.

Setup: We limit the amount of training data the algorithm can use and record the resulting auto-labeling error and coverage. We ensure all algorithms have sufficiently large but equal amounts of validation data. We run on Unit-Ball, IMDB, Tiny-Imagenet, CIFAR-10 datasets with the same values of n_s, n_b, and C as in previous experiments.

Results: Figures 4(a), 4(b) , and 2(b) indicate that TBAL and methods utilizing selective classification (AL+SC, PL+SC) maintain a high level of accuracy, even in scenarios where minimal training samples are used. This is expected as the threshold estimation method (when used with sufficient validation data) will find auto-labeling thresholds such that the auto-labeling error does not exceed ϵ_a. The impact of training data size can be seen clearly in the coverage achieved by the algorithms. As expected, with fewer training samples the model has low accuracy leading to low coverage. However, as more samples are acquired, a more accurate model within the function class is learned, resulting in increased coverage. The Appendix has additional discussion and results on other datasets.

10. Additional Experiments

In this section, we discuss additional experiments on the role of hypothesis class in auto-labeling datasets and experiments for studying the role of confidence function in auto-labeling. Finally, we visualize PaCMAP embeddings of the CIFAR-10 and MNIST data points to get a sense of auto-labeling regions in various rounds of the algorithm.

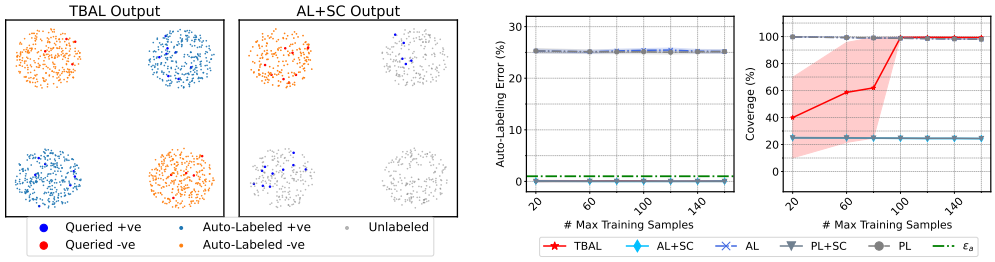
10.1 Additional Experiments on Role of the Hypothesis Class

First, we provide details of the datasets used in the following experiments.

Datasets:

XOR: is a synthetic dataset. Recall that it is created by uniformly drawing points from 4 circles, each centered at the corners of a square of with side length 4 centered at the origin. Points in the diagonally opposite balls belong to the same class. We generate a total of $N = 10,000$ samples, out of which we keep $8,000$ in X_{pool} and $2,000$ in the validation pool X_{val}.

MNIST: Deng (2012) is a standard image dataset of hand-written digits. We randomly split the standard training set into X_{pool} and the validation pool X_{val} of sizes 48,000 and 12,000 respectively. While training a linear classifier on this dataset we flatten the 28×28 images to vectors of size 784.

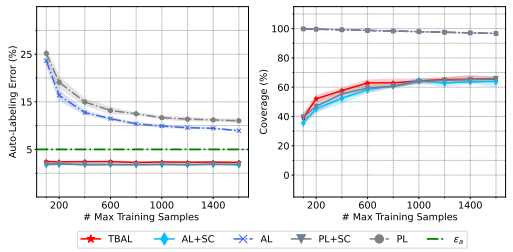


(a) Output of TBAL and AL+SC on XOR dataset

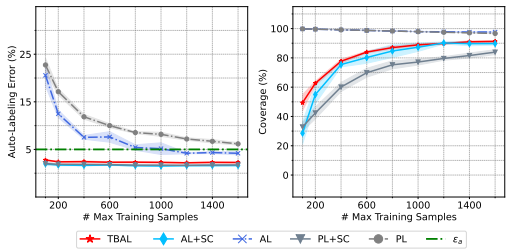
(b) Auto labeling performance of various methods

Figure 5: Comparison of Threshold-Based Auto-Labeling (TBAL) and Active-Learning followed by Selective Classification (AL+SC) on XOR-dataset. Left figure (a) shows samples that were auto-labeled, queried, and left unlabeled by these methods. Right figure (b) shows the auto-labeling error and coverage achieved. The lines show the mean and the shaded region shows 1-standard deviation estimated over 10 trials with different random seeds.

XOR Experiment: We run the TBAL algorithm 1 with an error tolerance of $\epsilon_a = 1\%$. we use 20% of N_q as seed training data and keep query size n_b as 5% of N_q. We compare it with active learning and active learning followed by selective classification. The given function class and selective classifier are both linear for all the algorithms. The results are shown in Figure 5. Clearly, there is no linear classifier that can correctly classify this data. We note that there are multiple optimal classifiers in the function class of linear classifiers and they will all incur an error of 25%. So, active learning algorithms can only output



(a) Auto-labeling MNIST data using a linear classifier. The validation size used here is 12k.



(b) Auto-labeling MNIST data using LeNet classifier. The validation size used here is 12k.

Figure 6: Auto-labeling performance on MNIST data using different models (hypothesis classes) as a function of samples available for training. The left figure (a) shows the results with the linear classifier and the right figure (b) shows the results with the LeNet classifier. The auto-labeling error threshold $\epsilon_a = 5\%$ in both experiments and the algorithms are given the same amount of validation data. The lines show the mean and the shaded region shows 1-standard deviation estimated over 5 trials with different random seeds.

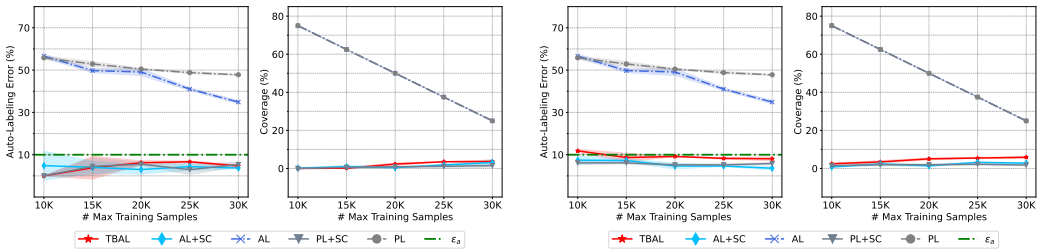
models that make at least 25% error. If we naively use the output model for auto-labeling, we can obtain near full coverage but incur 25% auto-labeling error. If we use the model output by active learning with threshold-based selective classification, then it can attain lower error in labeling. However, it can only label $\approx 25\%$ of the unlabeled data. In contrast, the TBAL algorithm is able to label almost all of the data accurately, i.e., attain close to 100% coverage, with an error close to 1% auto-labeling error.

MNIST Experiment: For training LeNet LeCun et al. (1998) we use SGD with a learning rate of 0.1, batch size of 32, and train for 20 epochs. We use auto-labeling error threshold $\epsilon_a = 5\%$. We use 20% of N_q as seed training data and keep query size n_b as 5% of N_q. The results are presented in Figure 6 we observe that TBAL using less powerful models can still yield highly accurate datasets with a significant fraction of points labeled by the models. This confirms the notion that bad models can still provide good datasets.

10.2 Role of Confidence Function

The confidence function g is used to obtain uncertainty scores is an important factor in auto-labeling. In particular, for threshold-based auto-labeling we expect the scores of correctly classified and incorrectly classified points to be reasonably well separated and if this is not the case then the algorithm will struggle to find a good threshold even if the given classifier has good accuracy in certain regions.

Setup We perform auto-labeling on the CIFAR-10 dataset using a small CNN network with 2 convolution layers followed by 3 fully connected layers PyTorch (2022). We use two different scores for auto-labeling, a) Usual softmax output b) Energy score with temperature = 1 LeCun et al. (2006). We vary the maximum number of training samples N_q and keep 20% of N_q as seed samples and query points in the batches of 10% of N_q. The model is



(a) Auto-labeling CIFAR-10 data using a small network and softmax scores. Validation size = 10k.

(b) Auto-labeling CIFAR-10 data using a small network and energy scores. Validation size = 10k.

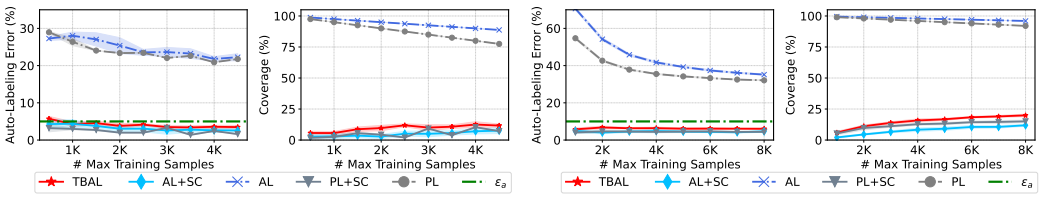
Figure 7: Auto-labeling performance on CIFAR-10 data using a small network and different scoring functions. The left figure (a) shows the results with softmax scores and the right figure (b) shows the results with the energy score. The auto-labeling error threshold $\epsilon_a = 10\%$ in both experiments and algorithms are given the same amount of validation data. The lines show the mean and the shaded region shows 1-standard deviation estimated over 5 trials with different random seeds.

trained for 50 epochs, using SGD with a learning rate of 0.05, batch size = 256, weight decay = $5e^{-4}$ and momentum=0.9. The auto-labeling threshold is set to 10%.

Results The results with softmax scores and energy scores used as confidence functions can be seen in Figures 9(a) and 9(b) respectively. We see that for both of these cases, TBAL does not obtain a coverage of more than $\approx 6\%$. We observe that using the energy score as the confidence function performed marginally better than using the softmax scores. We note that this is the case even though the test accuracies of the trained models were around 50% for most of the rounds. Note that CIFAR-10 has 10 classes, so an accuracy of 50% is much better than random guessing and one would expect to be able to auto-label a significant chunk of the data with such a model. However, the softmax scores and energy scores are not well calibrated, and therefore, when used as confidence functions, they result in a poor separation between correct and incorrect predictions by the model. This can be seen in Figure 9 where neither of the softmax and energy scores provides a good separation between the correct and incorrect predictions. We can also see that the energy score is marginally better in terms of the separation, which allows it to achieve slightly better auto-labeling coverage in comparison to using softmax scores. This suggests that more investigation is needed to understand the properties of good confidence functions for auto-labeling which is left to future work. For a more detailed visualization of the rounds of TBAL for this experiment, see Figures 11 and 12 in the Appendix.

10.3 Auto Labeling Visualization

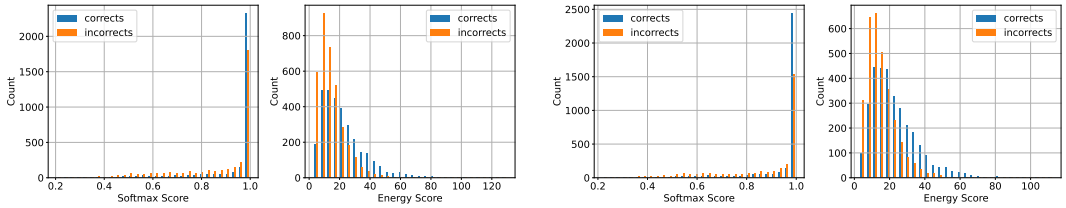
In this section, we visualize the process of TBAL. We use the dimensionality reduction method, PaCMAP Wang et al. (2021b), to visualize the features of the samples. For neural network models, we visualize the PaCMAP embeddings of the penultimate layer's output and for linear models, we use PaCMAP on the raw features. In these figures, each row



(a) IMDB: varying training samples size, validation samples size=5K, $\epsilon_a = 5\%, C_1 = 0.25$

(b) Tiny-Imagenet: varying training samples size,validation size=10K, $\epsilon_a = 10\%, C_1 = 0.25$

Figure 8: Results for varying the maximum number of samples algorithm can use for training while providing sufficient validation samples.



(a) Histogram of scores in round 2.

(b) Histogram of scores in round 6.

Figure 9: Histograms of scores computed on the validation data in a few rounds of TBAL run on CIFAR-10 with a small net. We picked two rounds where it auto-labeled the most i.e. around 800 points.

corresponds to one TBAL round. Each figure shows a few selected rounds of auto-labeling. Each figure has four columns (left to right), which show: **a**) The samples that are labeled by TBAL in the round are shown in that row. **b**) The embeddings for training samples in that round. **c**) The embeddings for validation data points in that round. **d**) The score distribution for the validation dataset in that round.

In Figure 10 we see visualizations for auto-labeling on the MNIST data using linear models. In this setting the data exhibits clustering structure in the PaCMAP embeddings learned on the raw features and the confidence (probability) scores produced are also reasonably well calibrated which leads to good auto-labeling performance.

The visualizations for the process of TBAL on CIFAR-10 using the small network (a small CNN network with 2 convolution layers followed by 3 fully connected layers PyTorch (2022)) with energy scores and soft-max scores for confidence functions are shown in Figures 11 and 12 respectively. We note that both the energy scores and soft-max scores do not seem to be calibrated to the correctness of the predicted labels which makes it difficult to identify subsets of unlabeled data where the current hypothesis in each round could have potentially auto-labeled. We also note that the test accuracies of the trained models were around 50% for most of the rounds of TBAL even though the small network model is not a powerful enough model class for this dataset. Note that CIFAR-10 has 10 classes, so the accuracy of 50% is much better than random guessing and one would expect to be able to auto-label a reasonably large chunk of the data with such a model if accompanied by a good confidence

function. This highlights the important role that the confidence function plays in a TBAL system and more investigation is needed which is left to future work.

Note that, in our auto-labeling implementation we find class specific thresholds. In these figures, we show the histograms of scores for all classes for simplicity. We want to emphasize that the visualization figures in this section are 2D representations (approximation) of the high-dimensional features (either of the penultimate layer or the raw features).

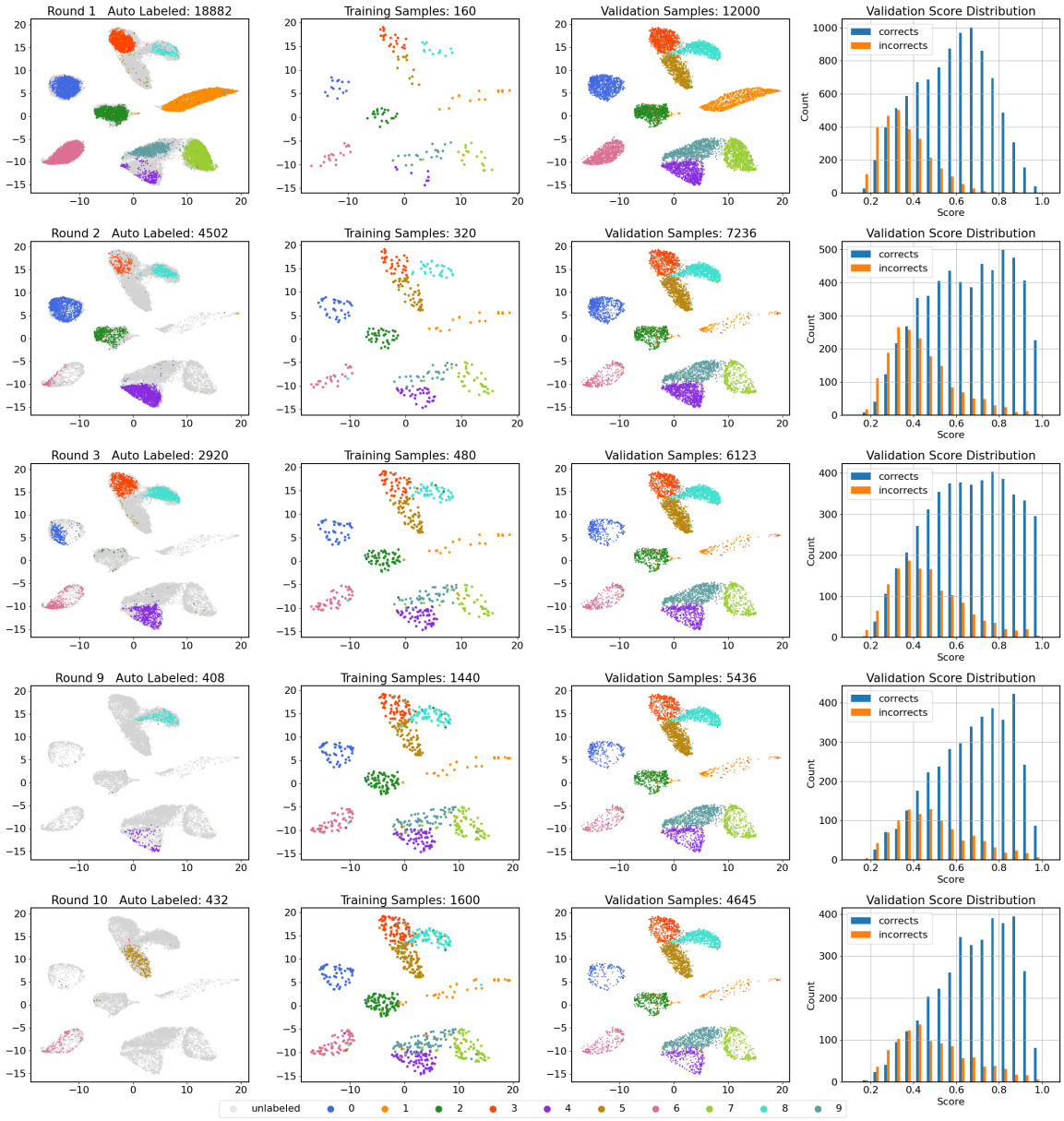


Figure 10: Auto-labeling MNIST data using linear classifiers. Validation size = 12k. Maximum training samples = 1600. Each round algorithm queries 160 samples. Coverage of auto-labeling is 62.9% with 98.0% accuracy. For the rounds we show, the test error rates are 21.4%, 13.9%, 12.5%, 10.2%, and 9.8%, respectively. For four columns (left to right), we show: **a**) The samples that are labeled by TBAL in this round. **b**) The embeddings for training samples. **c**) The embeddings for validation data points. **d**) The score distribution for the validation dataset.

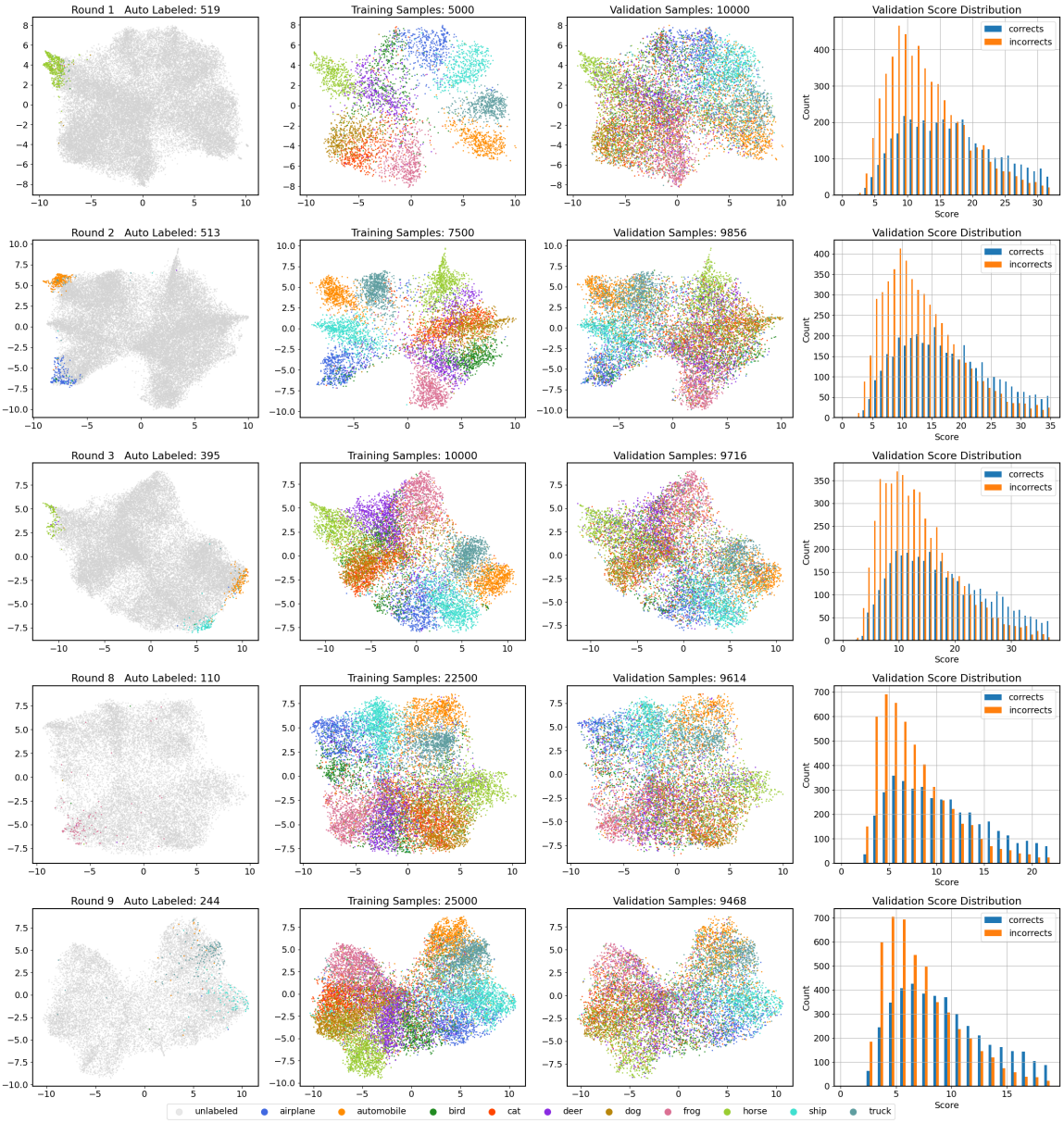


Figure 11: Auto-labeling CIFAR-10 data using a small network and energy scores. Validation size = 10k. Maximum training samples = 25k. Each round algorithm queries 2500 samples. Coverage of auto-labeling is 5.3% with 90.0% accuracy. For the rounds we show, the test error rates are 56.6%, 55.2%, 55.6%, 53.0%, and 49.3% respectively. For four columns (left to right), we show: **a**) The samples that are labeled by TBAL in this round. **b**) The embeddings for training samples. **c**) The embeddings for validation data points. **d**) The score distribution for the validation dataset.

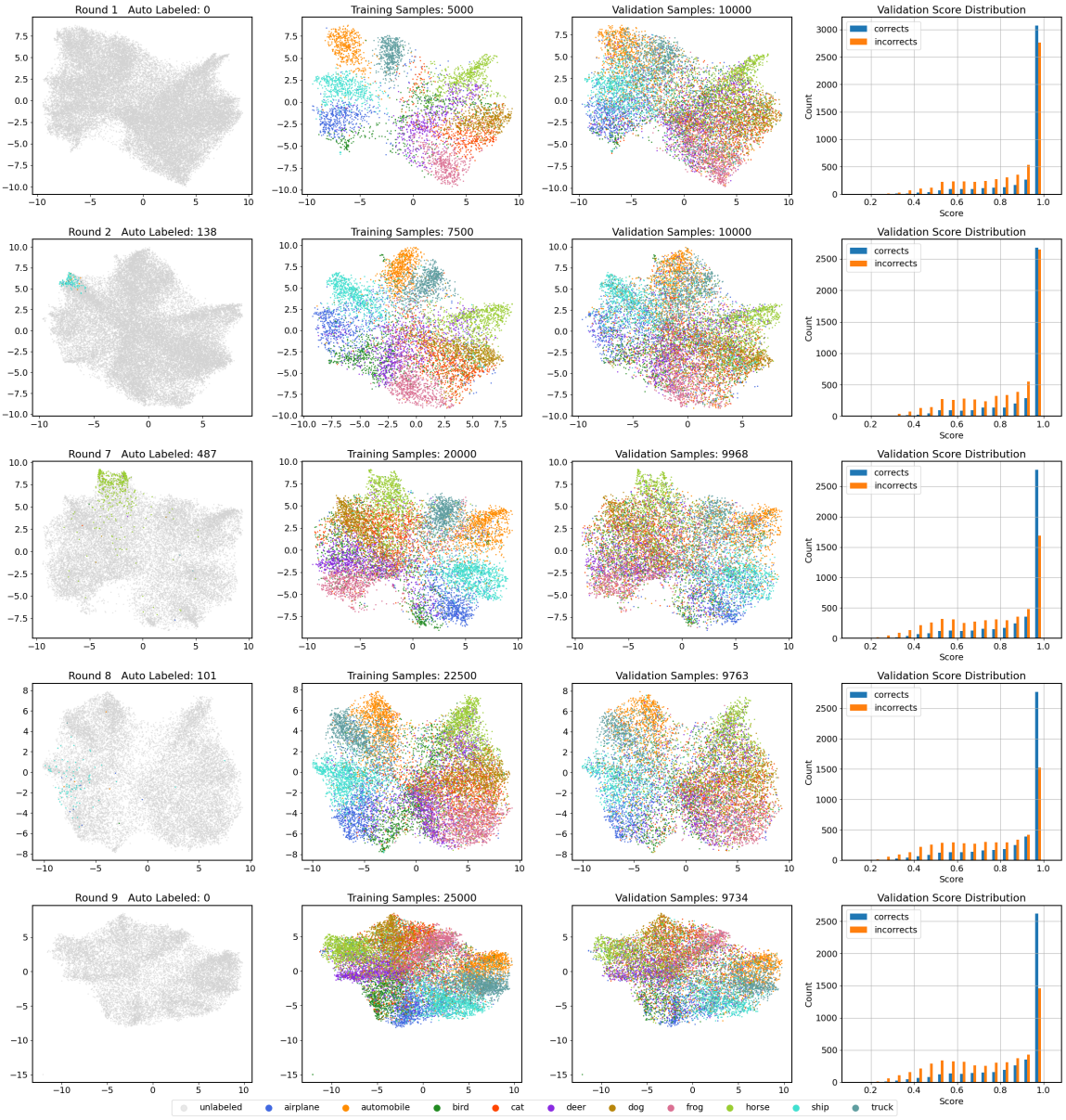


Figure 12: Auto-labeling CIFAR-10 data using a small network and softmax scores. Validation size = 10k. Maximum training samples = 25k. Each round algorithm queries 2500 samples. Coverage of auto-labeling is 2.3% with 91.0% accuracy. For the rounds visualized here in each row, the test error rates of the trained classifiers are 56.6%, 59.1%, 52.8%, 50.5%, and 51.7% respectively. For four columns (left to right), we show: **a**) The samples that are labeled by TBAL in this round. **b**) The embeddings for training samples. **c**) The embeddings for validation data points. **d**) The score distribution for the validation dataset.