ACTIVELAB: ACTIVE LEARNING WITH RE-LABELING BY MULTIPLE ANNOTATORS

Hui Wen Goh
Cleanlab
huiwen@cleanlab.ai

Jonas Mueller
Cleanlab
jonas@cleanlab.ai

ABSTRACT

In real-world data labeling, annotators often provide imperfect labels. It is thus common to employ multiple annotators to label data with some overlap between their examples. We study active learning in such settings, aiming to train an accurate classifier by collecting the fewest total annotations. Here we propose ActiveLab, a practical method to decide what to label next that works with any classifier model and can be used in pool-based batch active learning with one or multiple annotators. ActiveLab automatically estimates when it is more informative to re-label examples vs. labeling entirely new ones. This is a key aspect of producing high quality labels and trained models within a limited annotation budget. In experiments on image and tabular data, ActiveLab reliably trains more accurate classifiers with far fewer annotations than a wide variety of popular active learning methods.

1 INTRODUCTION

Model-agnostic active learning methods use outputs from some *arbitrary* type of trained prediction model in order to identify the *most informative* data to label, so that a more accurate version of the same model can be trained. Such general approaches are popular because they can be directly applied to many data modalities (image, text, etc.) as long as a reasonable model can be trained. Focusing on highly practical settings, we consider model-agnostic pool-based active learning with multiple data annotators that label a batch of many examples in between model (re)training runs. This setting is easy to setup and allows us to address common issues in real-world active learning such as: labelers who are imperfect, or expensive model (re)training that cannot be executed every time a new example is labeled. Working with annotators that may provide incorrect labels, it is useful to sometimes ask new annotators to provide extra labels for examples previously labeled by others. This allows us to verify the current consensus label or estimate a better one.

Here we introduce ActiveLab[1], a straightforward **active** learning algorithm that estimates *when* such re-**lab**eling will be more effective than labeling an entirely new example. A very general approach, ActiveLab can be used: with any type of classifier model (or ensemble of multiple models) and data modality, for active learning with multiple annotators where the set of annotators changes over time, for traditional active learning where each example is labeled at most once (Appendix H), and for active label cleaning where all data is already labeled by at least one annotator and the goal is to establish the highest quality consensus labels within a limited annotation budget.

2 METHODS

This paper focuses on classification tasks with K classes, for which some (arbitrary) classifier model $\mathcal{M}$ can be trained. For our ith example with feature values X_i, this model predicts a class probability vector $\widehat{p}_{\mathcal{M}}(Y_i \mid X_i)$ estimating the likelihood that X belongs to each class $k \in [K] := \{1, 2, \ldots, K\}$.

In the *pool*-based *batch* active learning settings we consider, each round involves the steps described below. In the beginning, we start with a training set $\mathcal{D}$ of examples that have at least one (noisy)

[1]Code for running our method: https://github.com/cleanlab/cleanlab/
Code for reproducing our benchmarks: https://github.com/cleanlab/multiannotator-benchmarks/tree/main/active_learning_benchmarks

annotation, where some of these examples may have been labeled by multiple annotators. We also have a pool of unlabeled examples $\mathcal{U}$ that have zero annotations. Our proposed active learning method may choose to collect new labels for examples in either $\mathcal{D}$ or $\mathcal{U}$. Based on classifier predictions $\widehat{p}$ and the currently-observed annotations $\mathcal{D}$, ActiveLab estimates an acquisition score s_i for each example. Examples with the lowest s_i values are those for which collecting an additional label is expected to be most informative when subsequently training $\mathcal{M}$. To avoid overfit/biased results, classifier predictions $\widehat{p}$ should be *out-of-sample*, coming from a copy of the model $\mathcal{M}$ that has never been trained with the example it is asked to predict the class of.

Active learning with multiple annotators

Input: $\mathcal{D}$: labeled examples with at least one annotation
Input: $\mathcal{U}$: unlabeled pool of examples (not yet annotated)
1: **for** $r = 1, 2, \ldots$ {rounds of active learning} **do**
2: Infer consensus labels $\widehat{Y}_i$ for annotated examples $x_i \in \mathcal{D}$ (some have multiple annotations)
3: Train classifier model $\mathcal{M}_r$ with these labels: $(x_i, \widehat{Y}_i)$
4: Obtain (out-of-sample) predicted class probabilities: $\widehat{p} = \mathcal{M}_r(x)$ for all $x \in \mathcal{D} \cup \mathcal{U}$
5: Use active learning method to score all examples: $s_i = A(\widehat{p}_i; \mathcal{D})$ for all $x_i \in \mathcal{D} \cup \mathcal{U}$
6: Assemble batch $\mathcal{B}$ of the B best-scoring examples, collect **one** additional label Y_{ij} for each $x_i \in \mathcal{B}$, and add new $\{Y_{ij}\}$ to the training data (updating $\mathcal{D}$,$\mathcal{U}$)
7: **end for**

We can obtain out-of-sample predictions for every $x_i \in \mathcal{D}$ by fitting our model via k-fold cross-validation in Step 3. For examples currently in the unlabeled pool $x \in \mathcal{U}$, Step 6 can collect their first label, and there may be already-labeled examples $x \in \mathcal{D}$ in the selected batch $\mathcal{B}$ for which we collect yet another label. There are many ways to operationalize the collection of labels in Step 6 of active learning. The examples to acquire an extra label for could be divided amongst a limited pool of annotators (some of which labeled other examples in previous active learning rounds), or these examples could be given to new annotators to label.

Notation. In the remaining notation, all definitions of objects are given with respect to the current round. Here we omit subscripts r and how objects change between rounds. In the current round, the set of annotated examples $\mathcal{D}$ contains n examples labeled by m annotators in total. $Y_{ij} \in [K]$ denotes the class annotator $\mathcal{A}_j$ chose for example $x_i \in \mathcal{D}$, with $Y_{ij} = \emptyset$ if annotator $\mathcal{A}_j$ did not label example i. $\mathcal{Y}_i$ is the set of collected labels for example x_i, with $|\mathcal{Y}_i| = 0$ if $x_i \in \mathcal{U}$. $\mathcal{I}_j$ is the subset of examples labeled by annotator $\mathcal{A}_j$, and $\mathcal{J}_i$ is the subset of annotators that labeled x_i.

2.1 ACTIVELAB

ActiveLab extends the CROWDLAB estimator of Goh et al. (2022). Some equations in this paper overlap with CROWDLAB, but we present them for completeness. Not every CROWDLAB equation is motivated here, curious readers can refer to detailed explanations by Goh et al. (2022).

Unlike ActiveLab, which is intended for guiding collection of additional labels, CROWDLAB is intended for analyzing a static dataset labeled by multiple annotators. Empirically it performs poorly when used for active learning. While both approaches estimate consensus labels in a similar fashion, they score examples differently. CROWDLAB estimates the likelihood that each current consensus label is *correct* or not, whereas ActiveLab estimates the utility of collecting *another* label to further improve the consensus and model trained therewith. CROWDLAB assigns very low scores to examples annotated by many labelers that heavily disagree, but even though their consensus label is unreliable, ActiveLab recognizes there is less utility in collecting one more label for such fundamentally difficult examples (vs. examples that currently have fewer annotations). Unlike CROWDLAB, ActiveLab also scores examples which currently have not been labeled yet. It must trade-off the potential information gain from collecting the 1st label for an example from $\mathcal{U}$ vs. the jth label for an example already labeled $j-1$ times.

We first describe how ActiveLab computes the score s_i for examples that have at least one annotation. Both CROWDLAB and ActiveLab are straightforward weighted ensembles which linearly combine multiple predictors to form a single estimate of class probabilities. In prediction competitions, such ensembles are often more accurate and better calibrated. One of these predictors is the (out-of-sample

predictions from a) trained classifier $\mathcal{M}$, abbreviated as $\widehat{p}_{\mathcal{M},i,k} := \widehat{p}_{\mathcal{M}}(Y_i = k \mid X = x_i)$. The other predictors are the annotators who previously labeled x_i. From the label Y_{ij} chosen by annotator $\mathcal{A}_j$, we form an annotator-estimated class probability vector $\widehat{p}_{\mathcal{A}_j,i,k} \approx p(Y_i = k \mid Y_{ij})$ that is directly comparable to the classifier predicted class probabilities (details further below). ActiveLab and CROWDLAB take a weighted average of this collection of probabilistic predictions to form a single vector of ensemble predicted class probabilities for each x_i.

CROWDLAB subsequently selects the most likely class under this ensemble estimate as the consensus label $\widehat{Y}_i$ representing our best guess of the true label Y_i. In Step 2 of each active learning round, we use CROWDLAB to estimate a single *consensus label* $\widehat{Y}_i$ that aggregates the available annotations $\mathcal{Y}_i$ for each example $x_i \in \mathcal{D}$. Subsequently in Step 5, ActiveLab scores $x_i \in \mathcal{D}$ via the likelihood that class $\widehat{Y}_i$ is correct under its ensemble estimate, expressed as:

$$\text{If } x_i \in \mathcal{D}: \quad s_i = \frac{w_{\mathcal{M}} \cdot \widehat{p}_{\mathcal{M},i,\widehat{Y}_i} + w_{\bar{\mathcal{A}}} \cdot \frac{1}{K} + \sum_{j \in \mathcal{J}_i} w_j \cdot \widehat{p}_{\mathcal{A}_j,i,\widehat{Y}_i}}{w_{\mathcal{M}} + w_{\bar{\mathcal{A}}} + \sum_{j \in \mathcal{J}_i} w_j} \tag{1}$$

$$\text{If } x_i \in \mathcal{U}: \quad s_i = \frac{w_{\mathcal{M}} \cdot \max_k \widehat{p}_{\mathcal{M},i,k} + w_{\bar{\mathcal{A}}} \cdot \frac{1}{K}}{w_{\mathcal{M}} + w_{\bar{\mathcal{A}}}} \tag{2}$$

The above estimates depend on $w_{\mathcal{M}}, w_j$ which determine *how much* to weigh the model $\mathcal{M}$ and each annotator $\mathcal{A}_j$. We estimate their relative trustworthiness (based on the observed annotations $\{Y_{ij}\}$) in order to select these weights, via the same procedure as CROWDLAB (details further below). Intuitively our estimate should down-weigh untrustworthy annotators or a poorly trained classifier, see Goh et al. (2022) for further discussion on this estimate's robustness against bad annotators/models. Unlike CROWDLAB, equation (1) also contains a uniform $1/K$ predictor that receives weight $w_{\bar{\mathcal{A}}} := \frac{1}{m}\sum_{j=1}^{m} w_j$, representing the weight assigned to our average annotator (across all examples).

Here is a fundamental difference between ActiveLab and CROWDLAB: under the former, the estimated likelihood that $\widehat{Y}_i$ is the correct class for $x_i \in \mathcal{D}$ is much lower (closer to uniform) for examples with few annotations. This regularization has smaller effect on examples with many annotations. Thus amongst the $x \in \mathcal{D}$, ActiveLab naturally favors acquiring labels for examples that currently have fewer annotations. ActiveLab also favors examples where annotators disagree with the consensus (note $\widehat{p}_{\mathcal{A}_j,i,\widehat{Y}_i}$ is much smaller if $Y_{ij} \neq \widehat{Y}_i$) or the classifier predicts the consensus to be unlikely. These are the $x_i \in \mathcal{D}$ whose current consensus label may be wrong, warranting re-labeling to determine whether a better label can be established.

Scoring examples from the unlabeled pool. Before delving into the details of $w_{\mathcal{M}}, w_j$, and $\widehat{p}_{\mathcal{A}_j}$, we describe how ActiveLab scores $x_i \in \mathcal{U}$. This is detailed in equation (2). Since we have no annotations for $x_i \in \mathcal{U}$, ActiveLab scores such examples only using the probabilistic predictions from our classifier $\widehat{p}_{\mathcal{M}}$. Many traditional active learning methods also operate this way (Munro, 2021). As seen in (2), the score s_i for $x_i \in \mathcal{U}$ is similarly computed as for $x_i \in \mathcal{D}$, except for modifications required to handle missing information. Since $\mathcal{J}_i = \emptyset$ in this case, we simply drop the annotator-predictors $\widehat{p}_{\mathcal{A}_j}$ from the weighted ensemble in order to obtain its estimate for unlabeled examples. And we simply take $\widehat{Y}_i = \arg\max_k \widehat{p}_{\mathcal{M},i,k}$, the class predicted by our classifier, since CROWDLAB cannot estimate a consensus label for $x_i \in \mathcal{U}$. Amongst the unlabeled examples, ActiveLab thus favors acquiring labels for those x_i for which the classifier is least confident (Munro, 2021).

Details for estimating weights and annotator likelihood. ActiveLab estimates $w_{\mathcal{M}}, w_j$, and $\widehat{p}_{\mathcal{A}_j}$ in the same fashion as CROWDLAB. We present the mathematical details here but refer readers to the explanations/motivations articulated by Goh et al. (2022). In equation (1), $\widehat{p}_{\mathcal{A}_j} \in \mathbb{R}^k$ is an "annotator likelihood" vector containing the probabilities that x_i belongs to each class given that annotator $\mathcal{A}_j$ chose the label Y_{ij}. It is very simply defined:

$$\widehat{p}_{\mathcal{A}_j,i,k} \approx p(Y_i = k \mid Y_{ij}) := \begin{cases} P & \text{when } Y_{ij} = k \\ \frac{1-P}{K-1} & \text{when } Y_{ij} \neq k \end{cases}$$

$P \geq 0$ is a global parameter shared across all annotators, estimated by computing the average annotator agreement across all examples that have more than one annotation. P reflects the probability that an annotator would select the consensus label for some arbitrary example (Goh et al., 2022).

The weights $w_{\mathcal{M}}, w_j$ in equation (1) estimate the trustworthiness of our classifier model and each annotator. The model weight is defined in terms of the normalized accuracy of the classifier's predictions with respect to the consensus label, over the subset of examples with more than one annotation. The weight w_j for annotator $\mathcal{A}_j$ is defined in terms of how much labels chosen by $\mathcal{A}_j$ agree with other annotators when they labeled the same examples as $\mathcal{A}_j$. More formally:

$$w_j := 1 - \frac{1 - g_j}{1 - A_{\text{MLC}}}, \qquad w_{\mathcal{M}} := \left(1 - \frac{1 - A_{\mathcal{M}}}{1 - A_{\text{MLC}}}\right) \cdot \sqrt{\frac{1}{n} \sum_{i \in \mathcal{D}} |\mathcal{J}_i|}$$

where g_j is the agreement between $\mathcal{A}_j$ and other annotators: $g_j := \dfrac{\sum_{i \in \mathcal{I}_j} \sum_{\ell \in \mathcal{J}_i, \ell \neq j} \mathbb{1}(Y_{ij} = Y_{i\ell})}{\sum_{i \in \mathcal{I}_j} (|\mathcal{J}_i| - 1)}$

$A_{\mathcal{M}}$ is the empirical accuracy of classifier model predictions with respect to the consensus labels:

$$A_{\mathcal{M}} := \frac{1}{|\mathcal{I}_+|} \sum_{i \in \mathcal{I}_+} \mathbb{1}\left(\widehat{Y}_i = \arg\max_k \widehat{p}_{\mathcal{M},i,k}\right) \tag{3}$$

Normalization factor A_{MLC} is the baseline accuracy (with respect to consensus labels) achieved by predicting the overall most labeled class Y_{MLC} (amongst all annotations for the dataset) for every example.

$$A_{\text{MLC}} := \frac{1}{|\mathcal{I}_+|} \sum_{i \in \mathcal{I}_+} \mathbb{1}(Y_{\text{MLC}} = \widehat{Y}_i) \tag{4}$$

To avoid bias (Goh et al., 2022), the accuracy estimates which determine P, w_j, and $w_{\mathcal{M}}$ are computed over the labeled examples that received more than one annotation: $\mathcal{I}_+ := \{i \in \mathcal{D} : |\mathcal{J}_i| > 1\}$.

2.2 Calibration of Classifier Predictions

While cross-validation enables us to produce out-of-sample predictions for each $x_i \in \mathcal{D}$, some types of models tend to nonetheless output overconfident predictions (Guo et al., 2017). Our active learning methods rely on the classifier to determine what data to label next and subsequently retrain another version of this same classifier. To mitigate overconfidence (or underconfidence), we *calibrate* the classifier's predicted class probabilities in Step 4 of each active learning round, before we compute ActiveLab scores via equation (1). We perform this calibration against the empirical distribution of the annotators' labels $\mathcal{Y}_i$ for each example in $\mathcal{D}$. Calibration is done by temperature scaling (Guo et al., 2017) the classifier's predicted probabilities $\widehat{p}_{\mathcal{M}}(Y_i \mid X_i)$ to minimize their (soft) cross entropy against the empirical distribution $\widehat{p}_{\text{emp}}$ of classes in $\mathcal{Y}_i$. We choose the temperature T to maximize:

$$\sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \widehat{p}_{\text{emp}}(Y_i = k \mid \{Y_{ij}\}_{j \in \mathcal{J}_i}) \cdot \log \widehat{p}^{(T)}_{\mathcal{M},i,k} \quad \text{where } \widehat{p}^{(T)}_{\mathcal{M},i,k} = \text{softmax}\left(\frac{\widehat{p}_{\mathcal{M},i,k}}{T}\right)$$

Subsequently, we calibrate the predictions for all examples in both $\mathcal{D}$ and $\mathcal{U}$ and compute ActiveLab scores using $\widehat{p}^{(T)}_{\mathcal{M},i,k}$ in place of $\widehat{p}_{\mathcal{M},i,k}$. Empirically, this calibration step improved a variety of active learning methods, allowing them to more robustly improve the accuracy of various types of models.

2.3 ActiveLab (Ensemble)

Ensemble methods aggregate outputs from multiple models into a single set of predictions that can be more accurate than any of the constituent models (Dietterich, 2000). Model ensembles are also popular in active learning; disagreeing predictions between models indicate areas of high epistemic uncertainty where annotating more data can greatly improve at least one of the constituent models (Seung et al., 1992). Here we present a straightforward extension of ActiveLab to ensemble settings.

Assuming there are L trained models in an ensemble, let $\widehat{p}_{\mathcal{M}_\ell}(Y_i \mid X_i)$ denote the class probabilities for x_i predicted by model $\mathcal{M}_\ell$ for $\ell = 1, 2, ..., L$. Here we can apply ActiveLab similarly as in the single-model case, but now allowing each model to have its own weight $w_{\mathcal{M}_1}, w_{\mathcal{M}_2}, ..., w_{\mathcal{M}_L}$ used

for averaging estimates. We use the following ActiveLab scores in ensemble settings:

$$\text{If } x_i \in \mathcal{D}: \quad s_i = \frac{w_{\bar{\mathcal{A}}} \cdot \frac{1}{K} + \sum_{\ell=1}^{L} w_{\mathcal{M}_\ell} \cdot \widehat{p}_{\mathcal{M}_\ell, i, \widehat{Y}_i} + \sum_{j \in \mathcal{J}_i} w_j \cdot \widehat{p}_{\mathcal{A}_j, i, \widehat{Y}_i}}{w_{\bar{\mathcal{A}}} + \sum_{\ell=1}^{L} w_{\mathcal{M}_\ell} + \sum_{j \in \mathcal{J}_i} w_j} \tag{5}$$

$$\text{If } x_i \in \mathcal{U}: \quad s_i = \frac{w_{\bar{\mathcal{A}}} \cdot \frac{1}{K} + \sum_{\ell=1}^{L} w_{\mathcal{M}_\ell} \cdot \widehat{p}_{\mathcal{M}_\ell, i, \widetilde{Y}_i}}{w_{\bar{\mathcal{A}}} + \sum_{\ell=1}^{L} w_{\mathcal{M}_\ell}} \tag{6}$$

Above the annotator weights $w_j, w_{\bar{\mathcal{A}}}$ and likelihoods $\widehat{p}_{\mathcal{A}_j}$ have the same definitions as in ActiveLab with a single model. Here consensus labels $\widehat{Y}_i$ are estimated from an similar ensemble extension of CROWDLAB, in which we propose to set $w_{\bar{\mathcal{A}}} = 0$ in equation (5) and identify which class $\widehat{Y}_i \in [K]$ maximizes the expression. Equation (6) shows we handle examples from the unlabeled pool in the same fashion as in the single-model case. For each $x_i \in \mathcal{U}$, we obtain a predicted class $\widetilde{Y}_i \in [K]$ from the ensemble classifier and treat $\widetilde{Y}_i$ as a proxy for its consensus label.

The weights $w_{\mathcal{M}_\ell}$ for each model are computed the same way as in ActiveLab with a single model. Each model's prediction accuracy with respect to consensus labels is again used to infer how trustworthy each model is relative to the annotators, with $A_{\mathcal{M}_\ell}$ and A_{MLC} defined as in (3) and (4).

$$w_{\mathcal{M}_\ell} := \left(1 - \frac{1 - A_{\mathcal{M}_\ell}}{1 - A_{\text{MLC}}}\right) \cdot \sqrt{\frac{1}{n} \sum_i |\mathcal{J}_i|}$$

To predict with our ensemble classifier after training, we can also take a weighted average of each model's predicted class probabilities using the same weights $w_{\mathcal{M}_\ell}$.

3 Experiments

Our subsequent experiments benchmark ActiveLab against many commonly used model/modality-agnostic methods for active learning and data re-labeling. These are described in Appendix A. In our experiments, each dataset is partitioned into train, test, and unlabeled pools. We have high-quality (i.e. ground truth) labels for the test set, which facilitates accurate evaluation of trained classifiers. No such ground-truth labels are available for the training set. Instead, all examples in the training set have been labeled by one or more (potentially noisy) annotators, and we consider this to be the dataset $\mathcal{D}$ for training an initial classifier, collected prior to active learning. At the outset, no labels are available for examples in the unlabeled pool. The train/test/unlabeled pools and the initial training annotations are identical across all runs/methods evaluated for the dataset. After training the model in Step 3 of each round of active learning, we evaluate its test accuracy against ground truth labels (only used for evaluation purposes). To acquire labels in Step 6, our experiments use a single new annotator to label the entire selected batch of data from a round of active learning.

Our main evaluation criterion is the test accuracy of classifier trained in each round of active learning. Each experiment (sequential active learning run) is repeated 5 times and we report the average model accuracy across the trials. We evaluate active learning methods on datasets of different modalities, training various classification models for these datasets to ensure our methods are model agnostic.

Wall Robot Navigation (Freire et al., 2009). This is a tabular dataset with 4 classes corresponding to directions a robot should navigate which are to be predicted from its sensor measurements. The initial train set for this dataset contains 500 examples, the unlabeled pool contains 1500 examples, and the test set used to measure the model accuracy contains 1000 examples. In each round of active learning between model training runs, we collect additional labels for the 100 examples with the lowest active learning scores from a single new annotator. We simulate imperfect annotators for this dataset. Some of these 100 examples may already have been previously labeled by other annotators and some may not have been labeled at all yet.

We consider 3 types of classifier models: Extremely Randomized Trees (Extra Trees) (Geurts et al., 2006), which was the most accurate model from the `sklearn` package on this dataset, fully-connected neural networks (MLP), K-Nearest Neighbors, and an ensemble composed of all 3.

CIFAR-10H (Peterson et al., 2019). This image classification dataset offers many annotated labels for each image in the CIFAR-10 test set, provided by different human annotators. Our experiment uses a subset of 1000 images as the initial training set, 4000 images in the unlabeled pool, and 5000 images in the test set. Our high-quality test set labels to measure model accuracy are those from the original CIFAR-10 dataset (Krizhevsky & Hinton, 2009), as Northcutt et al. (2021a) found the CIFAR-10 labels contain few errors. In each round of active learning, we collect additional labels from one new human annotator for the 500 images with the lowest scores s_i. We use an Imagenet-pretrained ResNet-18 classifier for single-model active learning. For ensemble-model active learning, our ensemble consists of three classifiers: ResNet-18, ResNet-34 and ResNet-50 (He et al., 2016).

4 RESULTS

Figures 1, 2 and S3 illustrate that ActiveLab significantly outperforms the other active learning methods in both the single-model and ensemble setting. These findings demonstrate that ActiveLab effectively selects examples to label and re-label in data of various modalities modeled with different types of classifiers. Unsurprisingly, active learning with ensemble models can produce higher accuracy than achieved with single models. Although note that single model accuracy when collecting data with ActiveLab can attain comparable performance to the ensemble models, especially for strong single models like in Figure 1

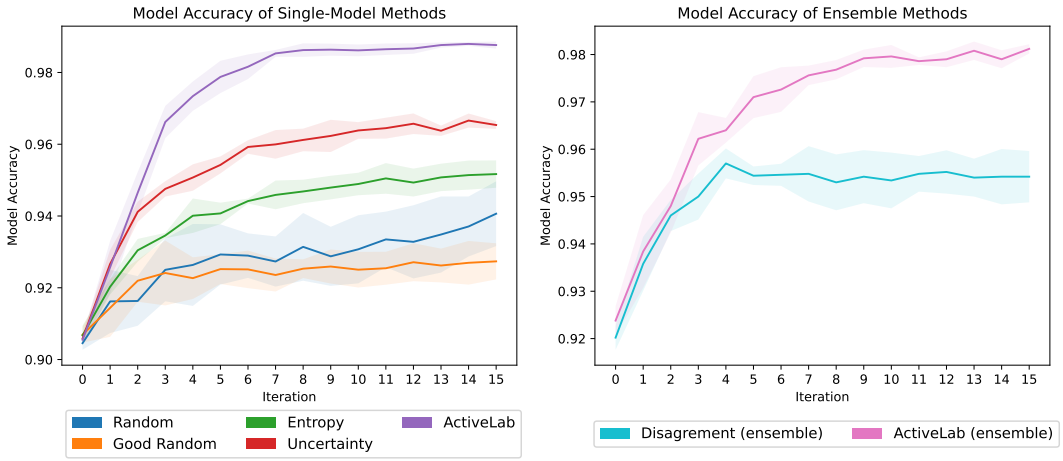

Figure 1: Evaluating active learning methods on the Wall Robot dataset to train an: ExtraTrees classifier (left) or ensemble of 3 models (right). Curves show test accuracy after each active learning iteration, averaged over 5 runs with the standard deviation in results shaded.

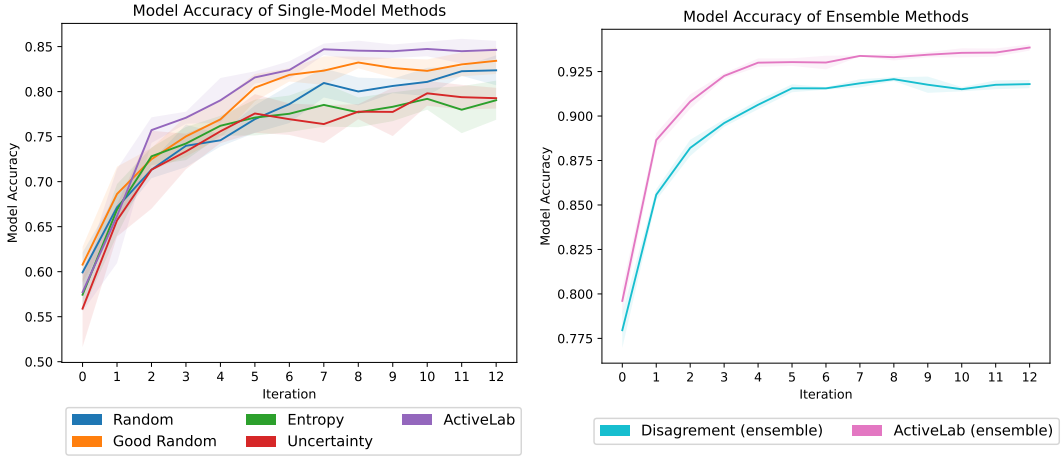

Figure 2: Evaluating active learning methods on CIFAR-10H to train a: ResNet-18 classifier (left) or ensemble of ResNet-18/34/50 models. Curves show the test accuracy after each iteration of active learning, averaged over 5 runs with the standard deviation in results shaded.

REFERENCES

Jason Baldridge and Miles Osborne. Active learning and the total cost of annotation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 9–16, 2004.

Mélanie Bernhardt, Daniel C Castro, Ryutaro Tanno, Anton Schwaighofer, Kerem C Tezcan, Miguel Monteiro, Shruthi Bannur, Matthew P Lungren, Aditya Nori, Ben Glocker, et al. Active label cleaning for improved dataset quality under resource constraints. *Nature communications*, 13(1): 1–11, 2022.

Derek Chen, Zhou Yu, and Samuel Bowman. Clean or annotate: How to spend a limited data collection budget. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pp. 152–168, 2022.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

Thomas G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, 2000.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. AutoGluon-Tabular: Robust and accurate AutoML for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

Ananda L. Freire, Guilherme A. Barreto, Marcus Veloso, and Antonio T. Varela. Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study. In *2009 6th Latin American Robotics Symposium (LARS 2009)*, pp. 1–6, 2009.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63:3–42, 2006.

RA Gilyazev and D Yu Turdakov. Active learning and crowdsourcing: A survey of optimization methods for data labeling. *Programming and Computer Software*, 44:476–491, 2018.

Hui Wen Goh, Ulyana Tkachenko, and Jonas Mueller. CROWDLAB: Supervised learning to infer consensus labels and quality scores for data with multiple annotators. *arXiv preprint arXiv:2210.06812*, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Sheng-Jun Huang, Jia-Lve Chen, Xin Mu, and Zhi-Hua Zhou. Cost-effective active learning from diverse labelers. In *IJCAI*, pp. 1879–1885, 2017.

David Martinez Iraola and Antonio Jimeno Yepes. Single versus multiple annotation for named entity recognition of mutations. *arXiv preprint arXiv:2101.07450*, 2021.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Johnson Kuan and Jonas Mueller. Model-agnostic label quality scoring to detect real-world label errors. In *ICML DataPerf Workshop*, 2022.

Christopher H Lin, Daniel S Weld, et al. To re (label), or not to re (label). In *Second AAAI conference on human computation and crowdsourcing*, 2014.

Christopher H Lin, M Mausam, and Daniel S Weld. Re-active learning: Active learning with relabeling. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Andrew McCallum, Kamal Nigam, et al. Employing em and pool-based active learning for text classification. In *ICML*, volume 98, pp. 350–358. Citeseer, 1998.

Rob Munro. *Human-in-the-loop machine learning*. Manning Publications, 2021.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Proceedings of the 35th Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*, December 2021a.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021b.

Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. Comparing Bayesian models of annotation. *Transactions of the Association for Computational Linguistics*, 6:571–585, 2018.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. Gaussian process classification and active learning with multiple annotators. In *International conference on machine learning*, pp. 433–441. PMLR, 2014.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, 1992.

Yan Yan, Glenn M Fung, Rómer Rosales, and Jennifer G Dy. Active learning from crowds. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 1161–1168, 2011.

Jie Yang, Thomas Drake, Andreas Damianou, and Yoelle Maarek. Leveraging crowdsourcing data for deep active learning an application: Learning intents in alexa. In *Proceedings of the 2018 World Wide Web Conference*, pp. 23–32, 2018.

Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B Chan. A comparative survey of deep active learning. *arXiv preprint arXiv:2203.13450*, 2022.

Liyue Zhao, Gita Sukthankar, and Rahul Sukthankar. Incremental relabeling for active learning with noisy crowdsourced annotations. In *IEEE third international conference on privacy, security, risk and trust*, pp. 728–733, 2011.

Yaling Zheng, Stephen Scott, and Kun Deng. Active learning from multiple noisy labelers with varied costs. In *2010 IEEE International Conference on Data Mining*, pp. 639–648. IEEE, 2010.

Appendix – ActiveLab: Active Learning with Re-Labeling by Multiple Annotators

A Related Work and Baseline Methods

While one can envision alternate methods that suggest which annotator should label which example (Huang et al., 2017), we find such a tightly-controlled setting too rigid for many applications. Step 6 is intentionally flexible. We also do not consider methods that can ask more than one annotator to review the same example within a round as such methods can be brittle (vs. our methods which collect at most one new label for each example in a round) (Baldridge & Osborne, 2004).

The most popular active learning methods are those like ActiveLab that can used with any classifier model for any data modality (Munro, 2021). While there has been extensive research on active learning (Zhan et al., 2022) and analyzing crowdsourced labels (Paun et al., 2018), few model/modality-agnostic active learning methods have been developed for settings with multiple annotators and data re-labeling (Lin et al., 2014). Many of the active learning methods proposed for such settings are specific to certain types of models or data types (Rodrigues et al., 2014; Zhao et al., 2011; Yan et al., 2011; Yang et al., 2018; Huang et al., 2017; Gilyazev & Turdakov, 2018; Iraola & Yepes, 2021). Other approaches like impact sampling (Lin et al., 2016) are too computationally expensive to run on problems like the image classification task in Section 3.

Our experiments benchmark ActiveLab against the following commonly used model/modality-agnostic methods for active learning and data re-labeling. Each method is applied in the same manner as ActiveLab to iteratively label a dataset, except which x_i are labeled is chosen via different s_i, and consensus labels for all $x_i \in \mathcal{D}$ are computed via majority-vote as used by Zheng et al. (2010).

Random. This method selects which examples to annotate entirely at random. It uses score: $s_i = x$ where $x \in [0, 1]$ is sampled uniformly at random and independently of i.

Good Random. This is a better variant of random selection that accounts for the number of annotations x_i already has: $s_i = x + |\mathcal{Y}_i|$ where $x \in [0, 1]$ is sampled uniformly at random. This pseudo-random selection prioritizes examples with the fewest number of labels collected thus far, a simpler variant of the approach of Chen et al. (2022). The unlabeled pool is labeled first prior to any re-labeling.

Entropy (Cohn et al., 1996). This method scores examples via the entropy of the model-predicted probabilities.

$$s_i = \sum_{k=1}^{K} \widehat{p}_{\mathcal{M},i,k} \cdot \log \widehat{p}_{\mathcal{M},i,k} \tag{7}$$

Uncertainty (Cohn et al., 1996). Measures how confident the model is in its predicted class: $s_i = \max_k \widehat{p}_{\mathcal{M},i,k}$.

Active Label Cleaning (Bernhardt et al., 2022). This approach was recently proposed for efficiently re-labeling an already-labeled dataset with multiple annotators. To select which data to collect an extra annotation for, Bernhardt et al. (2022) introduce a score that is a difference of two terms. The first term is the cross-entropy between the $\mathcal{M}$-predicted class probabilities and the empirical distribution of the annotators' labels for a particular example, and the second term is the entropy of the $\mathcal{M}$-predicted class probabilities.

$$\begin{aligned} s_i = &\sum_{k=1}^{K} \widehat{p}_{\mathcal{M},i,k} \cdot \log \widehat{p}_{\mathcal{M},i,k} \\ &- \sum_{k=1}^{K} \widehat{p}_{\text{emp}}(Y_i = k \mid \{Y_{ij}\}_{j \in \mathcal{J}_i}) \cdot \log \widehat{p}_{\mathcal{M},i,k} \end{aligned} \tag{8}$$

Disagreement (Ensemble) (Seung et al., 1992). Like ActiveLab (Ensemble), *disagreement* also employs an ensemble of multiple classifier models. This method measures the level of disagreement between different individual models' predictions. We employ a standard measure of disagreement for predicted class probabilities, where the score is defined as the total (soft) cross entropy between each model's predicted probabilities and the average estimate over all the models (McCallum et al., 1998).

$$s_i = -\frac{1}{L}\sum_{\ell=1}^{L}\sum_{k=1}^{K} \widehat{p}_{\mathcal{M}_\ell,i,k} \cdot \widehat{p}_{\bar{\mathcal{M}},i,k} \tag{9}$$

$$\text{where } \widehat{p}_{\bar{\mathcal{M}},i,k} = \frac{1}{L}\sum_{\ell=1}^{L} \widehat{p}_{\mathcal{M}_\ell,i,k}$$

To produce predictions from our ensemble classifier after running this method, we simply average the predictions from the individual models.

B TO LABEL OR RE-LABEL?

Since they are computed in a similar fashion, the s_i are directly comparable between $x_i \in \mathcal{D}$ vs. $\mathcal{U}$. ActiveLab thus naturally suggests when it is better to **re-label** an example from $\mathcal{D}$ vs. labeling a **new** example from $\mathcal{U}$. Cases when this might be true for some example $x_i \in \mathcal{D}$ include settings where: its annotations disagree (indicating that some annotators are noisy), or the model has atypically low confidence in its prediction (indicating x_i may be an outlier or high-influence datapoint whose label we should really get right), or the model confidently disagrees with the annotations. This last case is especially pertinent for examples x_i that only have a single annotation, where we may prefer to trust a confident prediction from a well-trained classifier over the given label which may be wrong (Northcutt et al., 2021a; Kuan & Mueller, 2022). Fixing labels for existing training data can improve a classifier more than noisily labeling additional data (Northcutt et al., 2021b; Iraola & Yepes, 2021). Section C empirically explores this. Mathematically, it is evident that ActiveLab will always prefer to label new examples from $\mathcal{U}$ if every annotation and the classifier (confidently) agree for all $x_i \in \mathcal{D}$.

When does ActiveLab prefer to re-label Consider x_i with a single annotation Y_{ij} and a different $x_\ell \in \mathcal{U}$, such that our classifier is equally confident in its predictions for both. In this case, deciding whether to re-label x_i vs. labeling x_ℓ specifically depends on: whether Y_{ij} matches the classifier's predicted class $\arg\max_k \widehat{p}_{\mathcal{M},i,k}$, and how much ActiveLab weights this annotator (w_j) vs. the average annotator ($w_{\bar{\mathcal{A}}}$) and the classifier ($w_{\mathcal{M}}$). If the classifier's prediction matches Y_{ij}, then ActiveLab will prefer to label x_ℓ. If the classifier disagrees with the annotation, then ActiveLab will prefer to re-label x_i whenever the CROWDLAB consensus label $\widehat{Y}_i \neq Y_{ij}$. This occurs if:

$$w_{\mathcal{M}}\big(\widehat{p}_{\mathcal{M},i,k^*} - \widehat{p}_{\mathcal{M},i,Y_{ij}}\big) > w_j\big(\widehat{p}_{\mathcal{A}_j,i,Y_{ij}} - \widehat{p}_{\mathcal{A}_j,i,k^*}\big)$$

where $k^* := \arg\max_k \widehat{p}_{\mathcal{M},i,k} \neq Y_{ij}$ in this example. The inequality is satisfied if: $w_{\mathcal{M}} \gg w_j$ (i.e. ActiveLab estimates the classifier is more trustworthy than annotator $\mathcal{A}_j$) and $\widehat{p}_{\mathcal{M},i,k^*} - \widehat{p}_{\mathcal{M},i,Y_{ij}} \gg \widehat{p}_{\mathcal{A}_j,i,Y_{ij}} - \widehat{p}_{\mathcal{A}_j,i,k^*}$ (i.e. the classifier predicts Y_{ij} is not the correct label confidently relative to the estimated accuracy of the data annotators).

C LABELING NEW EXAMPLES VS RE-LABELING IF WE HAVE INFINITE DATA

Traditional active learning only considers collecting at most one label per example and focuses entirely on the unlabeled pool rather than considering the option to re-label. If we have a huge unlabeled pool and a limited labeling budget, is there any utility in re-labeling? Our previous results clearly demonstrate the value of smart re-labeling when the size of $\mathcal{U}$ and labeling budgets are suitably matched. But with near-perfect annotators and an near-infinite unlabeled pool, re-labeling might not seem like a good idea (Lin et al., 2014). Thus we empirically investigate the question: At what degree of annotation-noise is there value in re-labeling when the size of $\mathcal{U}$ greatly exceeds our labeling budget?

We consider two settings: one where we only label *new* examples in each active learning round (*single label* case), and another where we can re-label examples if ActiveLab chooses to do so

(*multiannotator label* case). We run these approaches on a few variants of the Wall Robot Navigation dataset where we simulate annotators with different label noise rates. A higher noise rate annotator produces labels which are often wrong, while an annotator with noise rate 0 always selects labels that are correct. Similar to our previous Wall Robot benchmark, we conduct this experiment with an initial train set of 500 labeled examples, an unlabeled pool of 1500 examples, and test set of 1000 well-labeled examples. We label batches of 100 examples in each active learning round. Both *single label* and *multiannotator label* experiments start with the same labeled subset $\mathcal{D}$ (and always have the same annotator noise rates). In the *single label* experiment, active learning is done using the traditional entropy score only considering examples in $\mathcal{U}$. In the *multiannotator label* experiment, active learning is done via ActiveLab, which often selects a mixture of examples from $\mathcal{D}$ and $\mathcal{U}$ to collect an additional label for.

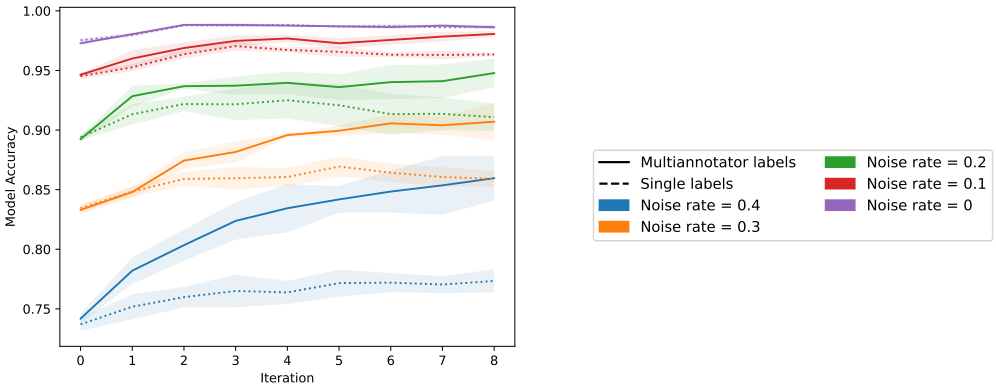

Figure S1: Comparing active learning methods that exclusively label new examples (*single labels*) vs. can also re-label examples instead (*multiannotator labels*), when annotators have different noise rates. Shown is the test accuracy of an ExtraTrees classifier trained on a certain number of total labels (corresponding to each iteration of active learning) for the Wall Robot Dataset. Curves are the average over 5 runs, and the standard deviation in results is shaded.

Figure S1 reveals that across all annotator noise levels, the model accuracy for the *mulitannotator labels* case is equal or better than for *single labels*. As expected, the difference in model accuracy between *single labels* and *mulitannotator labels* is larger when annotators are more noisy. This suggests it is rarely a bad idea to allow re-labeling if you have a method to do it adaptively like ActiveLab. It appears vital to re-label in settings with over 20% label noise. Our findings run contrary to the study of Lin et al. (2014), who acknowledged they were missing an effective active learning method with re-labeling at the time of their study.

D Active Label Cleaning

We also consider an *active label cleaning* setting, in which there are no additional unlabeled examples (Bernhardt et al., 2022). Here the goal is simply to selectively re-label the $x \in \mathcal{D}$ to establish the best consensus labels for this dataset. We evaluate various methods for this task using the *Wall Robot Complete* dataset described below. ActiveLab and other active learning methods are applied in the same manner as before, there is simply no unlabeled pool of examples to consider.

Wall Robot Complete. Similar to the Wall Robot Navigation tabular dataset, a key difference is that *Wall Robot Complete* has 2000 labeled examples in the initial training set, 1000 examples in the test set, and there is no unlabeled pool. As for *Wall Robot Navigation*, we collect additional labels for the 100 examples with the lowest active learning scores in each active learning round. Since all the examples already start out with some labels, this is a re-labeling (i.e. label cleaning) task, where we aim to obtain accurate consensus labels by having multiple annotators review the examples where this is necessary (Bernhardt et al., 2022).

Figure S2 shows that ActiveLab is also the best method for active label cleaning (re-labeling an already labeled dataset). It even outperforms the method Bernhardt et al. (2022) designed specifically for this setting. Unlike Bernhardt et al. (2022), ActiveLab estimates account for the number of annotations each example has and the quality of the annotators behind them. Existing active learning methods do not appear well-suited for such label cleaning tasks.

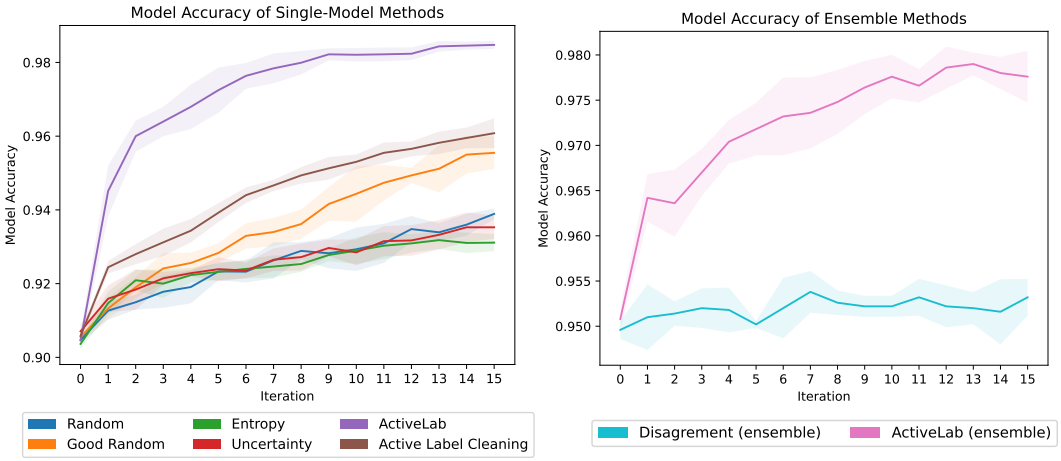

Figure S2: Evaluating active learning methods on the Wall Robot Complete dataset to train an: ExtraTrees classifier (left) or ensemble of 3 models (right). This is an active label cleaning task, with no unlabeled pool of examples. Curves show test accuracy after each iteration of re-labeling, averaged over 5 runs with the standard deviation shaded.

E Details of the Wall Robot Navigation Dataset

The original Wall-Following Robot Navigation dataset only has one label for each example. We adapt this dataset for our multi-annotator benchmark by simulating many different annotators to provide labels for requested examples. To simulate human annotators that make imperfect decisions (i.e. occasional labeling errors), we take the original set of labels from the Wall Robot dataset as ground truth labels. For each annotator, we add some random noise to their labels (noise rate = 0.15 for *Wall Robot Navigation* and noise rate = 0.2 for *Wall Robot Complete*), representing mislabeled examples. The randomly selected noisy annotations have an incorrect class (flipped probabilistically) that does not match the ground truth label. Using this method, we obtained 30 sets of labels, representing 30 annotators.

To setup the initial labeled and unlabeled pools $\mathcal{D}$ and $\mathcal{U}$, we completely dropped all the annotator labels for examples that begin unlabeled, while dropping a random fraction of the annotator labels for the examples in $\mathcal{D}$ that are labeled from the start, ensuring we keep at least one annotation for these examples. When collecting additional labels in each round of active learning, we simulate another annotator in the same fashion who labels the entire batch.

We also considered a second version of this benchmark with more heterogeneous annotators (including some very inaccurate outliers), and the results of the evaluation remained mostly the same as those presented here.

F EXPERIMENT DETAILS

In each round of active learning, we fit all models to $\mathcal{D}$ using 5-fold cross-validation. Additional details not mentioned here can be found in the code[2] for reproducing our experiments, as can the raw results of all active learning methods on all datasets.

For the experiments on the tabular Wall Robot dataset, we fit our classifier models using the `sklearn` package (Pedregosa et al., 2011). The models used were the: ExtraTreesClassifier with default hyperparameters, MLPClassifier with default hyperparameters except the max iteration set to 500 (to ensure convergence), and KNeighborsClassifier with default hyperparameters.

The image classifier models for our experiments on CIFAR-10H were fit using the AutoGluon AutoML package (Erickson et al., 2020) in order to avoid having to manually tune models and their optimization. We used various ResNet models initialized with default Imagenet-pretrained weights and then fine-tuned them on our dataset $\mathcal{D}$ (in a cross-valdated manner).

G ADDITIONAL RESULTS FOR WALL ROBOT

In addition to the Extra Trees model reported in Section 3, we repeat our single-model active learning experiments on the Wall Robot dataset using a Multilayer Perceptron (feedforward neural network) classifier. These additional results demonstrate that ActiveLab reliably produces larger improvement in model accuracy than other active learning methods, regardless which type of classifier model is being trained.

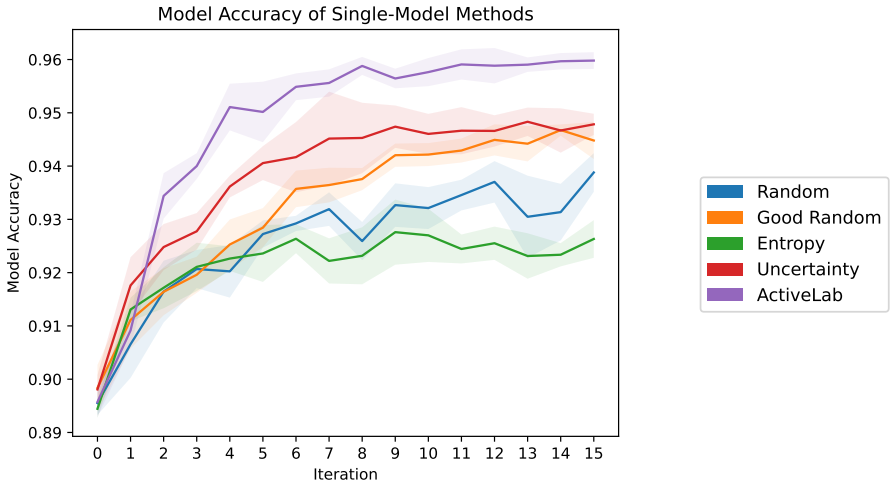

Figure S3: Evaluating active learning methods on the Wall Robot dataset to train a MLP classifier. Curves show the test accuracy after each active learning iteration, averaged over 5 runs with the standard deviation in results shaded.

[2] `https://github.com/cleanlab/multiannotator-benchmarks/tree/main/active_learning_benchmarks`

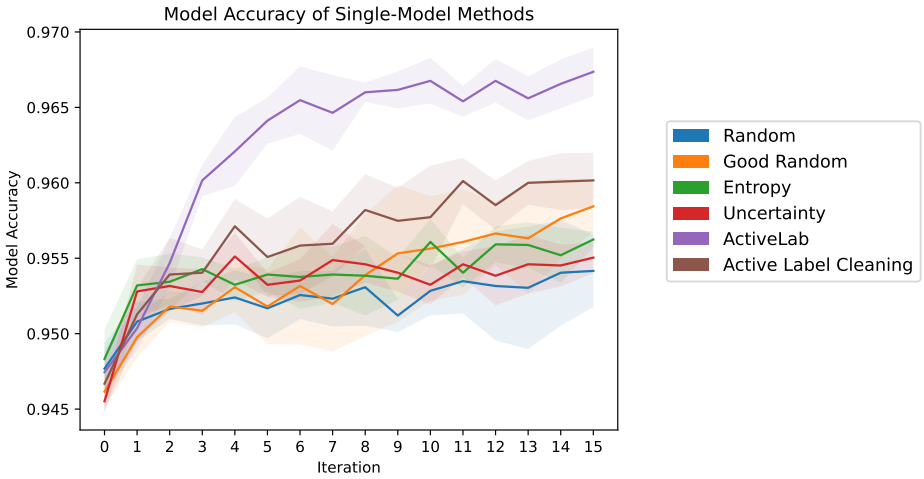

Figure S4: Evaluating active learning methods on the Wall Robot Complete dataset to train a MLP classifier. Curves show the test accuracy after each iteration of re-labeling, averaged over 5 runs with the standard deviation shaded.

H ACTIVE LEARNING IN SINGLE-LABEL SETTINGS

While ActiveLab is designed for scenarios where multiple annotators can label the same example, the method can also be applied for traditional active learning settings where we collect at most one label for each example. In this singly-labeled setting, we only score the $x_i \in \mathcal{U}$, as is common practice in pool-based active learning.

In such settings, we do not have data from multiple annotators to estimate the relative trustworthiness of the annotators and our model. Thus ActiveLab weights are undefined, but they are also not needed since we are only scoring unlabeled data without annotations in this setting. As a result, the natural ActiveLab score in such settings is:

$$s_i = \frac{\max_k \widehat{p}_{\mathcal{M},i,k} + \frac{1}{K}}{2} \quad \text{for } x_i \in \mathcal{U} \tag{10}$$

This is equivalent to simply relying on the confidence of the classifier, and thus equivalent to selecting examples via the aforementioned *Uncertainty* baseline method, a classic technique for active learning (Cohn et al., 1996; Munro, 2021).

In this singly-labeled setting, we provide a benchmark of this active learning method against two alternatives: randomly selecting examples to label, or using the entropy score. We use the same version of the Wall Robot Navigation dataset as our other benchmarks (with a noisy annotator). The initial train set contains 500 examples, and there are 1500 examples in the unlabeled pool. Each round, we select the 100 unlabeled examples with the lowest score to label and add them to the labeled subset. After training, model accuracy is similarly measured on a held-out test set of 1000 examples.

Figure S5 shows that ActiveLab exhibits comparable performance to the entropy baseline method in the singly-labeled setting. Both methods significantly outperform random selection of examples, even in this setting with noisy labels.

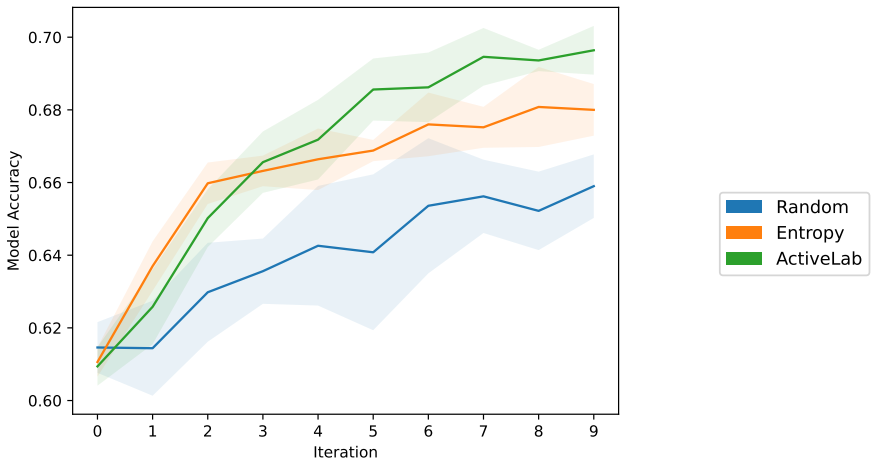

Figure S5: Evaluating active learning methods in the traditional singly-labeled setting on the Wall Robot dataset to train an ExtraTrees classifier. Curves show the test accuracy after each active learning iteration, averaged over 5 runs with the standard deviation shaded.