Fine-tuning Vision Classifiers On A Budget

Sunil Kumar
Groundlight.ai
Seattle, WA 98122
sunil@groundlight.ai

Ted Sandler
Groundlight.ai
Seattle, WA 98122
ted@groundlight.ai

Paulina Varshavskaya
Groundlight.ai
Seattle, WA 98122
paulina@groundlight.ai

Abstract

Fine-tuning modern computer vision models requires accurately labeled data for which the ground truth may not exist, but a set of multiple labels can be obtained from labelers of variable accuracy. We tie label quality to confidence derived from historical labeler accuracy using a simple naive-Bayes model. Imputing true labels in this way allows us to label more data on a fixed budget without compromising label or fine-tuning quality. We present experiments on a dataset of industrial images that demonstrates that our method, called Ground Truth Extension (GTX), enables fine-tuning ML models using fewer human labels.

1 Introduction

Fine-tuning modern ML models requires large quantities of high-quality, accurately labeled and relevant data. What counts as high quality? The two standard approaches for any specific dataset labeling task are either to engage trustworthy experts and treat their labels as ground truth, or crowd-source multiple labels of variable quality for each data point and aggregate them to impute trustworthy labels.

The quality of the aggregation result matters. A dataset with a significant presence of label noise from variable-quality human labels is problematic, especially for large models that can easily overfit to labeling errors. This is specifically true in computer vision where deep learning models have been shown to fit even random labels [1]. Learning in the presence of noisy labels has been addressed in the literature in a number of ways from direct dataset cleaning to regularization and ensemble methods [2]. The cost of label collection also matters. For a fixed labeling budget, minimizing the number of potentially noisy labels required to estimate the true label of a data point makes the labeling budget go further in terms of the total number of examples whose true labels we can infer.

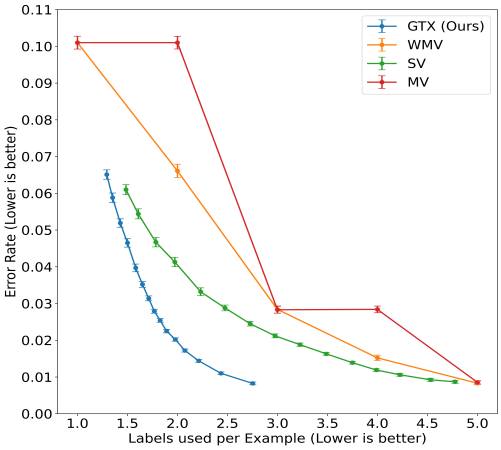


Figure 1: **Efficiency of Label Aggregation Methods.** The figure shows the error rate of various label aggregation methods as a function of the number of labels per example. GTX demonstrates superior efficiency by using fewer labels to achieve lower error rates. See Section 3.4 for more details.

So what is (ground) truth anyway? What constitutes a trustworthy or accurate label in the absence of expert opinion? In this paper, we tie label quality to a confidence estimated from historical labeler accuracy and show that using the one-coin naive-Bayes response model described in [3] to estimate

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).

this confidence allows us to label more data on a fixed budget without compromising label quality or the quality of the fine-tuned model. Our focus is on optimizing a finite labeling budget to get the most truth and best quality data for a more label-efficient fine-tuning. This focus is complementary to the active learning question of which examples to select for labeling [4].

The authors of [3] focus on the scenario in which no true labels are available, and therefore they utilize the EM algorithm to estimate both the true labels and the labeler accuracies. However, in many settings, one has access to some labeler accuracy information. For instance, it is often possible to collect at least a small number of expertly labeled examples which we call ground truth labels, and to evaluate the non-expert labelers on these. Or alternatively, we may have observed labelers' performances on previous labeling tasks for which ground truth labels were available. In these cases, we can run inference with the naive-Bayes model to infer the true labels (the E-step) and forego the M-step to re-estimate labeler accuracies. This is our method which we call Ground Truth Extension (GTX).

Therefore, in this paper, we consider a scenario where labelers come from a known and limited workforce cohort, such that we can estimate their individual labeling accuracies, but do not have the ability to route specific data points to particular labelers, as is common when using online data labeling services such as LabelBox[1] or in vertically integrated human-in-the-loop ML services such as Groundlight.ai. We also do not consider large-pool crowdsourcing marketplace-based services such as Amazon Mechanical Turk, where we may not have access to labeler accuracy statistics on prior labeling tasks.

We demonstrate across a variety of circumstances that the GTX method of imputing ground truth from multiple labels of variable quality, which we describe in section 2, outperforms standard and weighted majority voting in accuracy of imputed labels. We show this in a series of controlled synthetic data experiments (section 3), as well as in an industrial visual classification task where GTX-inferred labels are used to fine-tune a CNN vision model (section 4).

2 Ground Truth Extension

The GTX method is a framework for decision-making about label trustworthiness. It can facilitate more data to be labeled accurately on a constrained budget and consists of three parts:

1. A probabilistic model for estimating the true label and its confidence given noisy labels from non-expert labelers and estimates of their accuracies.
2. A method to estimate labeler accuracies from a comparatively small set of labels provided by experts – *ground truth labels*.
3. A strategy for determining which examples to (noisily) label in order to maximize the number of examples whose true labels can be inferred at a given level of confidence.

2.1 Naive-Bayes model of true labels

We use the one-coin naive-Bayes model from [3, 5, 6] to infer a true label estimate and our confidence in its correctness from a collection of noisy labels. This model assumes that each labeler acts independently and has a fixed, but unknown, level of accuracy that is independent of the true label. The label confidence is taken to be the probability of that label under the one-coin probabilistic model.

Let $y_i \in \{0, 1\}$ denote the true label of example i and let y_i^j be the label that labeler $j \in \mathcal{J}$ assigns to example i. Since not all labelers label each instance, we use $\mathcal{J}(i)$ to denote the set of labelers who labeled example i and $\mathcal{L}(i) = \{y_i^j \mid j \in \mathcal{J}(i)\}$ to be the set of noisy labels they provided.

In the model, the probability that a randomly selected example i has true label y is $\pi_y = P(Y_i = y)$, and the probability that labeler j assigns the correct label to example i is $\alpha_j = P(Y_i^j = y_i)$ for all i. The labelers' responses are assumed independent and therefore

$$P(\mathcal{L}(i) \mid Y_i = y) = \prod_{j \in \mathcal{J}(i)} \alpha_j^{\mathbf{I}[y_i^j = y]} (1 - \alpha_j)^{\mathbf{I}[y_i^j \neq y]}, \tag{1}$$

[1] https://docs.labelbox.com/docs/quality-analysis#view-and-assess-quality-analysis-performance

where $\mathbf{I}[\cdot]$ is the indicator function that equals 1 if its argument is true and 0 otherwise.

By Bayes rule, the probability that example i has true label y is

$$P(Y_i = y \mid \mathcal{L}(i)) \propto P(Y_i = y)\, P(\mathcal{L}(i) \mid Y_i = y) \tag{2}$$

$$= \pi_y \prod_{j \in \mathcal{J}(i)} \alpha_j^{\mathbf{I}[y_i^j = y]} (1 - \alpha_j)^{\mathbf{I}[y_i^j \neq y]}. \tag{3}$$

Of course, the values of parameters π_y and α_j are unknown and must be provided. In Section 2.2, we show how to estimate the α_j from a small set of expertly labeled examples. For π_y, we simply assume that each class has equal prior probability. Denoting the estimates as $\hat{\alpha}_j$ and $\hat{\pi}_y$, we can estimate the probability that example i has label y by plugging them into equation (3).

To predict a hard label for example i, we can select $\hat{y}_i = \arg\max_y \hat{P}(Y_i = y \mid \mathcal{L}(i))$ where $\hat{P}(\cdot)$ conveys that we are using the estimated parameters in equation (3). To predict soft labels, we can use the posterior probabilities from the model, $\hat{P}(Y_i = y \mid \mathcal{L}(i))$. We call the posterior probabilities the confidence scores for the hard, argmax labels.

2.2 Estimating labeler accuracies

Estimates of labeler accuracy may already be available from commercial labeling services. If no such data is available, one will need a small collection of known ground truth data denoted $\mathcal{D}_{\text{assess}}$ and have to dedicate a portion of the labeling budget to assessing labelers in order to estimate α_j.

In our work, we use the maximum likelihood estimate (MLE) to compute $\hat{\alpha}_j$ for each $j \in \mathcal{J}$ as the proportion of correctly labeled instances by labeler j in the assessment set, separately from the fixed labeling budget:

$$\hat{\alpha}_j = \frac{1}{|\mathcal{D}_{\text{assess}}|} \sum_{i \in \mathcal{D}_{\text{assess}}} \mathbf{I}[Y_i^j = y_i]. \tag{4}$$

2.3 Strategy for label collection

For practical applications under a limited labeling budget, we want to maximize the number of examples that are labeled without compromising quality. To balance this trade-off, we consider two strategies for collecting labels dynamically. In both cases, we assume that we cannot control which labelers label which examples, and that labelers are selected without replacement.

Strategy 1: Confidence threshold We collect labels on a given example until the computed confidence in the aggregate label exceeds a predefined confidence threshold τ, up to a maximum of κ labels per example. Practically, κ ensures that we do not exhaust the labeling budget on particularly difficult instances. This strategy is a good choice for streaming datasets where not all data points are available at once, or for extremely large datasets where the dataset size is larger than the labeling budget.

Strategy 2: Uncertainty sampling We use the active learning strategy of uncertainty sampling [7] to focus labeling effort on the most uncertain or ambiguous examples. Uncertainty u_i is computed based on the confidence in the aggregate label for each example i as follows:

$$u_i = 1 - \max_y \hat{P}(Y_i = y \mid \mathcal{L}(i)). \tag{5}$$

After each label, we update u_i and select the example about which we are most unsure next. We repeat this process until the entire labeling budget is spent. Prior to collecting the first label, each example's uncertainty is identical and maximum, such that we always collect a single label for all examples first before requesting any additional labels. This strategy is a good choice for traditional datasets that are smaller than the labeling budget.

We report results from applying each strategy in section 3.

3 Synthetic Data Experiments

In order to evaluate the effectiveness of the GTX method under controlled conditions, we construct a simulation framework to model possible datasets and labelers.

3.1 Label aggregation methods

We simulate labeler behavior with the probabilistic model described in section 2.1 under varying conditions, and compare to three common label aggregation strategies:

Majority voting (MV) The standard approach where each labeler has an equal vote, and the label with the majority of votes is selected. Confidence in the aggregate label is calculated as the proportion of votes for the selected label over the total number of votes.

Weighted majority voting (WMV) Each labeler's vote is weighted by their estimated accuracy $\hat{\alpha}_j$. Labelers with higher accuracies have more influence on the final decision. Confidence is calculated as the total weight supporting the selected label divided by the total sum of weights.

Soft voting (SV) Each labeler's vote is distributed between the two labels based on their estimated accuracy $\hat{\alpha}_j$. Specifically, labeler j contributes a weight of $\hat{\alpha}_j$ to the label they provided and a weight of $1 - \hat{\alpha}_j$ to the other label. The aggregated label is determined by selecting the label with the higher total weight. Confidence is calculated as the total weight supporting the selected label divided by the sum of weights for both labels.

3.2 Experimental setup

We design experiments to assess the performance of GTX versus the three baselines under different labeling budgets and labeler reliability conditions.

Labeler reliability We draw an α_j value for each labeler from a uniform distribution over the interval $[a, b]$: $\mathcal{U}(a, b)$. To account for variations in labeler skill and dataset difficulty, we model two different situations. In the first, labelers are more accurate and $\alpha_j \sim \mathcal{U}(0.8, 1.0)$. In the second, labelers are less accurate and $\alpha_j \sim \mathcal{U}(0.6, 0.9)$.

Labeling budget We report experimental results given the constraint of a fixed labeling budget of B labels collected. In the figures and tables below, sometimes we report results in terms of total labels collected, and elsewhere for clarity sometimes we report results as a function of the average number of labels per example $k = B/N$, where N is the number of labeled examples, which is typically not equal to the size of the dataset. In these cases the budget does not have to be allocated equally to each example.

Evaluation metrics We use two metrics to evaluate the performance of the label aggregation methods: 1) the error rate, or proportion of examples where the aggregate label $\hat{y}_i$ does not match the true label y_i: Err $= \frac{1}{N}\sum_{i=1}^{N} \mathbf{I}(\hat{y}_i \neq y_i)$; and 2) the mean absolute error (MAE), a measure of the average discrepancy between the true label and the method's confidence in that label: MAE $= \frac{1}{N}\sum_{i=1}^{N} \left| y_i - \hat{P}(y_i = 1 \mid \mathcal{L}(i)) \right|$. MAE penalizes aggregation methods that are both incorrect and overconfident in their predictions and rewards those providing calibrated confidence values.

3.3 GTX performance using confidence thresholds

First, we evaluate our method using the confidence threshold strategy described in section 2.3. We set a labeling budget of $B = 15000$ labels and initialize ten labelers. We construct a dataset larger than our budget to ensure that we are not data limited. We vary confidence thresholds with a range between 0.85 and 0.99 and a $\kappa = 5$ for both GTX and SV. For MV and WMV, which give 1.0 confidence after a single label, we collect a constant number of labels for each example, and vary this between 1 and 5. We repeat each experiment 100 times, each time initializing a new dataset and new labelers. For each method, we report results for the confidence threshold that minimizes error (Table 1).

More Accurate Labelers					
Method	Average k	Best τ	Labeled Examples (N)	Error Rate	MAE
MV	5	—	3,000 $\pm$ 0.0	0.86% ($\pm$ 0.04%)	10.0% ($\pm$ 0.17%)
WMV	5	—	3,000 $\pm$ 0.0	**0.83% ($\pm$ 0.04%)**	9.73% ($\pm$ 0.17%)
SV	4.78	0.99	3,140 $\pm$ 20.9	0.87% ($\pm$ 0.04%)	16.8% ($\pm$ 0.32%)
GTX	**2.75**	0.99	**4,517 $\pm$ 67.0**	**0.83% ($\pm$ 0.04%)**	**1.42% ($\pm$ 0.06%)**

(a) Comparison for more accurate labelers ($\alpha_j \sim \mathcal{U}(0.8, 1.0)$)

Less Accurate Labelers					
Method	Average k	Best τ	Labeled Examples (N)	Error Rate	MAE
MV	5	—	3,000 $\pm$ 0.0	10.4% ($\pm$ 0.30%)	25.1% ($\pm$ 0.30%)
WMV	5	—	3,000 $\pm$ 0.0	10.3% ($\pm$ 0.30%)	24.3% ($\pm$ 0.30%)
SV	5	0.91	3,000 $\pm$ 0.0	9.36% ($\pm$ 0.30%)	36.1% ($\pm$ 0.30%)
GTX	**4.07**	0.96	**3,690 $\pm$ 33.0**	**9.30% ($\pm$ 0.30%)**	**13.9% ($\pm$ 0.40%)**

(b) Comparison for less accurate labelers ($\alpha_j \sim \mathcal{U}(0.6, 0.9)$)

Table 1: **Comparison of methods using the confidence threshold strategy.** For each method, we report the confidence threshold/fixed number of labels per example that minimizes error over 100 simulations. Results are presented for (a) more accurate labelers and (b) less accurate labelers. See section 3.3 for details.

For more accurate labelers (Table 1a), we find that all methods were able to achieve similar error rates. However, our method only required $k = 2.75$ labels per example to achieve this performance, while the next most efficient option, SV, required $k = 4.78$. This efficiency allows us to label approximately 50% more data within the same budget. We find that our method is significantly more calibrated as well, with a MAE of 1.42%, while the alternatives report MAEs between 10% and 15%.

For less accurate labelers (Table 1b), we again observe that our method achieves a comparable error rate to the alternatives we consider. Our method requires only $k = 4.07$ labels per example compared to $k = 5$ for the other methods. This efficiency allows us to label approximately 23% more data within the same budget. Moreover, our method demonstrates significantly better calibration, reporting a MAE of 13.9%, whereas the alternatives have MAEs ranging from 24% to 36%. These results suggest that even when labelers are less reliable, our method still enables more data to be labeled within a fixed budget.

We plot our results across all trials for the more accurate labelers in Figure 1. This figure illustrates the relationship between efficiency and error rate, showing that GTX requires fewer labels per example to achieve similar error rates compared to the other options.

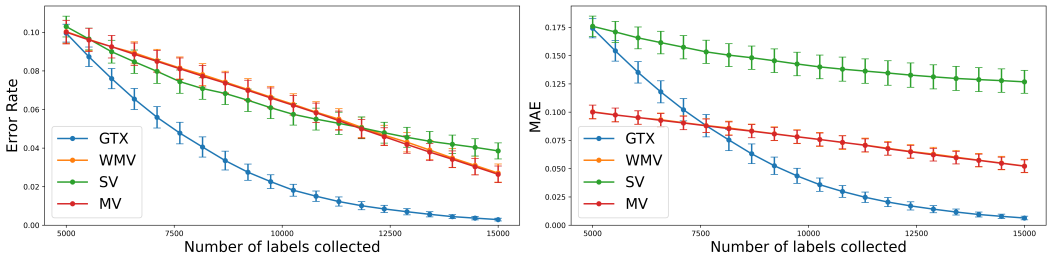
	More Accurate Labelers		Less Accurate Labelers	
Method	Error Rate	MAE	Error Rate	MAE
MV	2.64% ($\pm$ 0.43%)	5.22% ($\pm$ 0.55%)	17.9% ($\pm$ 0.95%)	22.2% ($\pm$ 0.96%)
WMV	2.70% ($\pm$ 0.47%)	5.24% ($\pm$ 0.59%)	17.4% ($\pm$ 0.80%)	22.0% ($\pm$ 0.84%)
SV	3.85% ($\pm$ 0.42%)	12.7% ($\pm$ 1.02%)	14.9% ($\pm$ 0.88%)	32.7% ($\pm$ 1.02%)
GTX	**0.29% ($\pm$ 0.07%)**	**0.64% ($\pm$ 0.16%)**	**9.46% ($\pm$ 0.80%)**	**16.8% ($\pm$ 1.25%)**

Table 2: **Comparison of methods using the uncertainty sampling strategy.** For each method, we report the results over 10 simulations. Results are presented for more accurate labelers ($\alpha_j \sim \mathcal{U}(0.8, 1.0)$) and less accurate labelers ($\alpha_j \sim \mathcal{U}(0.6, 0.9)$). See section 3.4 for details.

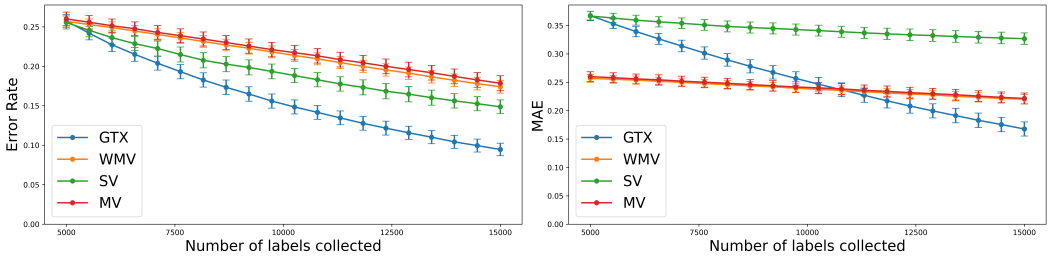
3.4 GTX performance using uncertainty sampling

We also evaluate our method using the uncertainty sampling strategy (described in section 2.3). We construct a dataset with 5000 examples and initialize 10 labelers. We set a labeling budget of $k = 3$ labels per example, or $B = 15000$ labels in total. We label this dataset with each method using uncertainty sampling, repeating each simulation 10 times. After each label is collected, we record the error rate and MAE across the entire dataset (Figure 2) in order to visualize the dynamics of each method. We only plot results after each example received an initial label, as the metrics we report are not very meaningful on unlabeled data. For both more accurate and less accurate labelers, we observe that GTX has meaningfully lower error rates on average than the most performant alternative throughout the entire iterative process.

With more accurate labelers, we report an average final error rate of 0.29% (Table 2). For most of the simulation, SV is the next best alternative. However, its performance trails off after 12,500 labels are collected, allowing MV and WMV to outperform it. MV concludes the simulation with the second best performance, an error of 2.64%. We notice that the error on this task is correlated with the final MAE. SV performs very poorly on MAE, and we conjecture that this hurts the ability of uncertainty sampling to select informative examples for labeling. With less accurate labelers, we observe that all methods perform worse but GTX continues to outperform the baselines.



(a) Results for more accurate labelers ($\alpha_j \sim \mathcal{U}(0.8, 1.0)$)



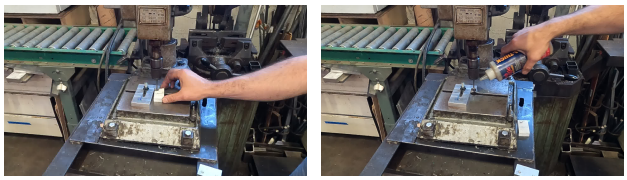
(b) Results for less accurate labelers ($\alpha_j \sim \mathcal{U}(0.6, 0.9)$)

Figure 2: **Dynamics of the uncertainty sampling strategy.** For each method, we report how error rate and MAE improve as we collect labels using uncertainty sampling over 10 trials. The dataset has 5000 examples. We plot data starting at 5000 labels collected, after each example is labeled once. The error bars report standard error. Results are presented for (a) more accurate labelers and (b) less accurate labelers. See section 3.4 for details.

4 Fine-tuning With Imputed Labels

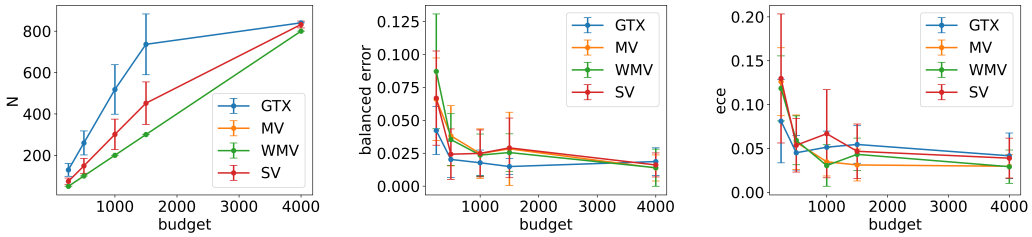
We demonstrate the performance of GTX in generating good imputed labels for fine-tuning through an experiment in image classification.

Dataset We collect a binary image classification dataset from a video taken in an industrial setting, where an operator is drilling holes in a wooden part. The dataset classifies individual video frames based on whether the operator adheres to best drilling practices as shown in figure 3. There is a total of 1,050 images in this dataset, which we make publicly available at `https://huggingface.co/datasets/sunildkumar/singlehole-part-tapping`.

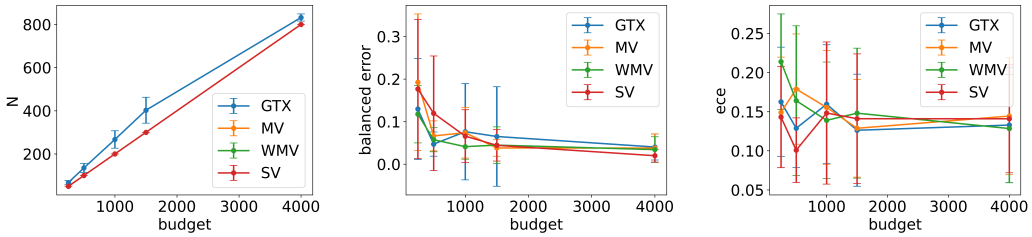


(a) class=YES (b) class=NO

Figure 3: Example YES and NO frames from the singlehole-part-tapping dataset for the question: *Is the wood block being drilled correctly? It must be touching the metal guide block on the left and the person must be holding it with his thumb and at least one additional finger.*



(a) Accurate labelers cohort results ($\alpha_j \sim \mathcal{U}(0.8, 1.0)$)



(b) Less accurate labelers cohort results ($\alpha_j \sim \mathcal{U}(0.6, 0.9)$)

Figure 4: Number of training set examples (leftmost column) and finetuning results as a function of labeling budget for GTX compared with the three baselines. We report the balanced error rate (middle column) and expected calibration error (rightmost column) on the withheld evaluation set.

Labeling We simulate two cohorts of labelers following the same process as in section 3: a high-quality cohort ($\alpha_j \sim \mathcal{U}(0.8, 1.0)$) and a lower-quality cohort ($\alpha_j \sim \mathcal{U}(0.6, 0.9)$). We reserve 10% of the images in the dataset for labeler assessments to estimate individual labeler accuracies, and use those estimates when applying label aggregation methods. We reserve an additional 10% of the dataset for evaluation. For different fixed labeling budget amounts, so long as there is remaining budget, we collect up to a maximum of $\kappa = 5$ labels for each image from these synthetic labelers, and we choose a confidence threshold of $\tau = 0.95$ for label aggregation methods requiring a threshold. Once the budget has run out, the set of N labeled images constitutes the training set for our fine-tuning experiments. We repeat the labeling and aggregation for each of the four methods MV, WMV, SV and GTX with 5 different random seeds and labeling budgets ranging from 250 to 4000 labels total.

Training We fine-tune a pretrained EfficientNet-b0 [8] for 3 epochs on each of the training datasets with imputed labels obtained by each of the four methods, and report its performance on the held out evaluation set. We repeat the training and evaluation 3 times for each of the 5 random seeds.

Figure 4 shows the results. We observe that GTX labels significantly more data that the baselines on smaller budgets and with higher-quality labelers. In these situations therefore the resulting models train on a larger dataset and see an increase in performance (lower balanced error) as well as lower expected calibration error. For the accurate labeler cohort, the repeated-measures Friedman's χ-square test on the balanced error reports that labeling methods are different (p-value=2.5e-09), and post-hoc Dunn pairwise analyses shows that GTX's balanced error is significantly lower than MV

(p-value=0.043) and WMV (p-value=0.038). In this set of experiments, however, we observe that increasing the labeling budget to the point where most of the available training data points can be labeled by majority vote, or reducing labeler quality to the point where more labels must be collected per example on average by GTX, erases its advantage (see also Appendix A). Further analysis and experiments are needed to determine the situations where GTX labeling would be more efficient specifically for fine-tuning.

5 Related Work

Many authors have studied the problem of training machine learning models on noisy labels [2, 9, 10], while others have studied the problem of detecting and pruning incorrect labels from a training set [11]. In a related line of work, authors have examined the problem of inferring accurate labels from unreliable, crowd-sourced ones [3, 12, 13, 14]. In our work, we apply methods from this latter line of research as a means of learning from unreliably labeled data. We also incorporate margin-based sampling from the active learning literature when determining which examples require additional labels and should be prioritized for labeling next [4].

6 Limitations, Extensions, and Conclusion

We have described the GTX method for collecting and using labels of variable quality in order to obtain a more accurately labeled dataset for fine-tuning ML models on a fixed labeling budget. Here, we discuss the limitations and possible extensions to this approach.

Labeler accuracies and assessment strategies A fundamental assumption of the GTX method is that we have access to estimates of individual labeler accuracies. Often those can be obtained for returning labelers in the form of benchmark assessments, a version of which we describe in section 2.2, or inter-annotator consensus statistics, based on performance on previous datasets and labeling tasks. However, if labeling accuracies are not available, some of the labeling budget will need to be spent on labeler quality assessments before we can benefit from the GTX approach. We have shown in section 3 that GTX achieves greater predicted label accuracy than the baselines for a fixed labeling budget, and so we argue that this would be a good investment of a portion of the budget. We leave it to future work to study strategies for selecting the number of assessments needed for a good estimate of labeler accuracy. Here confidence intervals or Bayesian credible intervals could be used.

Streaming data Our uncertainty sampling experiments assumed that all unlabeled examples in the dataset are immediately available to use. However, in some settings examples arrive sequentially in a stream (e.g., from a video camera). The GTX approach using a confidence threshold strategy is still directly applicable in such a streaming setting, but there would be a smaller pool of examples to solicit labels from for uncertainty sampling (since many examples would have yet to arrive). With a smaller pool of examples to pick from, we should expect the benefits of GTX to be less pronounced.

Modeling labeler error In this paper, our labeler error model is limited to a single accuracy estimate for each labeler. An obvious extension is to model labeler error with variable error rates based on example class which has been explored previously [3].

Fine-tuning on estimated labels Further work is needed to fully determine the conditions under which fine-tuning models specifically benefit from the labeling efficiency of GTX.

Conclusion We have proposed a method called Ground Truth Extension (GTX), which outperforms standard aggregation baselines for predicting the true label of an example with several available labels of variable quality. We have shown how to use existing ground truth to estimate individual labeler accuracies, how to decide whether to solicit more labels for a given data point, and how to find the best aggregate label based on a simple probabilistic model of labeler behavior. GTX generates more accurate labels than majority-vote baselines. And we show that these more accurate labels produce more accurate fine-tuned models on real image data when labelers have high accuracy. Future work can address method limitations related to assessing and modeling labeler error.

References

[1] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[2] Mengting Li and Chuang Zhu. Noisy label processing for classification: A survey, 2024.

[3] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

[4] Burr Settles. Active learning literature survey. 2009.

[5] Jungseul Ok, Sewoong Oh, Jinwoo Shin, and Yung Yi. Optimality of belief propagation for crowdsourced classification. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 535–544. JMLR.org, 2016.

[6] Ashish Khetan, Zachary C. Lipton, and Animashree Anandkumar. Learning from noisy singly-labeled data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[7] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. *CoRR*, abs/cmp-lg/9407020, 1994.

[8] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019.

[9] Moseli Mots'oehli and Kyungim Baek. Deep active learning in the presence of label noise: A survey, 2023.

[10] Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 1196–1204, 2013.

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

[12] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of machine learning research*, 11(4), 2010.

[13] Jafar Muhammadi, Hamid Reza Rabiee, and Abbas Hosseini. Crowd labeling: a survey. *arXiv preprint arXiv:1301.2774*, 2013.

[14] Victor S. Sheng and Jing Zhang. Machine learning with crowdsourcing: A brief summary of the past research and future directions. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 9837–9843. AAAI Press, 2019.

Supplementary Material

Appendix A Fine-tuning on imputed labels

We find that less accurate labelers erase the fine-tuning advantage of GTX over the baselines, as shown in figure 5.

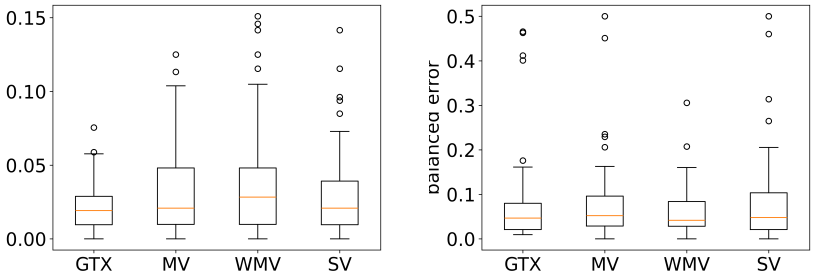


Figure 5: Boxplots of balanced error on the evaluation set after fine-tuning a pretrained EfficientNet-b0 on the singlehole-part-tapping image classification dataset: labels from the accurate labeler cohort (left) vs less accurate labeler cohort (right). Note different y scales.