Model-Agnostic Label Quality Scoring to Detect Real-World Label Errors

Johnson Kuan [1] **Jonas Mueller** [1]

Abstract

We consider algorithms to find wrongly labeled data, which lurks in many real-world applications and hampers training/evaluation of ML models. We present the first empirical study of various scoring methods for this task on real datasets with naturally-occurring label errors (as opposed to synthetically introduced label errors). The label quality scores considered here can be utilized with arbitrary classification models. We examine five popular image recognition models (and ensembles thereof) to comprehensively characterize how well different scores detect label errors in practice.

1. Introduction

While supervised machine learning (ML) assumes the labels we train/evaluate our model on are correct, this is often not the case in real-world datasets (Müller & Markert, 2019; Northcutt et al., 2021a; Kang et al., 2022). One report by Hivemind & Cloudfactory found that datasets labeled via third-party data annotation vendors contained between 7% to 80% percent label errors. To address this, ML researchers have studied methods for supervised *ML with data that is Noisily Labeled* (Song et al., 2022; Natarajan et al., 2013). The goal of this task, which we abbreviate as **MLwNL**, is training models on noisily labeled data to produce accurate predictions on new test examples.

While most MLwNL methods aim to achieve this via modified training objectives and other modeling tricks (Song et al., 2022; Sukhbaatar & Fergus, 2014; Jiang et al., 2018; Zhang & Sabuncu, 2018; Nishi et al., 2021), an alternative option is to first identify which examples are mislabeled, manually deal with these examples, and finally train models on the resulting cleaned dataset (Northcutt et al., 2021b). As advocated for in the practice of data-centric AI, this latter workflow improves the dataset independently of the models. A key challenge here is the task of *Label Error Detection* (**LED**) in which we aim to find which examples are labeled correctly vs not. Effective algorithms for LED should help human reviewers efficiently find the bad labels in a dataset.

While LED methods are intended for real-world applications with messy data, existing LED benchmarks have relied upon synthetically introduced label errors in which labels are randomly swapped with other labels (Brodley & Friedl, 1999; Müller & Markert, 2019; Northcutt et al., 2021b; Gu et al., 2021). However label errors in the wild can exhibit very different distributions (Jiang et al., 2020; Wei et al., 2022) and it is thus important to study LED with data and labels from real-world applications. To our knowledge, this paper presents the first real-world benchmark of label error detection methods. While the real-world datasets studied here have been previously used to benchmark ML methods for supervised learning with noisy labels, the goal there is to maximize predictive accuracy in the presence of bad labels. In contrast, we study methods for *directly identifying which examples are badly labeled*. Here performance is evaluated by how accurately each method identifies which examples are correctly labeled vs. not (via ground-truth labels available only during evaluation, but not while scoring the noisy given labels). Appendix A delves into reasons why LED is a task of independent interest from MLwNL.

With the shift towards more data-centric AI, LED will play a prominent role in future ML workflows. Despite this, little is known about which LED methods work best on real data with naturally-occuring label errors. All of the top-performing LED methods identified here are publicly available in the `cleanlab` package[1] to easily run on any dataset, as is the code[2] to reproduce our benchmarks.

2. Scoring Label Quality

This paper specifically focuses on classification datasets where each example is labeled with one of K classes. Given an example x, a trained classification model $h(\cdot)$ outputs predicted probabilities $\mathbf{p} = h(x) = [p_1, \ldots, p_K]$, where p_k estimates the probability that x belongs to class k. Because predictions may be overfit for examples from the classifier's

[1]Cleanlab. Correspondence to: Jonas M <jonas@cleanlab.ai>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

[1]https://github.com/cleanlab/cleanlab

[2]https://github.com/cleanlab/label-error-detection-benchmarks/

training set, we always employ out-of-sample $\mathbf{p}$ ensuring the classifier has not seen x during training. Throughout we obtain out-of-sample predictions for every example in a dataset by fitting each model via 5-fold cross-validation.

Here we consider various scores for evaluating how likely a particular example is labeled correctly (lower scores indicate labels more likely to be wrong). All scores studied here are entirely *model-agnostic* and can be computed with any classifier. For any given example, its label quality score solely depends on the predicted probabilities output by a trained classifier $\mathbf{p} \in \mathbb{R}^K$ and the given label $y \in \{1, \ldots, K\}$. In particular, we consider the following label quality scores:

Self-Confidence is the model's estimated probability that the example x belongs to the class of its given label y.

$$\text{Score}(x, y, \mathbf{p}) = p_y \quad \text{for } y \in \{1, \ldots, K\} \tag{1}$$

Self-confidence is a natural score based on the likelihood of the given labels under our model, which Müller & Markert (2019); Northcutt et al. (2021b) previously illustrated can uncover some label errors in real datasets.

Normalized-Margin is defined as the gap between our model's estimated probability of: the given label vs. the otherwise most likely class (that is not the given label).

$$\text{Score}(x, y, \mathbf{p}) = p_y - p_{k*} \quad \text{for } k^* = \underset{k \neq y \in \{1,\ldots,K\}}{\text{argmax}} \; p_k \tag{2}$$

Northcutt et al. (2021b) previously considered normalized-margin as an alternative score of label quality, finding it can be more effective than self-confidence in synthetic benchmarks where labels are randomly flipped to incorrect classes. Assuming our model outputs trustworthy predictions, then the smallest normalized-margin scores occur for examples where the true label has been replaced by another class in the given label. While such swapping of classes is the only type of label error in most existing synthetic LED benchmarks, we note that real-world datasets can exhibit other types of label errors such as out-of-distribution (OOD) examples for which no label in the set of classes really applies.

Confidence-weighted Entropy is a novel score we define as the ratio of the self-confidence and the (normalized) entropy H_K of the predicted probabilities over all classes.

$$\text{Score}(x, y, \mathbf{p}) = \frac{p_y}{H_K(\mathbf{p})}$$
$$\text{where } H_K(\mathbf{p}) = -\frac{1}{\log K} \sum_{k=1}^{K} p_k \cdot \log(p_k)$$

The motivation to still consider the likelihood of the given label being correct, as in self-confidence, but place greater emphasis on finding OOD examples. OOD examples may not belong to any of the specified classes {1,…,K}, and thus are expected to receive higher entropy predictions from a classifier capable of proper uncertainty estimation.

We also consider two scores which solely depend on the classifier prediction and not the given label. Originally intended for use in active learning as a way to find examples whose label the model is current least certain about, these scores have also been proposed for finding bad labels in an already-labeled dataset (Warmerdam, 2021; Munro, 2021).

(Negative) **Entropy** of the predicted probabilities $\mathbf{p}$.

$$\text{Score}(x, y, \mathbf{p}) = \sum_{k=1}^{K} p_k \cdot \log(p_k) \tag{3}$$

Least-Confidence is defined as the probability our model assigns to class it finds most likely for x.

$$\text{Score}(x, y, \mathbf{p}) = p_{k*} \quad \text{for } k^* = \underset{k \in \{1,\ldots,K\}}{\text{argmax}} \; p_k \tag{4}$$

Adjusted Scores. We can adjust label quality scores based on our model's propensity to predict certain classes:

$$\text{Adjusted-Score}(x, y, \mathbf{p}) = \text{Score}(x, y, \tilde{\mathbf{p}})$$
$$\text{where } \tilde{\mathbf{p}} \propto \mathbf{p} - \bar{\mathbf{p}} \text{ with } \bar{p}_k = \frac{1}{|X_k|} \sum_{x' \in X_k} h(x')$$

and X_k defined as the subset of examples whose given label is k. Adjusting predictions via the confidence threshold $\bar{\mathbf{p}}$ was proposed by Northcutt et al. (2021b) to obtain theoretical performance guarantees for particular LED methods. Here we re-normalize $\tilde{\mathbf{p}}$ to be a valid probability distribution. This adjustment also naturally accounts for class imbalance.

Ensemble Label Quality Scores. Although the aforementioned label-quality scores can be used with any model, their LED performance ultimately depends on the quality of the predictions output by the model. Model ensembling can improve predictions by training multiple diverse types of models and then aggregating their outputs. Given J different trained classifiers $h^{(1)}, \ldots, h^{(J)}$, a straightforward ensemble prediction can be computed as

$$\mathbf{p}_{\text{ens}} = h_{\text{ens}}(x) = \sum_{j=1}^{J} w_j \cdot h^{(j)}(x) = \sum_{j=1}^{J} w_j \cdot \mathbf{p}^{(j)} \tag{5}$$

where aggregation weights w_j typically sum to one (for example, a uniform average of the different models' predictions is commonly used). Once the predictions have been aggregated into a single $\mathbf{p}_{\text{ens}}$, this ensemble-prediction can be used with our label quality scores in a straightforward fashion. Here we alternatively consider aggregating label-quality scores computed with respect to each individual model, rather than aggregating their predictions and

then computing a single label-quality score. The resulting *ensemble label quality score* is given by:

$$\text{Score}\big(x, y, \{\mathbf{p}^{(j)}\}_{j=1}^{J}\big) = \sum_{j=1}^{J} w_j \cdot \text{Score}\big(x, y, \mathbf{p}^{(j)}\big) \quad (6)$$

Again, uniform aggregation weights may be used. Here we also consider non-uniform aggregation weights selected as: $w_j = exp(-T \cdot \ell_j)$ where ℓ_j denotes the log-loss between the given label y and jth model's prediction $\mathbf{p}^{(j)}$, and T is a hyperparameter (more details in Appendix E).

3. Experiments

Beyond the fact that image classification forms one of the cornerstone tasks of ML, our study is focused on image data in part due to the ease of visually verifying detected label errors for images, and because *true labels* happen to be available for some image datasets. Enabling quantitative evaluation of LED methods, these true labels are assumed to reflect the underlying ground truth class (eg. obtained from more diligent expert reviewing effort than the annotation process for assigning the labels given in the dataset).

Datasets. We consider the following image classification datasets with naturally-occurring label errors (Table S4):

Food-101n (Lee et al., 2018). We only consider the subset of 53k training images for which true labels are available.

Cifar-10n-worst, **Cifar-10n-agg** (Wei et al., 2022). The former contains more noisy labels than the latter.

For reference, we also include these datasets common in past studies of data-centric AI and label error detection:

Roman-numeral (Ng, 2021) with miscellaneous issues.

Cifar-10s (Northcutt et al., 2021b) with **s**ynthetic errors.

Models. To evaluate our scores across various models, we consider some of the most popular architectures from the `timm` library for image classification (Wightman, 2019): ResNet-18, ResNet-50d, EfficientNet-B1, Twins PCPVT, and the Swin Transformer. Table S3 lists the accuracy achieved by the trained classifier on each dataset.

Evaluation Metrics. As the broader purpose of label error detection is flagging suspicious examples for human review, we evaluate our LED methods using metrics from *information retrieval*. The following metrics only depend on the ranking of the scores rather than their absolute values:

AUROC for classifying whether each label is correct vs. not. This evaluates both the precision and recall of our scores.

For applications with large datasets, we believe high precision is more important than high recall. Without high precision, a human reviewer will not find the scores very effective for discovering label errors. Regardless what a LED algorithm outputs, a data analyst will be hard-pressed to manually review even 1% of a dataset whose sample size is in the millions. Our study measures precision at two particular levels via the following evaluation metrics:

Lift @ 100 measures how many times more prevalent label errors are in the top-100 scoring examples vs. entire dataset.

Lift @ #Errors is the lift at T instead of 100, for T defined as the true number of label errors in the dataset.

Results. Figures 1, S1, S2 illustrate that which method is best unsurprisingly depends on the types of label errors present in the data. The specific model used plays a smaller role. For any one score, the model's predictive accuracy significantly affects LED performance. The overall best LED performance is achieved using our most accurate model, the Swin Transformer, together with either the *confidence weighted entropy* or *self-confidence* scores (or normalized variants thereof). Tables 1, 2, S1 zoom in on this model to highlight how our label quality scores behave with effective models. The *least-confidence* and *entropy* scores perform poorly overall, confirming the intuition that good LED methods should account for the given label.

The overall best method to identify label errors utilizes an ensemble label quality score leveraging multiple models (Figures 2, S3, S4). Amongst ensembling options, aggregating the individual model's label quality scores is similar but occasionally better than: first aggregating their predictions and then computing a single label quality score. While a uniformly-weighted ensemble (which leverages the diversity across models) is sometimes the overall best approach, it underperforms just using the single best model on many datasets. In contrast, our weighted ensemble approach never significantly underperforms the best single model, while substantially outperforming it on the *Roman-numeral* dataset.

Adjustment of the scores does not reliably improve their performance, although may still be useful if we wish to grant equal consideration to label errors from different classes in an imbalanced dataset rather than simply finding the most label errors overall as evaluated here. We observe that *confidence weighted entropy* performs much better on datasets that contain more OOD examples vs. Cifar-10 where there are few such examples. Our overall results remain similar between *Cifar-10n-agg* and *Cifar-10n-worst* suggesting these findings generalize across different noise levels under a given systematic distribution of label errors. In contrast, the results are not similar between *Cifar-10n* (naturally occurring label errors) and *Cifar-10s* (synthetically introduced label errors), which demonstrates the importance of benchmarking LED with real label errors in the wild as opposed to the simulation studies conducted in the past.

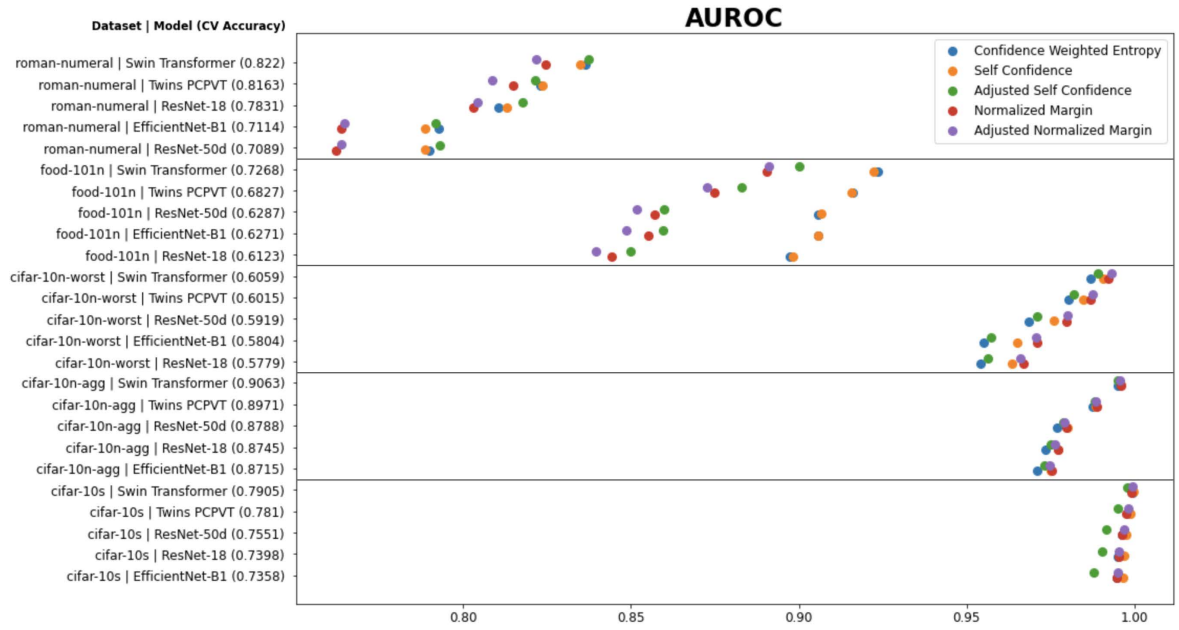

Figure 1. AUROC for LED achieved by label quality scores for each dataset and model. Models are ordered by accuracy on each dataset.

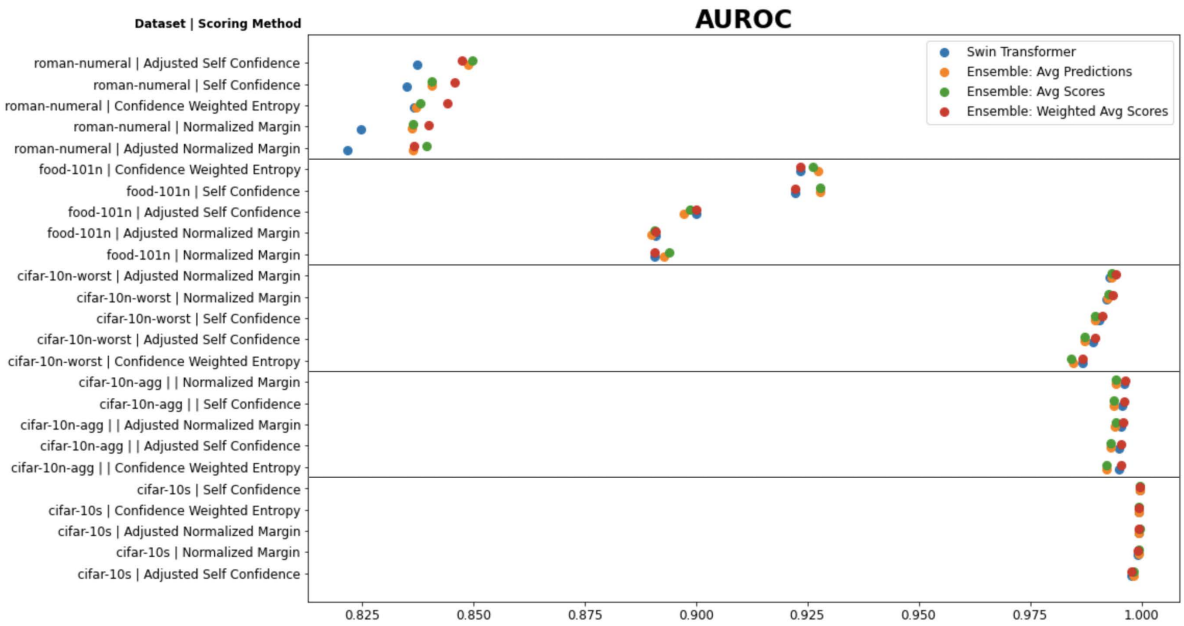

Figure 2. AUROC of label quality scores used with best single model (Swin Transformer) vs. ensembling strategies with multiple models.

Table 1. Lift @ 100 for LED with Swin Transformer model.

Label Quality Score	roman-numeral	food-101n	cifar-10n-worst	cifar-10n-agg	cifar-10s
Self Confidence	5.78	4.99	2.49	11.10	5.01
Normalized Margin	4.34	4.45	2.49	11.10	5.01
Adjusted Self Confidence	6.24	4.34	2.49	11.10	5.01
Adjusted Normalized Margin	4.34	4.07	2.49	11.10	5.01
Confidence Weighted Entropy	6.16	4.83	2.49	11.10	5.01
Entropy	4.11	5.37	1.84	5.77	0.10
Least Confidence	3.27	5.32	2.06	4.99	0.30

Table 2. Lift @ #Errors for LED with Swin Transformer model.

Label Quality Score	roman-numeral	food-101n	cifar-10n-worst	cifar-10n-agg	cifar-10s
Self Confidence	4.05	3.80	2.36	10.34	4.94
Normalized Margin	3.89	3.16	2.37	10.36	4.92
Adjusted Self Confidence	4.11	3.42	2.35	10.35	4.87
Adjusted Normalized Margin	3.85	3.33	2.38	10.37	4.92
Confidence Weighted Entropy	4.17	3.84	2.33	10.29	4.93
Entropy	3.01	2.81	1.28	3.59	0.52
Least Confidence	2.84	2.73	1.32	3.67	0.61

References

Brodley, C. E. and Friedl, M. A. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.

Chen, P., Liao, B. B., Chen, G., and Zhang, S. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, 2019.

Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., and Liu, Y. Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations*, 2020.

Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., and Shen, C. Twins: Revisiting the design of spatial attention in vision transformers. In *Advances in Neural Information Processing Systems*, 2021.

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., and Smola, A. AutoGluon-Tabular: Robust and accurate AutoML for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Gu, K., Masotto, X., Bachani, V., Lakshminarayanan, B., Nikodem, J., and Yin, D. An instance-dependent simulation framework for learning with label noise. *arXiv preprint arXiv:2107.11413*, 2021.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., and Li, M. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Hivemind and Cloudfactory. Crowd vs. managed team: A study on quality data processing at scale. URL https://go.cloudfactory.com/hubfs/02-Contents/3-Reports/Crowd-vs-Managed-Team-Hivemind-Study.pdf.

Huang, J., Qu, L., Jia, R., and Zhao, B. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, 2018.

Jiang, L., Huang, D., Liu, M., and Yang, W. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning*, 2020.

Kang, D., Arechiga, N., Pillai, S., Bailis, P., and Zaharia, M. Finding label and model errors in perception data with learned observation assertions. *arXiv preprint arXiv:2201.05797*, 2022.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Lee, K.-H., He, X., Zhang, L., and Yang, L. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

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

Müller, N. M. and Markert, K. Identifying mislabeled instances in classification datasets. In *International Joint Conference on Neural Networks*, 2019.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, 2013.

Ng, A. Data-centric AI competition. *DeepLearning.AI and Landing AI*, 2021. URL https://https-deeplearning-ai.github.io/data-centric-comp/.

Nishi, K., Ding, Y., Rich, A., and Hollerer, T. Augmentation strategies for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Northcutt, C. G., Athalye, A., and Mueller, J. 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.

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

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Pruthi, G., Liu, F., Kale, S., and Sundararajan, M. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 2020.

Reed, S. E., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. In *ICLR (Workshop)*, 2015.

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Sukhbaatar, S. and Fergus, R. Learning from noisy labels with deep neural networks. *arXiv preprint arXiv:1406.2080*, 2014.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

Warmerdam, V. D. Doubtlab helps you find bad labels, 2021. URL https://koaning.github.io/doubtlab/.

Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., and Liu, Y. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022.

Wightman, R. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Zhang, Y., Zheng, S., Wu, P., Goswami, M., and Chen, C. Learning with feature-dependent label noise: A progressive approach. In *International Conference on Learning Representations*, 2021.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, 2018.

Appendix:
Model-Agnostic Label Quality Scoring to Detect Real-World Label Errors

A. Why focus on Label Error Detection vs. ML with Noisy Labels?

There are numerous reasons to consider our LED task independently of MLwNL's focus on predictive performance (Brodley & Friedl, 1999; Müller & Markert, 2019; Northcutt et al., 2021b). Many methods for MLwNL are highly model-specific, relying on techniques like loss modification (Natarajan et al., 2013; Zhang & Sabuncu, 2018; Jiang et al., 2018), iterative training (Jiang et al., 2018; Han et al., 2018; Reed et al., 2015), or special bespoke architectures (Sukhbaatar & Fergus, 2014). In contrast, the LED methods considered here can be utilized with *any* classification model. Thus while particular MLwNL methods may or may not benefit from various advances in ML modeling (as the state-of-the-art marches relentlessly forward), all of our LED methods benefit directly from any form of modeling improvement.

Furthermore, the output of LED methods is typically used to directly improve the dataset itself. Subsequently, arbitrary advanced modeling techniques can be applied to the improved dataset, leading to potentially better predictions than obtainable via model-specific MLwNL. In particular, a data analyst or domain expert can manually review/correct the output of LED methods and easily produce greater improvements in dataset quality just by allocating extra reviewing time. Manually improving the predictive accuracy from MLwNL can be far less trivial, often requiring ML expertise and modeling creativity. We also emphasize that these two paradigms are entirely complementary: LED can be used to partially improve a portion of the dataset with subsequent application of MLwNL to account for the remaining noisy labels (Brodley & Friedl, 1999; Northcutt et al., 2021b).

Another challenge in real-world ML projects beyond improving predictions is estimating their accuracy. Unlike MLwNL, LED plays a fundamental role in properly estimating predictive performance given noisy labels. Without LED, accuracy estimates computed from test sets with erroneous labels can lead to suboptimal models potentially being selected for deployment (Northcutt et al., 2021a). Beyond model selection, many other critical decisions also hinge on proper performance estimation such as deciding: when to introduce ML into a product, how much compute/investment to allocate to ML, and how to best translate ML predictions into actions with consequences.

We finally note that, although many MLwNL algorithms could also be applied for the task of LED (Song et al., 2022), past evaluations of MLwNL with naturally-occurring label errors have all focused on predictive accuracy of the trained models rather than how effectively they can detect label errors (Lee et al., 2018; Jiang et al., 2020; Zhang et al., 2021; Wei et al., 2022). Recall the simple label quality scores we evaluate here can be trivially employed with *any* regularly trained classification model. This is a key advantage over other LED methods with more restrictive requirements, as our experiments reveal model accuracy is one of the most important factors determining the performance of methods for LED. In order to remain competitive, a good LED method must be usable with future state-of-the-art models regardless of their architecture and training strategy.

B. Supplementary Results

Additional raw results can be found in the Github repository[3] which reproduces our experiments. Note that the performance of the *Entropy* and *Least-Confidence* scores is listed in the tables, but omitted from figures to avoid clutter (these scores performed poorly overall). The Lift @ 100 metric favors high-precision scores, whereas Lift @ #Errors favors scores capable of detecting a sizeable chunk of the overall set of label errors present in the data.

[3] https://github.com/cleanlab/label-error-detection-benchmarks/

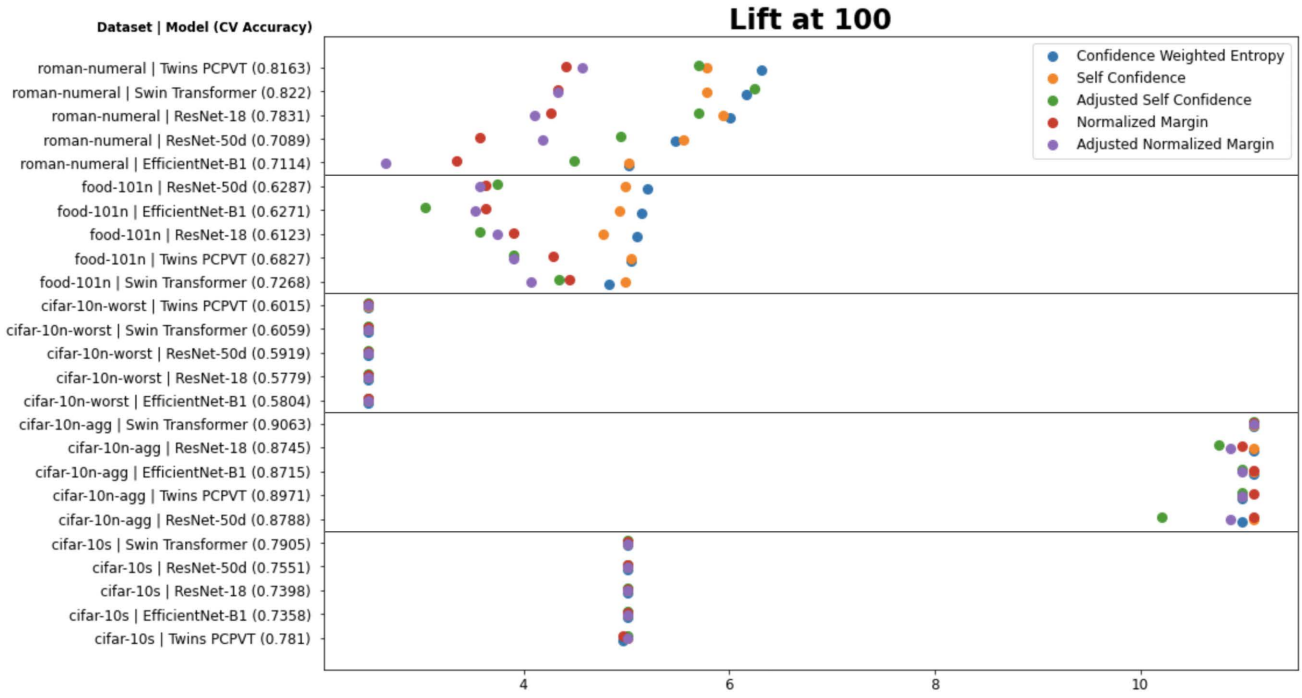

Figure S1. Lift @ 100 achieved by label quality scores for each dataset and model. Models are ordered by accuracy on each dataset.

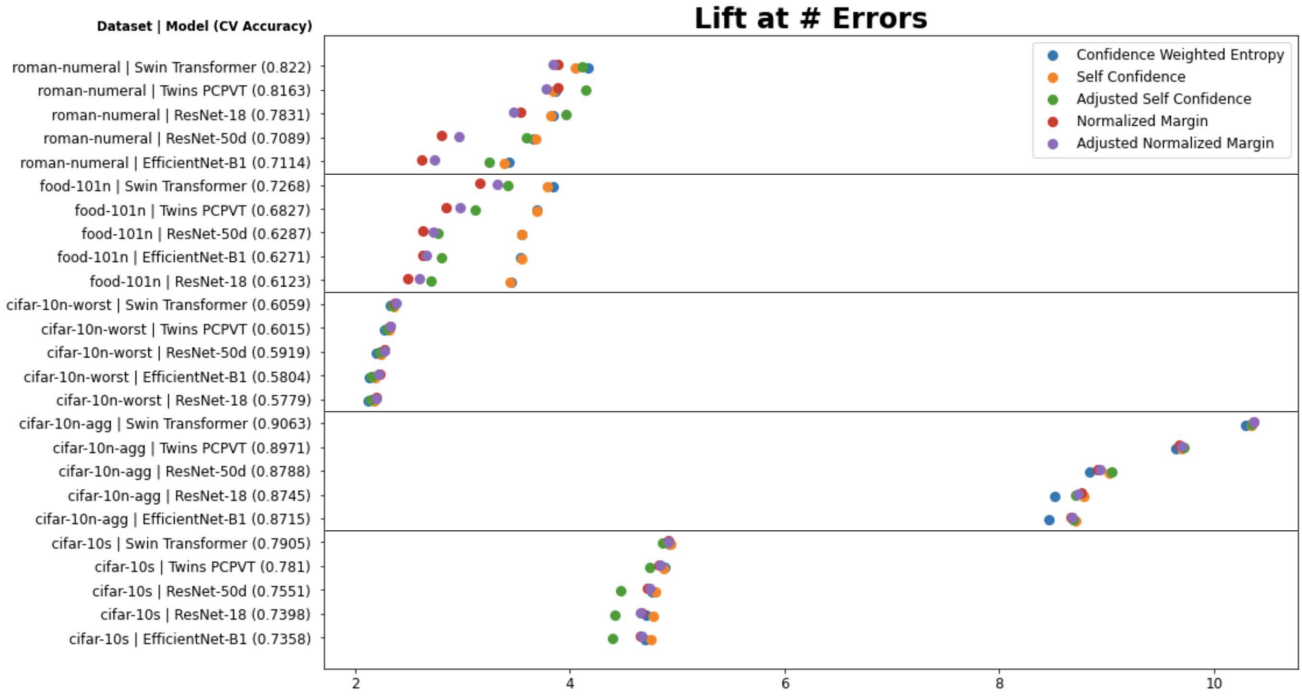

Figure S2. Lift @ T achieved by label quality scores for each dataset and model; T is the true number of label errors in each dataset.

Table S1. AUROC achieved by label quality scores with Swin Transformer model.

Score	roman-numeral	food-101n	cifar-10n-worst	cifar-10n-agg	cifar-10s
Self Confidence	0.8350	0.9223	0.9905	0.9958	0.9996
Normalized Margin	0.8247	0.8906	0.9921	0.9960	0.9991
Adjusted Self Confidence	0.8373	0.9001	0.9892	0.9950	0.9977
Adjusted Normalized Margin	0.8218	0.8911	0.9929	0.9955	0.9993
Confidence Weighted Entropy	0.8366	0.9234	0.9868	0.9951	0.9993
Entropy	0.7716	0.8159	0.6380	0.7572	0.3541
Least Confidence	0.7620	0.8102	0.6549	0.7613	0.3637

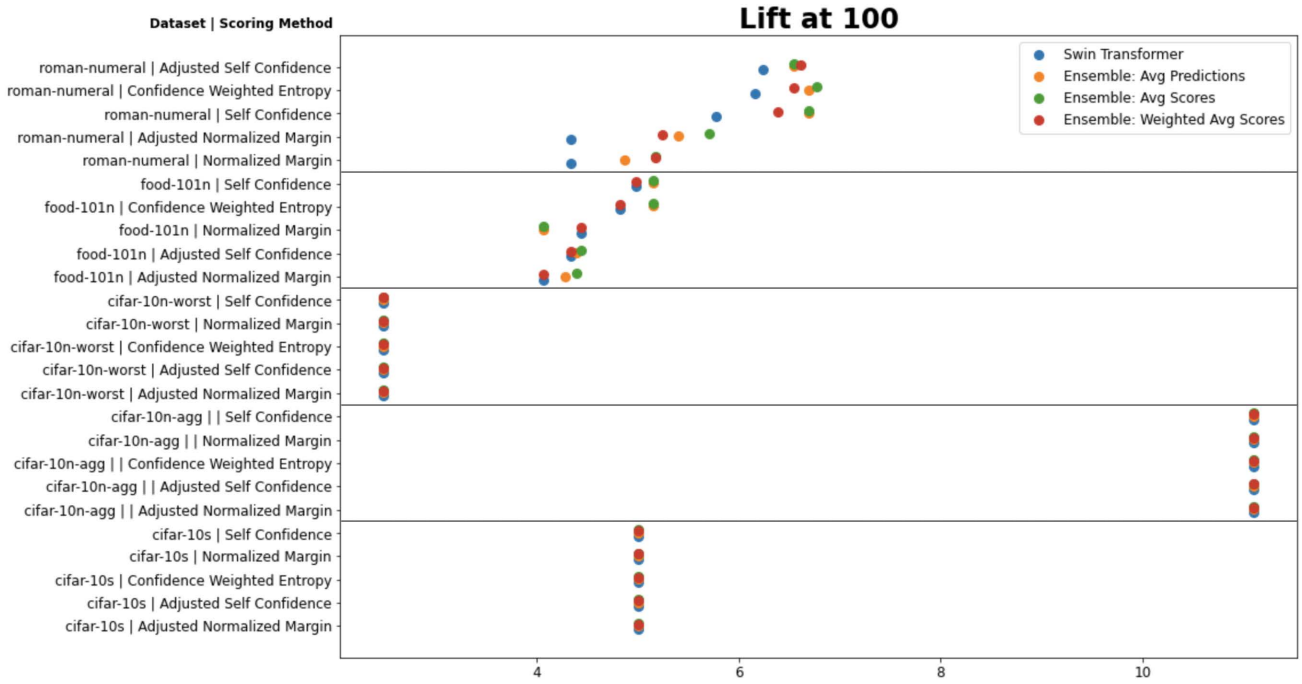

Figure S3. Lift @ 100 achieved by various label quality scores for each dataset. Here we consider various ensembling strategies as well as scores computed with respect to the best individual model, the Swin Transformer.

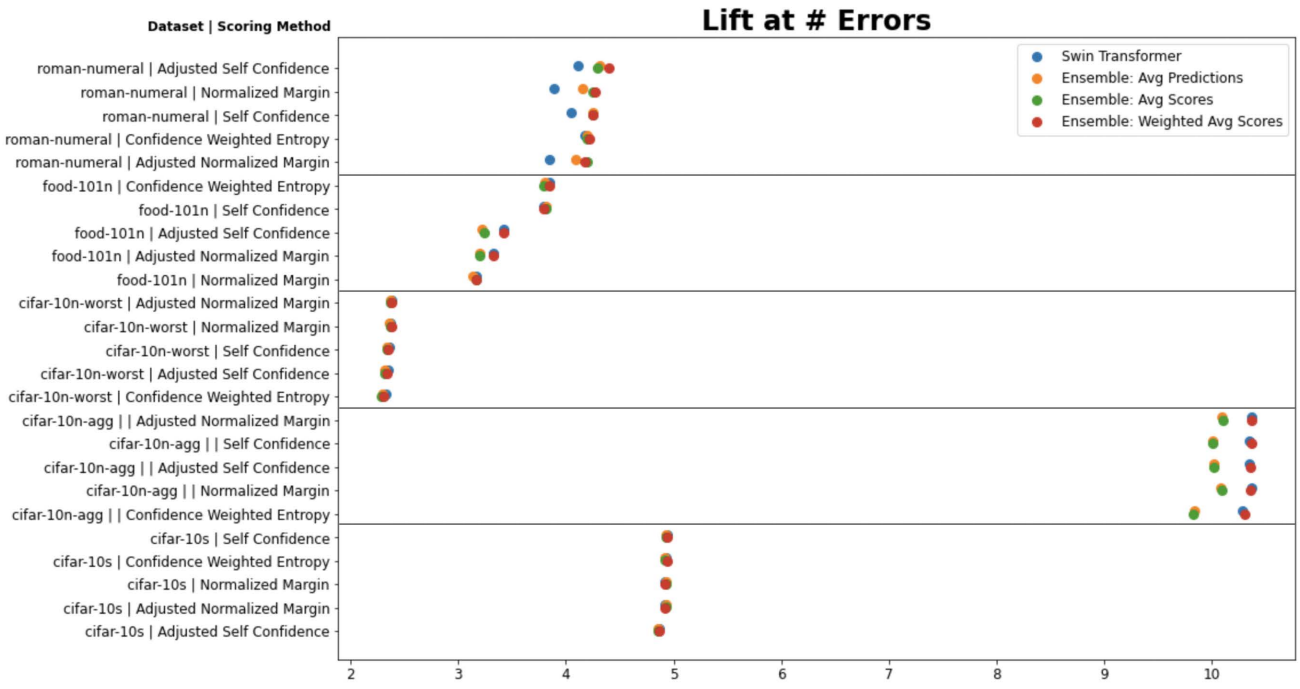

Figure S4. Lift @ T achieved by label quality scores for each dataset, where T is the true number of label errors in each dataset. Here we consider various ensembling strategies as well as scores computed with respect to the best individual model, the Swin Transformer.

C. Comparing Label Quality Scores with Label Error Filters

While much past work has considered detecting label errors via label quality scores (Lee et al., 2018; Müller & Markert, 2019; Huang et al., 2019; Pruthi et al., 2020; Warmerdam, 2021), a key issue is selecting the score threshold below which to flag a label as being incorrect. In practice, such a threshold can be manually determined via human review of a few labels from different score-ranges or based on the number of examples there is time/willingness to review (Müller & Markert, 2019). There exists an alternative class of *filter* methods for LED that circumvent this issue by not relying on numeric label quality scores, but rather using discrete binary statistics to determine a label is either wrong or not (Brodley & Friedl, 1999; Patrini et al., 2017; Chen et al., 2019; Cheng et al., 2020; Northcutt et al., 2021b). Here we compare our label quality scores against a particularly effective model-agnostic set of such filter methods that Northcutt et al. (2021b) proposed as a suite of *confident learning* algorithms to identify label errors.

In particular, we consider the following 4 confident learning filter options:

confident_joint_off_diag, **prune_by_noise_rate**, **prune_by_class**, **both** which are respectively detailed as *CL method* 2, 3, 4, 5 in the paper of Northcutt et al. (2021b). Here we use the same names and implementations provided in the `cleanlab` package[4] (except for the filter option *confident_joint_off_diag*, which is called *confident_learning* in `cleanlab`, even though all of the other above filters also belong to the *confident learning* family of algorithms from Northcutt et al. (2021b)).

We also evaluate the following simple baseline previously considered by Brodley & Friedl (1999); Northcutt et al. (2021b); Lee et al. (2018) and others working on LED:

predicted_neq_given flags an example as having the wrong label if the model's class prediction does not agree with the given label, i.e. if $y \neq \underset{k=1,\dots K}{\operatorname{argmax}}\ p_k$ from $\mathbf{p} = h(x)$.

Table S2 compares these filter options on our datasets, and Figures S5, S6, S7, S8, S9 compare the precision/recall of these hard filters vs. some of our continuous scoring functions (across different score thresholds). These results indicate that for a properly-chosen threshold, our scores can match the LED precision/recall of the filter methods in the 3 versions of Cifar-10 (where there are few OOD examples), and can outperform the filters in the *Food-101n* and *Roman-numeral* datasets (where there are more OOD examples). Note that the theory under which these filter methods were developed only accounts for certain forms of noise in which some labels are replaced with other classes (Northcutt et al., 2021b), as opposed to the existence of OOD examples for which no label in the given set of classes is appropriate. All methods fare worse on the *Food-101n* and *Roman-numeral* datasets where there are more different ways a label can be wrong than in the Cifar-10 variants. Theoretical analysis of label error detection under a broader range of possible error types and more realistic assumptions remains an important line of research (Zhang et al., 2021).

[4] https://github.com/cleanlab/cleanlab

Table S2. Treating LED as a binary classification task (label is correct vs. not) and the outputs of each label filter method as class predictions, we can evaluate these predictions according to their accuracy, precision, or recall. On each dataset, we employ these filters with predictions from our most accurate model, the Swin Transformer.

Dataset	Filter method	Precision	Recall	Accuracy
roman-numeral	prune_by_noise_rate	0.4684	0.1989	0.8651
roman-numeral	prune_by_class	0.6582	0.2796	0.8863
roman-numeral	both	0.5526	0.1129	0.8714
roman-numeral	confident_joint_off_diag	0.4257	0.1156	0.8633
roman-numeral	predicted_neq_given	0.4544	0.6156	0.8523
food-101n	prune_by_noise_rate	0.5508	0.3401	0.8272
food-101n	prune_by_class	0.7542	0.4847	0.8759
food-101n	both	0.7096	0.2425	0.8421
food-101n	confident_joint_off_diag	0.5264	0.2157	0.8196
food-101n	predicted_neq_given	0.5512	0.8168	0.8436
cifar-10n-worst	prune_by_noise_rate	0.9813	0.8724	0.9420
cifar-10n-worst	prune_by_class	0.9833	0.8974	0.9526
cifar-10n-worst	both	0.9880	0.8492	0.9352
cifar-10n-worst	confident_joint_off_diag	0.9766	0.8204	0.9199
cifar-10n-worst	predicted_neq_given	0.9624	0.9432	0.9624
cifar-10n-agg	prune_by_noise_rate	0.9808	0.6238	0.9650
cifar-10n-agg	prune_by_class	0.9955	0.6333	0.9667
cifar-10n-agg	both	0.9957	0.5148	0.9561
cifar-10n-agg	confident_joint_off_diag	0.9898	0.5410	0.9581
cifar-10n-agg	predicted_neq_given	0.9110	0.9476	0.9869
cifar-10s	prune_by_noise_rate	0.9784	0.9957	0.9948
cifar-10s	prune_by_class	0.9619	0.9978	0.9917
cifar-10s	both	0.9798	0.9947	0.9948
cifar-10s	confident_joint_off_diag	0.9854	0.9762	0.9924
cifar-10s	predicted_neq_given	0.9514	0.9994	0.9897

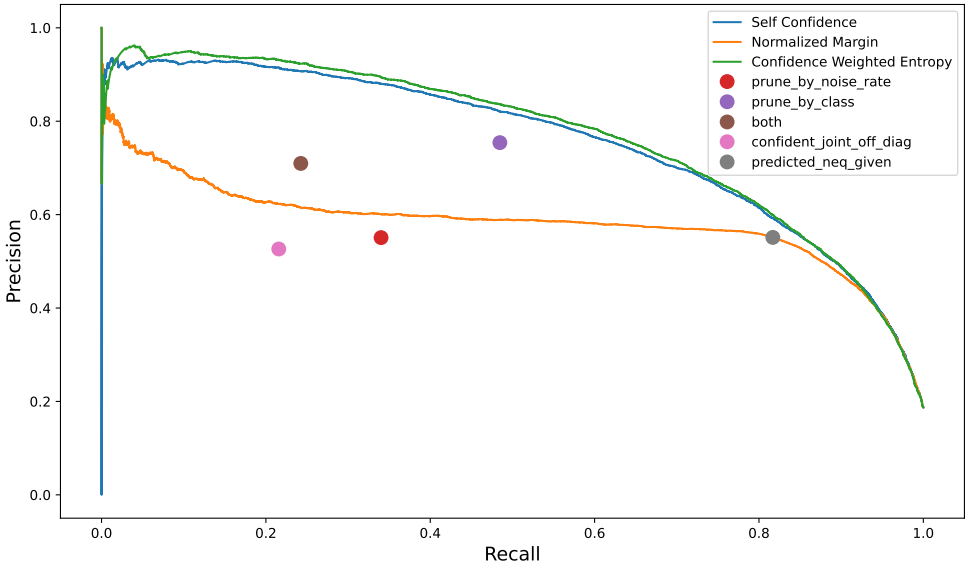

Figure S5. Precision-Recall of label error detection on Food-101n dataset using the Swin Transformer model.

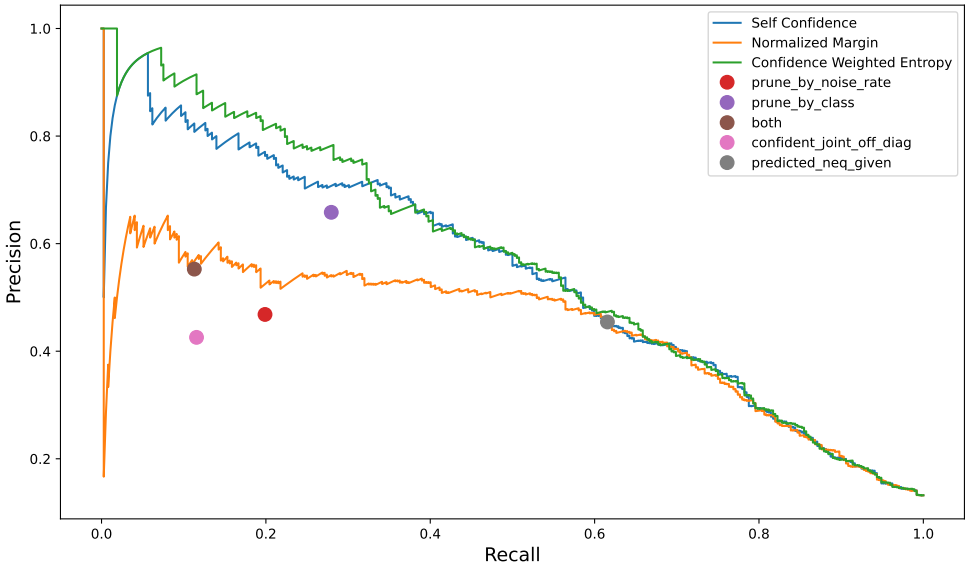

Figure S6. Precision-Recall of label error detection on Roman-numeral dataset using the Swin Transformer model.

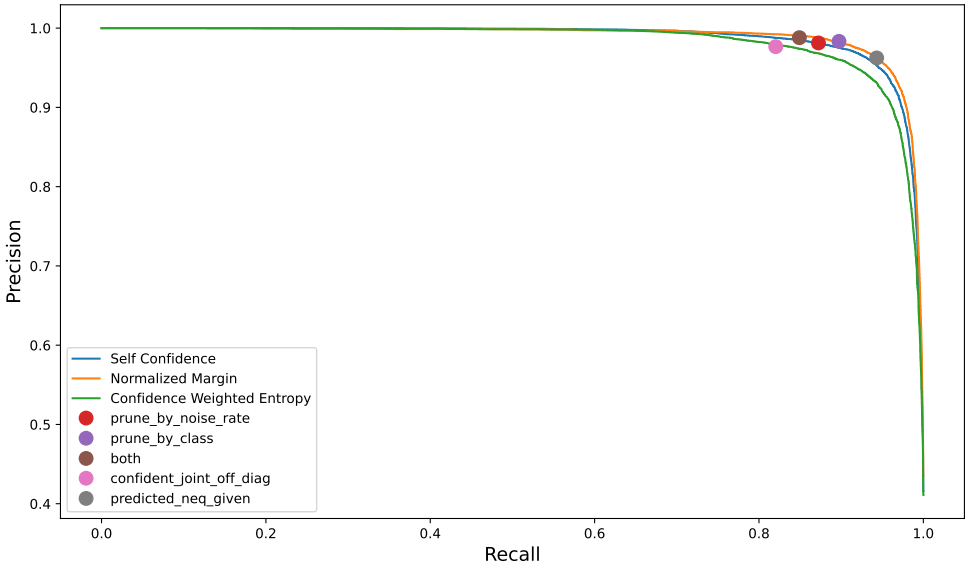

Figure S7. Precision-Recall of label error detection on Cifar-10n-worst dataset using the Swin Transformer model.

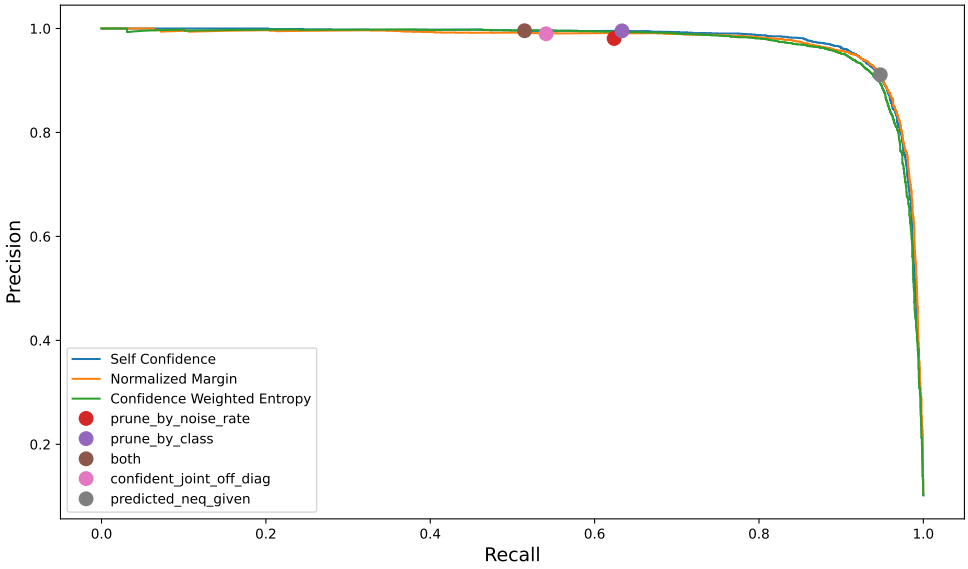

Figure S8. Precision-Recall of label error detection on Cifar-10n-agg dataset using the Swin Transformer model.

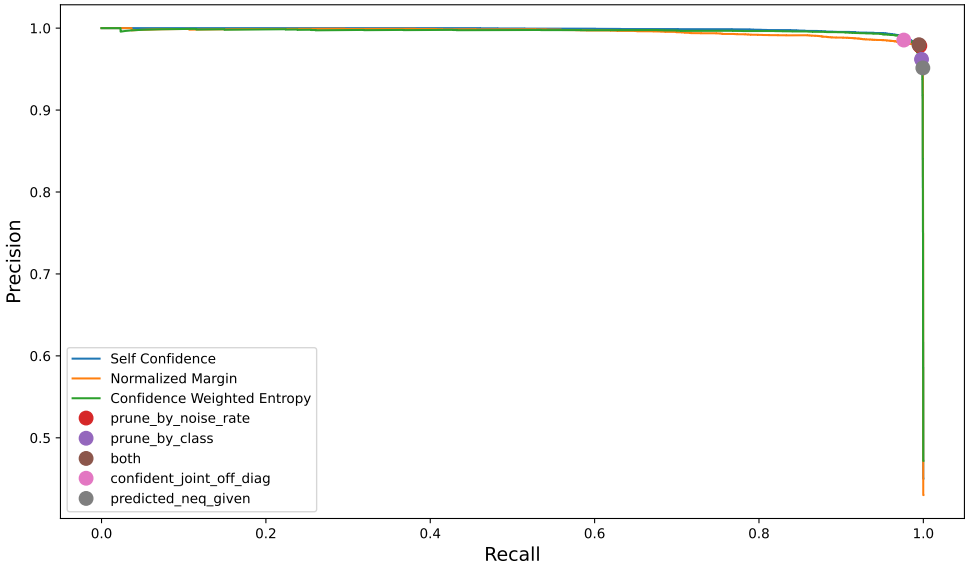

Figure S9. Precision-Recall of label error detection on Cifar-10s dataset using the Swin Transformer model.

D. Experiment Details

To compute many of the label quality scores, we utilize implementations from the `cleanlab` package[5]. Note that in cleanlab, all scores are transformed and rescaled to lie between 0 and 1, in a manner that preserves their relative ordering and thus does not change their LED performance. To avoid having to manually tune/manage model training, all classifiers are fit using the `autogluon` AutoML package (Erickson et al., 2020). Each model is initialized with default pretrained weights and then fine-tuned on our dataset (we use 5-fold cross-validation). Additional details can be found in the code[6] for reproducing our experiments, as can raw results of all methods on all datasets.

Table S3. Accuracy achieved by trained classifiers on each dataset (estimated via 5-fold cross-validation). We also report the accuracy of ensembles that aggregate the predictions of our 5 individual models via a uniform (Lakshminarayanan et al., 2017) or weighted average.

Model	roman-numeral	food-101n	cifar-10n-worst	cifar-10n-agg	cifar-10s
Swin Transformer (Liu et al., 2021)	0.8220	0.7268	0.6059	0.9063	0.7905
Twins PCPVT (Chu et al., 2021)	0.8163	0.6827	0.6015	0.8971	0.7810
EfficientNet-B1 (Tan & Le, 2019)	0.7114	0.6271	0.5804	0.8715	0.7358
ResNet-50d (He et al., 2019)	0.7089	0.6287	0.5919	0.8788	0.7551
ResNet-18 (He et al., 2016)	0.7831	0.6123	0.5779	0.8745	0.7398
Ensemble (Average Predictions)	0.8418	0.7108	0.6064	0.9040	0.7853
Ensemble (Weighted Average; Weights $\propto$ Log Loss)	0.8488	0.7268	0.6078	0.9065	0.7907

E. Details of Weighted Ensemble

Selecting hyperparameter T. Recall that our ensemble weights depend on hyperparameter T. In practice, we can choose the value of T for which the resulting weights lead to the best ensemble-prediction $\mathbf{p}_{\text{ensemble}}$ when used to aggregate predictions rather than scores as in (5). Here we apply grid-search over T optimizing with respect to the overall *log-loss* (Gneiting & Raftery, 2007) between y and $\mathbf{p}_{\text{ensemble}}$ across the dataset. We favor the log-loss metric as it accounts for the accuracy and calibration of $\mathbf{p}$, which both intuitively affect the resulting label quality scores.

Why aggregate scores instead of predictions? Note that for the self-confidence score, it is equivalent to aggregate predictions before computing a single LQS vs. aggregating LQS values from each model into an ensemble LQS. However the latter has some conceptual advantages for other scores. For example, consider a K-way classification task with K models that all predict a different class with high confidence for some given x. Averaging model predictions in this scenario will lead to near-uniform predicted probabilities, so x may fail to be flagged as wrongly labeled, with only a moderate label quality score if we use say the normalized margin. In contrast, the ensemble score for normalized margin will be quite low, as there will be K-1 models that confidently say this is label error and only 1 that does not. Thus x will likely be flagged as badly labeled, as it should be (if we assume our models are reasonably behaved). More generally, we might consider each model's predicted probability $\mathbf{p}^{(j)}$ is an unbiased (but imperfect) estimate of $\mathbf{p}^*$, the true probability of each class given x. Our goal is to estimate the target quantity $\text{Score}(x, y, \mathbf{p}^*)$, which would presumably be the best version of each particular score. Supposing we use uniform aggregation weights and $\text{Score}(\cdot)$ is a nonlinear function of $\mathbf{p}^*$, then aggregating predictions first can produce a biased estimate of the target (due to linearity of expectation), whereas aggregating scores produces an unbiased estimate.

F. Dataset Details

While previous MLwNL benchmarks have studied our chosen datasets, they have to our knowledge not been considered for evaluating LED methods (except for the synthetic *Cifar-10s* dataset). Unlike supervised learning benchmarks, we do

[5] https://github.com/cleanlab/cleanlab

[6] https://github.com/cleanlab/label-error-detection-benchmarks/

not consider train/test splits in our setting. In LED, one is often interested in identifying all of the bad labels across an entire given dataset, which is a key step to ensure the data are of reasonable quality. To this end, we therefore compute label quality scores for all examples and evaluate their effectiveness for detecting all label errors in the dataset.

For each dataset in our benchmark, external ground truth labels are available (see Table S4 for the source of ground truth), which differ from some of the given labels in the dataset on which our LED methods are run. The ground truth labels in each dataset are reserved solely for evaluation. We do not allow LED methods to access any of the ground truth labels since typical ML applications merely have a dataset and some given labels of unknown quality. In our datasets, the given labels may be incorrect for various reasons listed in Table S4. These label errors arise naturally in the datasets named with suffix `__n`, whereas their source is known only by the Data-centric AI competition organizers (Ng, 2021) for the *roman-numeral* data. Glancing through the data, we see far more out-of-distribution examples in *roman-numeral* than in Cifar-10. Finally as a reference to facilitate comparison with past work (Northcutt et al., 2021b; Müller & Markert, 2019; Gu et al., 2021), we also include *cifar-10s* with synthetic label errors introduced by Northcutt et al. (2021b) who randomly replaced some labels with other classes. While our findings for these synthetic label errors qualitatively agree with the conclusions of prior work, our results appear substantially different on the other datasets, highlighting the need for benchmarks with real label errors.

Table S4. Descriptions of each dataset, including the source of the noisy given labels and true underlying labels, as well as a link to the data. Note that the labels in the original Cifar-10 data have been previously validated as being quite high-quality (Northcutt et al., 2021a).

Dataset	Description
food-101n (Lee et al., 2018)	Noisy labels arise from dataset being curated via web-crawling. Labels were manually verified for subset of 53k training images, we discard the rest of the dataset. https://kuanghuei.github.io/Food-101N/
cifar-10n-worst (Wei et al., 2022)	Noisy labels obtained from 3 Amazon Mechanical Turk annotators. Given label is incorrect if any of the 3 annotators chose incorrect label (high noise). True labels are the original Cifar-10 labels. http://ucsc-real.soe.ucsc.edu:1995/Home.html
cifar-10n-agg (Wei et al., 2022)	Noisy labels obtained from 3 Amazon Mechanical Turk annotators. Given label is majority-vote aggregate of the annotators' choices (less noise). True labels are the original Cifar-10 labels. http://ucsc-real.soe.ucsc.edu:1995/Home.html
roman-numeral (Ng, 2021)	Dataset from Andrew Ng's 2021 Data-centric AI competition. Noisy labels stem from unknown sources (known only to competition organizers). Contains a myriad of different types of label errors. For this paper, we have manually verified labels for all examples in this dataset. https://github.com/cleanlab/label-error-detection-benchmarks/ (andrew-ng-dcai-comp-2021-manual-review-for-label-errors.xlsx)
cifar-10s (Northcutt et al., 2021b)	Synthetically introduced class-conditional label noise with 20% noisy labels, 40% sparsity in label-swapping rates between classes. True labels are the original Cifar-10 labels. https://github.com/cleanlab/label-error-detection-benchmarks/ (cifar10_train_dataset_noise_amount_0.2_sparsity_0.4_20220326055753.csv)

Figure S10. Example label errors in each dataset (Cifar-10s not shown as it only has synthetically-introduced label errors).

G. Discussion

This work presented a first empirical evaluation of label error detection methods on real-world data with naturally-occuring label errors. We hope our study helps inform which label quality scoring methods will be effective for identifying label errors in practice. More comprehensive benchmarks will be required for a deeper understanding of the empirical strengths/weaknesses of each score, given the extreme variation in datasets and types of label errors encountered in the wild. Important extensions to our LED benchmark left for future work include:

1. Adding real-world datasets beyond the image modality, which will require collecting more ground truth labels. In new benchmarks, it is important to ensure there is no systematic distribution shift (i.e. confounding or selection bias) in the examples for which ground truth labels happen to be available.
2. Adding prediction tasks beyond classification, such as those with structured labels like object detection, where datasets are also plagued by label errors (Kang et al., 2022).
3. Also evaluating model-specific label quality scores and other LED methods that leverage particular properties of certain models (Lee et al., 2018; Cheng et al., 2020; Song et al., 2022).