PERVASIVE LABEL ERRORS IN TEST SETS DESTABILIZE MACHINE LEARNING BENCHMARKS

Curtis G. Northcutt
ChipBrain, MIT
Boston, MA, USA
curtis@chipbrain.com
cgn@mit.edu

Anish Athalye
Dept. of EECS
MIT
Cambridge, MA, USA
aathalye@mit.edu

Jonas Mueller
Amazon Web Services
East Palo Alto, CA, USA
jonaswmueller@gmail.com

ABSTRACT

We identify label errors in the *test* sets of 10 of the most commonly-used computer vision, natural language, and audio datasets, and subsequently study the potential for these label errors to affect benchmark results. Errors in test sets are numerous and widespread: we estimate an average of 3.4% errors across the 10 datasets,[1] where for example 2916 label errors comprise 6% of the ImageNet validation set. Putative label errors are identified using confident learning algorithms and then human-validated via crowdsourcing (54% of the algorithmically-flagged candidates are indeed erroneously labeled). Traditionally, machine learning practitioners choose which model to deploy based on test accuracy — our findings advise caution here, proposing that judging models over correctly labeled test sets may be more useful, especially for noisy real-world datasets. Surprisingly, we find that lower capacity models may be practically more useful than higher capacity models in real-world datasets with high proportions of erroneously labeled data. For example, on ImageNet with corrected labels: ResNet-18 outperforms ResNet-50 if the prevalence of originally mislabeled test examples increases by just 6%. On CIFAR-10 with corrected labels: VGG-11 outperforms VGG-19 if the prevalence of originally mislabeled test examples increases by just 5%.

1 INTRODUCTION

Large labeled data sets have been critical to the success of supervised machine learning across the board in domains such as image classification, sentiment analysis, and audio classification. Yet, the processes used to construct datasets often involve some degree of automatic labeling or crowdsourcing, techniques which are inherently error-prone (Sambasivan et al., 2021). Even with controls for error correction (Kremer et al., 2018; Zhang et al., 2017), errors can slip through. Prior work has considered the consequences of noisy labels, usually in the context of *learning* with noisy labels, and usually focused on noise in the *train* set. Some past research has concluded that label noise is not a major concern, because of techniques to learn with noisy labels (Patrini et al., 2017; Natarajan et al., 2013), and also because deep learning is believed to be naturally robust to label noise (Rolnick et al., 2017; Sun et al., 2017; Huang et al., 2019; Mahajan et al., 2018).

However, label errors in *test* sets are less-studied and have a different set of potential consequences. Whereas *train* set labels in a small number of machine learning datasets, e.g. in the ImageNet dataset, are well-known to contain errors (Northcutt et al., 2021; Shankar et al., 2020; Hooker et al., 2019), labeled data in *test* sets is often considered "correct" as long as it is drawn from the same distribution as the train set — this is a fallacy — machine learning *test* sets can, and do, contain pervasive errors and these errors can destabilize ML benchmarks.

Researchers rely on benchmark test datasets to evaluate and measure progress in the state-of-the-art and to validate theoretical findings. If label errors occurred profusely, they could potentially undermine the framework by which we measure progress in machine learning. Practitioners rely on their own real-world datasets which are often more noisy than carefully-curated benchmark datasets.

[1] To view the mislabeled examples in these benchmarks, go to https://labelerrors.com.

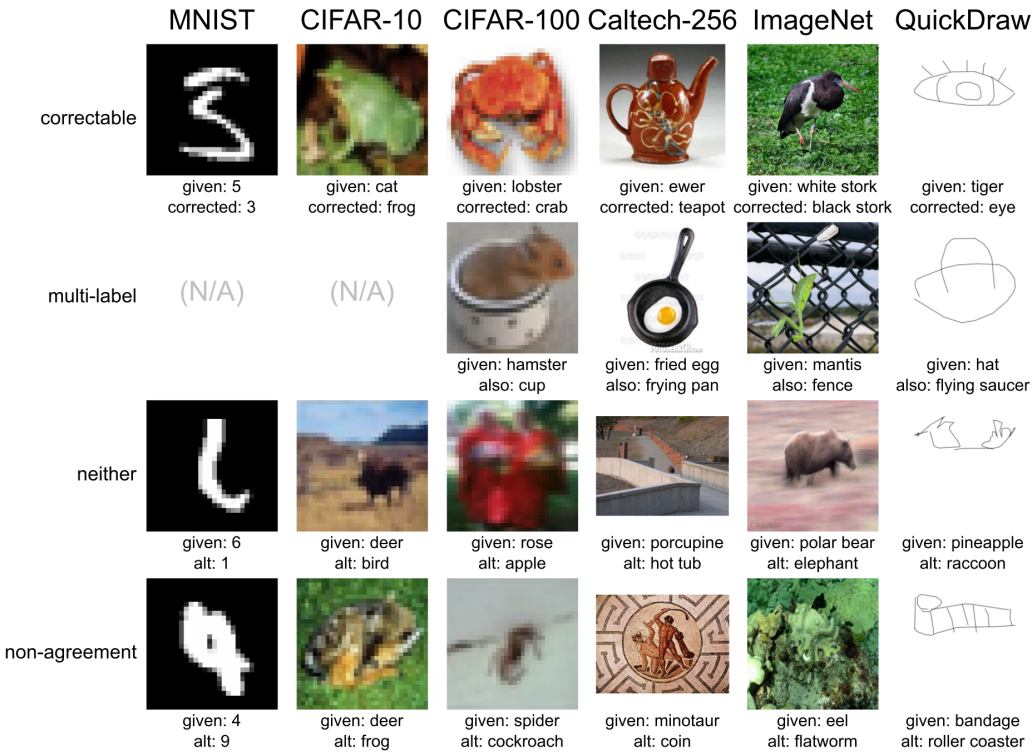


Figure 1: An example label error from each category (Sec. 4) for image datasets. The figure shows given labels, human-validated corrected labels, also the second label for multi-class data points, and CL-guessed alternatives. A browser for all label errors across all 10 datasets is available at https://labelerrors.com. Errors from text and audio datasets are also included on the website.

Label errors in these test sets could potentially lead practitioners to incorrect conclusions about which models actually perform best in the real world.

We present the first study that identifies and systematically analyzes label errors across 10 commonly-used datasets across computer vision, natural language processing, and audio processing. Unlike prior work on noisy labels, we do not experiment with synthetic noise but with naturally-occurring errors. Rather than exploring a novel methodology for dealing with label errors, which has been extensively studied in the literature (Cordeiro & Carneiro, 2020), this paper aims to characterize the prevalence of label errors in the test data of popular benchmarks used to measure ML progress, and we subsequently analyze practical consequences of these errors, and in particular, their effects on model selection. Using *confident learning* (Northcutt et al., 2021), we algorithmically identify putative label errors in test sets at scale [2], and we validate these label errors through human evaluation, estimating an average of 3.4% errors. We identify, for example, 2916 (6%) errors in the ImageNet validation set (which is *commonly used as a test set*), and estimate over 5 million (10%) errors in QuickDraw. Figure 1 shows examples of validated label errors for the image datasets in our study.

We use ImageNet and CIFAR-10 as case studies to understand the consequences of test set label errors on benchmark stability. While there are numerous erroneous labels in these benchmarks' test data, we find that relative rankings of models in benchmarks are unaffected after removing or correcting these label errors. However, we find that these benchmark results are *unstable*: higher-capacity models (like NasNet) undesirably reflect the distribution of systematic label errors in their predictions to a far greater degree than models with fewer parameters (like ResNet-18), and this effect *increases* with the prevalence of mislabeled test data. This is not traditional overfitting. Larger models are

[2]To find all label errors, we use the cleanlab implementation of confident learning open-sourced at: https://github.com/cgnorthcutt/cleanlab

able to generalize better to the given noisy labels in the test data, but this is problematic because these models produce *worse* predictions than their lower-capacity counterparts when evaluated on the corrected labels for mislabeled test examples.

In real-world settings with high proportions of erroneously labeled data, lower capacity models may thus be practically more useful than their higher capacity counterparts. For example, it may appear NasNet is superior to ResNet-18 based on the test accuracy over originally given labels, but NasNet is in fact worse than ResNet-18 based on the test accuracy over corrected labels. Since the latter form of accuracy is what matters in practice, ResNet-18 should actually be deployed instead of NasNet here – but this is unknowable without correcting the test data labels.

To evaluate how benchmarks of popular pre-trained models change, we incrementally increase the noise prevalence by controlling for the proportion of correctable (but originally mislabeled) data within the test dataset. This procedure allows us to measure the noise prevalence in each test set where benchmark rankings change. For example, on ImageNet with corrected labels: ResNet-18 outperforms ResNet-50 if the prevalence of originally mislabeled test examples increases by just 6%.

Our contributions include:

1. Using a simple algorithmic + crowdsourcing pipeline to identify and validate label errors, we discover label errors are pervasive in test sets of popular benchmarks used in nearly all machine learning research.
2. We provide a cleaned and corrected version of each test set [3], in which a large fraction of the label errors have been corrected by humans. We hope future research on these benchmarks will use this improved test data instead of the original erroneous labels.
3. We analyze the implications of pervasive test set label errors. We find that higher capacity models perform better on the subset of incorrectly-labeled test data in terms of their accuracy on the original labels (i.e., what one traditionally measures), but these models perform worse on this subset than their simpler counterparts in terms of their accuracy on corrected labels (i.e., what one cares about in practice, but cannot measure without the manually-corrected test data we provide).
4. In case studies with commonly-used benchmark datasets, we identify the prevalence of originally mislabeled test data needed to destabilize ML benchmarks, i.e., for low-capacity models to outperform high-capacity models. We discover that merely slight increases in the test label error prevalence would cause model selection to favor the wrong model based on standard test accuracy.

Our findings imply ML practitioners might benefit from correcting test set labels to benchmark how their models will perform in real-world deployment, and by using simpler/smaller models in applications where labels for their datasets tend to be noisier than the labels in gold-standard benchmark datasets. One way to ascertain whether a dataset is noisy enough to suffer from this effect is to correct at least the test set labels, e.g. using our straightforward approach.

2 Background and Related work

Experiments in learning with noisy labels (Patrini et al., 2016; Van Rooyen et al., 2015; Natarajan et al., 2013; Jindal et al., 2016; Sukhbaatar et al., 2015) suffer a double-edged sword: either synthetic noise must be added to clean training data to measure performance on a clean test set, at the expense of modeling *actual* real-world label noise (Jiang et al., 2020), or, a naturally noisy dataset is used and accuracy is measured on a noisy test set. In the noisy WebVision dataset (Li et al., 2017), accuracy on the ImageNet validation is often reported as a "clean" test set, however, related works (Recht et al., 2019; Northcutt et al., 2021; Tsipras et al., 2020; Hooker et al., 2019) have already shown the existence of label issues in ImageNet. Unlike these works, we focus exclusively on existence and implications of label errors in the test set, and extend our analysis to many types of datasets. Although extensive pior work deals with label errors in the *training* set (Frénay & Verleysen, 2014;

[3] A corrected version of each test set is provided at https://github.com/cgnorthcutt/label-errors.

Cordeiro & Carneiro, 2020), much less work has been done to understand the implications of label errors in the *test set*.

Crowd-sourced curation of labels via multiple human workers (Zhang et al., 2017; Dawid & Skene, 1979; Ratner et al., 2016) is a common method for validating/correcting label issues in datasets, but it can be exorbitantly expensive for large datasets. To circumvent this issue, we only validate subsets of datasets by first estimating which examples are most likely to be mislabeled. To achieve this, we lean on a number of contributions in uncertainty quantification for finding label errors based on prediction/label agreement via confusion matrices (Xu et al., 2019; Hendrycks et al., 2018; Chen et al., 2019; Lipton et al., 2018), however these approaches lack either robustness to class imbalance or theoretical support for realistic settings with *asymmetric, non-uniform noise*. For robustness to class imbalance and theoretical support for exact uncertainty quantification, we use the model-agnostic framework, confident learning (CL) (Northcutt et al., 2021), to estimate which labels are erroneous across diverse datasets. Northcutt et al. (2021) have demonstrated that CL more accurately identifies label errors than other label-error identification methods. Unlike prior work that only models symmetric label noise (Van Rooyen et al., 2015), we quantify class-conditional label noise with CL, validating the correctable nature of the label errors via crowdsourced workers. Human validation confirms the noise in common benchmark datasets is indeed primarily systematic mislabeling, not just random noise or lack of signal (e.g. images with fingers blocking the camera).

Datasets An overview of each dataset, how it was created, and any alterations we made for the experiments in this paper, is discussed in Appendix A.

3 IDENTIFYING LABEL ERRORS IN BENCHMARK DATASETS

Here we summarize our algorithmic label error identification performed prior to crowd-sourced human verification. The primary contribution of this section is not in the methodology, which is covered extensively in (Northcutt et al., 2021), but in its utilization as a *filtering* process to significantly (often as much as 90%) reduce the number of examples requiring human validation in the next step.

To identify label errors in a test dataset with n examples and m classes, we first characterize label noise in the dataset using the confident learning (CL) framework (Northcutt et al., 2021) to estimate $\boldsymbol{Q}_{\tilde{y},y^*}$, the $m \times m$ discrete joint distribution of observed, noisy labels, $\tilde{y}$, and unknown, true labels, y^*. Inherent in $\boldsymbol{Q}_{\tilde{y},y^*}$ is the assumption that noise is class-conditional (Angluin & Laird, 1988), depending only on the latent true class, not the data. This assumption is commonly used (Goldberger & Ben-Reuven, 2017; Sukhbaatar et al., 2015) because it is reasonable. For example, in ImageNet, a *tiger* is more likely to be mislabeled *cheetah* than *flute*.

The diagonal entry, $\hat{p}(\tilde{y}{=}i, y^*{=}i)$, of matrix $\boldsymbol{Q}_{\tilde{y},y^*}$ is the probability that examples in class i are correctly labeled. Thus, if the dataset is error-free, then $\sum_{i\in[m]} \hat{p}(\tilde{y}{=}i, y^*{=}i) = 1$. The fraction of label errors is $\rho = 1 - \sum_{i\in[m]} \hat{p}(\tilde{y}{=}i, y^*{=}i)$ and the number of label errors is $\rho \cdot n$. To find label errors, we choose the top $\rho \cdot n$ examples ordered by the normalized margin: $\hat{p}(\tilde{y}{=}i; \boldsymbol{x}, \boldsymbol{\theta}) - \max_{j\neq i} \hat{p}(\tilde{y}{=}j; \boldsymbol{x}, \boldsymbol{\theta})$ (Wei et al., 2018). Table 1 shows the number of CL guessed label issues for each test set across ten popular ML benchmark datasets. Confident learning estimation of $\boldsymbol{Q}_{\tilde{y},y^*}$ is summarized in the appendices in (Sec. B).

Computing out-of-sample predicted probabilities Estimating $\boldsymbol{Q}_{\tilde{y},y^*}$ for CL noise characterization requires two inputs for each dataset: (1) out-of-sample predicted probabilities $\hat{\boldsymbol{P}}_{k,i}$ ($n \times m$ matrix) and (2) the test set labels $\tilde{y}_k$. We observe the best results computing $\hat{\boldsymbol{P}}_{k,i}$ by pre-training on the train set, then fine-tuning (all layers) on the test set using cross-validation to ensure $\hat{\boldsymbol{P}}_{k,i}$ is out-of-sample. If pre-trained models are open-sourced (e.g. ImageNet), we use them instead of pre-training ourselves. If the dataset did not have an explicit test set (e.g. QuickDraw and Amazon Reviews), we skip pre-training, and compute $\hat{\boldsymbol{P}}_{k,i}$ using cross-validation on the entire dataset. For all datasets, we try common models that achieve reasonable accuracy with minimal hyper-parameter tuning, and use the model yielding the highest cross-validation accuracy, reported in Table 1.

Using this approach allows us to find label errors without manually checking the entire test set, because CL identifies potential label errors automatically.

Table 1: Test set errors are prominent across common benchmark datasets. Errors are estimated using confident learning (CL) and validated by human workers on Mechanical Turk.

Dataset	Modality	Size	Model	Test Set Errors				
				CL guessed	MTurk checked	validated	estimated	% error
MNIST	image	10,000	2-conv CNN	100	100 (100%)	15	-	0.15
CIFAR-10	image	10,000	VGG	275	275 (100%)	54	-	0.54
CIFAR-100	image	10,000	VGG	2235	2235 (100%)	585	-	5.85
Caltech-256	image	30,607	ResNet-152	4,643	400 (8.6%)	65	754	2.46
ImageNet*	image	50,000	ResNet-50	5,440	5,440 (100%)	2,916	-	5.83
QuickDraw	image	50,426,266	VGG	6,825,383	2,500 (0.04%)	1870	5,105,386	10.12
20news	text	7,532	TFIDF + SGD	93	93 (100%)	82	-	1.11
IMDB	text	25,000	FastText	1,310	1,310 (100%)	725	-	2.9
Amazon	text	9,996,437	FastText	533,249	1,000 (0.2%)	732	390,338	3.9
AudioSet	audio	20,371	VGG	307	307 (100%)	275	-	1.35

*Because the ImageNet test set labels are not publicly available, the ILSVRC 2012 validation set is used.

Table 2: Mechanical Turk validation confirming the existence of pervasive label errors and categorizing the types of label issues.

Dataset	Test Set Errors Categorization					
	non-errors	errors	non-agreement	correctable	multi-label	neither
MNIST	85	15	2	10	-	3
CIFAR-10	221	54	32	18	0	4
CIFAR-100	1650	585	210	318	20	37
Caltech-256	335	65	25	22	5	13
ImageNet	2524	2916	598	1428	597	293
QuickDraw	630	1870	563	1047	20	240
20news	11	82	43	22	12	5
IMDB	585	725	552	173	-	-
Amazon	268	732	430	302	-	-
AudioSet	32	275	-	-	-	-

4 VALIDATING LABEL ERRORS

We validated the algorithmically identified label errors with a Mechanical Turk study. For three datasets with a large number of errors (Caltech-256, QuickDraw, and Amazon Reviews), we checked a random sample; for the rest, we checked all identified errors.

We presented workers with hypothesized errors and asked them whether they saw the (1) given label, (2) the top CL-predicted label, (3) both labels, or (4) neither label in the example. To aid the worker, the interface included high-confidence examples drawn from the training set of the given class and the CL-predicted class. Figure 2 shows the Mechanical Turk worker interface, showing a data point from the CIFAR-10 dataset.

Each CL-identified label error was independently presented to five workers. We consider the example validated (an "error") if fewer than three of the workers agree that the data point has the given label (*agreement threshold = 3 of 5*) , otherwise we consider it to be a "non-error" (i.e. the original label was correct). We further categorize the label errors, breaking them down into (1) "correctable", where a majority agree on the CL-predicted label; (2) "multi-label", where a majority agree on both labels appearing; (3) "neither", where a majority agree on neither label appearing; and (4) "non-agreement", a catch-all category for when there is no majority. Table 2 summarizes the results, and Figure 1 shows examples of validated label errors from image datasets.

5 IMPLICATIONS OF LABEL ERRORS IN TEST DATA

Finally, we consider how these pervasive test set label errors may affect ML practice in real-world applications. To clarify the discussion, we first introduce some useful terminology.

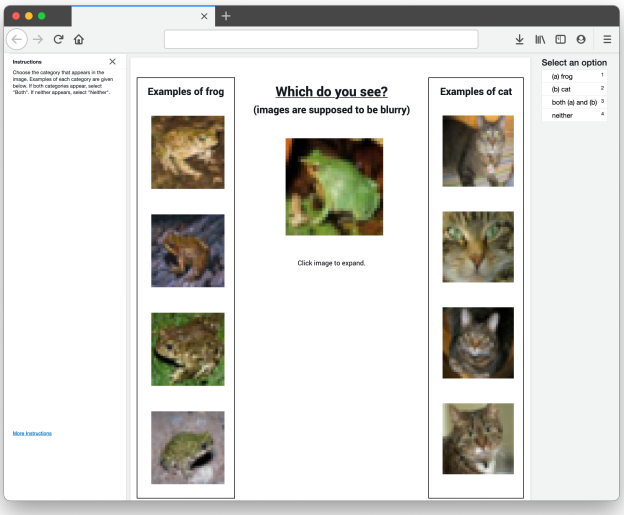


Figure 2: Mechanical Turk worker interface showing an example from CIFAR-10 (with given label "cat"). For each data point algorithmically identified as a potential label error, the interface presents the data point, along with examples belonging to the given class. The interface also shows data points belonging to the confidently predicted class. Either the given is shown as option (a) and predicted is shown as option (b), or vice versa (chosen randomly). The worker is asked whether the image belongs to class (a), (b), both, or neither.

Definition 1 (original accuracy, $\tilde{A}$)**.** *The accuracy of a model's predicted labels over a given dataset, computed with respect to the original labels present in the dataset. Measuring this over the test set is the standard way practitioners evaluate their models today.*

Definition 2 (corrected accuracy, A^*)**.** *The accuracy of a model's predicted labels, computed with respect to a new version of the given dataset in which previously identified erroneous labels have been corrected through human revision to the true class when possible and removed when not. Measuring this over the test set is preferable to $\tilde{A}$ for evaluating models (because A^* better reflects performance in real-world applications).*

In the following definitions, "$\setminus$" denotes a set difference, $\mathcal{D}$ denotes the full test dataset, and $\mathcal{B} \subset \mathcal{D}$ denotes the subset of benign test examples that CL did *not* flag as likely label errors.

Definition 3 (unknown-label set, $\mathcal{U}$)**.** *The subset of CL-flagged test examples for which human labelers could not agree on a correct label ($\mathcal{U} \subset \mathcal{D} \setminus \mathcal{B}$). This includes examples where human reviewers agreed that multiple classes or none of the classes are appropriate.*

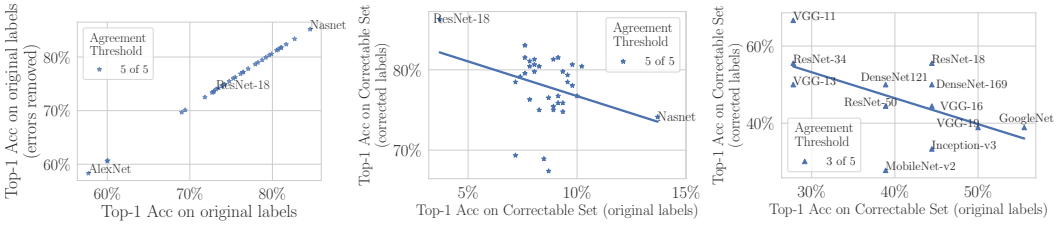
Definition 4 (pruned set, $\mathcal{P}$)**.** *The remaining test data after removing $\mathcal{U}$ from $\mathcal{D}$ ($\mathcal{P} = \mathcal{D} \setminus \mathcal{U}$).*

Definition 5 (correctable set, $\mathcal{C}$)**.** *The subset of CL-flagged examples for which human-validation reached consensus on a different label than the originally given label ($\mathcal{C} = \mathcal{P} \setminus \mathcal{B}$).*

Definition 6 (noise prevalence, N)**.** *The percentage of the* pruned set *comprised of the* correctable set*, i.e. what fraction of data received the wrong label in the original benchmark when a clear alternative ground-truth label should have been assigned (disregarding any data for which humans failed to find a clear alternative). Here we operationalize noise prevalence as $N = \frac{|\mathcal{C}|}{|\mathcal{P}|}$.*

These definitions imply $\mathcal{B}, \mathcal{C}, \mathcal{U}$ are disjoint with $\mathcal{D} = \mathcal{B} \cup \mathcal{C} \cup \mathcal{U}$, and also $\mathcal{P} = \mathcal{B} \cup \mathcal{C}$. In subsequent experiments, we report corrected test accuracy over $\mathcal{P}$ after correcting all of the labels in $\mathcal{C} \subset \mathcal{P}$. We ignore the unknown-label set $\mathcal{U}$ (and no longer include those examples in our estimate of noise prevalence) because it is unclear how to measure *corrected accuracy* for examples whose true underlying label remains ambiguous. Thus the *noise prevalence* reported throughout this section differs from the fraction of label errors originally found in each of the test sets.

A major issue in ML today is that one only sees the original test accuracy in practice, whereas one would prefer to base modeling decisions on the corrected test accuracy instead. Our subsequent discussion highlights the potential implications of this mismatch. What are the consequences of test



(a) ImageNet Val set Acc. (b) ImageNet Correctable set Acc. (c) CIFAR-10 Correctable set Acc.

Figure 3: Benchmark ranking comparison of 34 models pre-trained on ImageNet and 13 pre-trained on CIFAR-10 (more details in Tables S2 and S1 and Fig. S1, in the Appendix). Benchmarks are unchanged by removing label errors (a), but change drastically on the Correctable set with original (erroneous) labels versus corrected labels, e.g. Nasnet: 1/34 $\rightarrow$ 29/34, ResNet-18: 34/34 $\rightarrow$ 1/34.

set label errors? Figure 3 compares performance on the ImageNet validation set, *commonly used in place of the test set*, of 34 pre-trained models from the PyTorch and Keras repositories. Figure 3a confirms the observations of Recht et al. (2019); benchmark conclusions are largely unchanged by using a corrected test set, i.e. in our case by removing errors.

However, we find a surprising result upon closer examination of the models' performance *on the erroneously labeled data*, which we call the "correctable set" $\mathcal{C}$. When evaluating models *only* on the subset of test examples in $\mathcal{C}$, models which perform best on the original (incorrect) labels perform the worst on corrected labels. For example, ResNet-18 (He et al., 2016) significantly outperforms NasNet (Zoph et al., 2018) in terms of corrected accuracy over $\mathcal{C}$, despite exhibiting far worse original test accuracy. The change in ranking can be dramatic: Nasnet-large drops from ranking 1/34 $\rightarrow$ 29/34, Xception drops from ranking 2/34 $\rightarrow$ 24/34, ResNet-18 increases from ranking 34/34 $\rightarrow$ 1/34, and ResNet-50 increases from ranking 20/24 $\rightarrow$ 2/24 (see Table S1 in the Appendices). We verified that the same trend occurs independently across 13 models pre-trained on CIFAR-10 (Figure 3c), e.g. VGG-11 significantly outperforms VGG-19 (Simonyan & Zisserman, 2014) in terms of corrected accuracy over $\mathcal{C}$. Note that all numbers reported here are over subsets of the held-out test data, so this is not overfitting in the classical sense.

This phenomenon, depicted in Figures 3b and 3c, may indicate two key insights: (1) lower-capacity models provide unexpected regularization benefits and are more resistant to learning the asymmetric distribution of noisy labels, (2) over time, the more recent (larger) models have architecture/hyperparameter decisions that were made on the basis of the (original) test accuracy. Learning to capture systematic patterns of label error in their predictions allows these models to improve their original test accuracy, but this is not the sort of progress ML research should aim to achieve. Harutyunyan et al. (2020); Arpit et al. (2017) have previously analyzed phenomena similar to (1), and here we demonstrate that this issue really does occur for the models/datasets widely used in current practice. (2) is an undesirable form of overfitting, albeit not in the classical sense (as the original test accuracy can further improve through better modeling of label errors), but rather overfitting to the specific benchmark (and quirks of the original label annotators) such that accuracy improvements for erroneous labels may not translate to superior performance in a deployed ML system.

This phenomenon has important practical implications for real-world datasets with greater noise prevalence than the highly curated benchmark data studied here. In these relatively clean benchmark datasets, the noise prevalence is an underestimate as we could only verify a subset of our candidate label errors rather than all test labels, and thus the potential gap between original vs. corrected test accuracy is limited for these particular benchmarks. However, this gap increases proportionally for data with more (correctable) label errors in the test set.

To evaluate how benchmarks of popular pre-trained models change, we randomly and incrementally remove correctly-labeled examples, one at a time, until only the original set of mislabeled test data (with corrected labels) is left. We create alternate versions (subsets) of the pruned benchmark test data $\mathcal{P}$, in which we additionally randomly omit some fraction, x, of $\mathcal{B}$ (the test examples that were not identified to have label errors). This effectively increases the proportion of the resulting test dataset comprised of the correctable set $\mathcal{C}$, and reflects how test sets function in applications with greater prevalence of label errors. If we remove a fraction x of benign test examples (in $\mathcal{B}$) from $\mathcal{P}$, we estimate the noise prevalence in the new (reduced) test dataset to be $N = \frac{|\mathcal{C}|}{|\mathcal{P}|-x|\mathcal{B}|}$. By varying

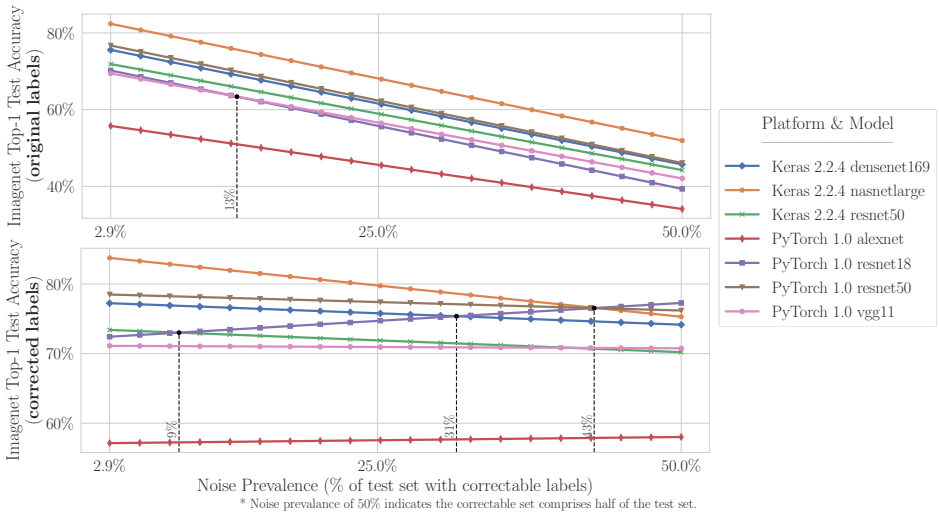


Figure 4: ImageNet top-1 original accuracy (top panel) and corrected accuracy (bottom panel) vs Noise Prevalence (agreement threshold = 3). Vertical lines indicate noise levels at which the ranking of two models changes (in terms of original/corrected accuracy). The left-most point ($N = 2.9\%$) on the x-axis is $|\mathcal{C}|/|\mathcal{P}|$, i.e. the (rounded) estimated noise prevalence of the pruned set, $\mathcal{P}$. The leftmost vertical dotted line in the bottom panel is read, "The Resnet-50 and Resnet-18 benchmarks cross at noise prevalence $N = 8.6\%$, implying Resnet-18 outperforms Resnet-50 when N increases by around 6% relative to the original pruned test data ($N = 2.9\%$ originally, c.f. Table 2).

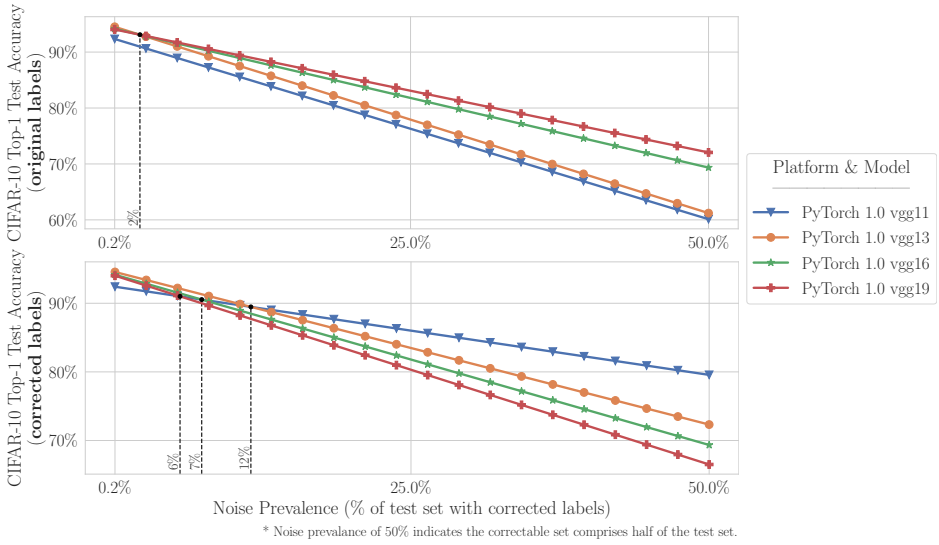


Figure 5: CIFAR-10 top-1 original accuracy (top panel) and corrected accuracy (bottom panel) vs Noise Prevalence (agreement threshold = 3). For additional details, see the caption of Fig. 4.

x from 0 to 1, we can simulate any noise prevalence ranging from $|\mathcal{C}|/|\mathcal{P}|$ to 1. We operationalize averaging over all choices of removal by linearly interpolating from benchmark accuracies on the corrected test set ($\mathcal{P}$, with corrected labels for the subset $\mathcal{C}$) to accuracies on the erroneously labeled subset ($\mathcal{C}$, with corrected labels).

For a given model, $\mathcal{M}$, its resulting accuracy (as a function of x) over the reduced test data is given by $A(x;\mathcal{M}) = \frac{A_{\mathcal{C}}(\mathcal{M})\cdot|\mathcal{C}|+(1-x)\cdot A_{\mathcal{B}}(\mathcal{M})\cdot|B|}{|\mathcal{C}|+(1-x)\cdot|B|}$, where $A_{\mathcal{C}}(\mathcal{M})$ and $A_{\mathcal{B}}(\mathcal{M})$ denote the (original or corrected) accuracy over the correctable set and benign set, respectively (accuracy before removing any examples). Here $A_{\mathcal{B}} = A^*_{\mathcal{B}} = \tilde{A}_{\mathcal{B}}$ because no erroneous labels were identified in $\mathcal{B}$. The expectation is taken over which fraction x of examples are randomly removed from $\mathcal{B}$ to produce the

reduced test set: the resulting expected accuracy, $A(x; \mathcal{M})$, is depicted on the y-axis of Figures 4-5. As our removal of test examples was random from the non-mislabeled set, we expect this reduced test data is representative of test sets that would be used in applications with a similarly greater prevalence of label errors. Note that we ignore non-correctable data with unknown labels ($\mathcal{U}$) throughout this analysis, as it is unclear how to report a better version of the accuracy for such ill-specified examples.

Over alternative (reduced) test sets created by imposing increasing degrees of noise prevalence in ImageNet/CIFAR-10, Figures 4-5 depict the resulting original (erroneous) test set accuracy and corrected accuracy of the models, expected on each alternative test set. For a given test set (i.e. point along the x-axis of these plots), the vertical ordering of the lines indicates how models would be favored based on original accuracy or corrected accuracy over this test set. Unsurprisingly, we see that more flexible/recent architectures tend to be favored on the basis of original accuracy, regardless of which test set (of varying noise prevalence) is considered. This aligns with conventional expectations that powerful models like NasNet will outperform simpler models like ResNet-18. However, if we shift our focus to the corrected accuracy (i.e. what actually matters in practice), it is no longer the case that more powerful models are reliably better than their simpler counterparts: the performance strongly depends on the degree of noise prevalence in the test data. For datasets where label errors are common, a practitioner is more likely to select a model (based on original accuracy) that is not actually the best model (in terms of corrected accuracy) to deploy.

Finally, we note that this analysis only presents a loose lower bound on the magnitude of these issues. We only identified a subset of the actual correctable set as we are limited to human-verifiable label corrections for a subset of data candidates (algorithmically prioritized via confident learning). Because the actual correctable sets are likely larger, our noise prevalence estimates are optimistic in favor of higher capacity models. Thus, the true gap between between corrected vs. original accuracy may be larger and of greater practical significance, even for the gold-standard benchmark datasets considered here. For many application-specific datasets collected by ML practitioners, the noise prevalence will be greater than the numbers presented here: thus, it is imperative to be cognizant of the distinction between corrected vs. original accuracy, and to utilize careful data curation practices, perhaps by allocating more of an annotation budget to ensure higher quality labels in the test data.

6 Conclusion

Traditionally, ML practitioners choose which model to deploy based on test accuracy — our findings advise caution here, proposing that judging models over correctly labeled test sets may be more useful, especially for noisy real-world datasets. Small increases in the prevalence of originally mislabeled test data can destabilize ML benchmarks, indicating that low-capacity models may actually outperform high-capacity models in noisy real-world applications, even if their measured performance on the original test data may be worse. This gap increases as the prevalence of originally mislabeled test data increases. It is imperative to be cognizant of the distinction between corrected vs. original test accuracy, and to follow dataset curation practices that maximize high-quality test labels, even if budget constraints limit you to lower-quality training labels.

This paper shares new findings about pervasive label errors in test sets and their effects on benchmark stability, but does not address whether the apparent overfitting of high-capacity models versus low-capacity models is due to overfitting to train set noise, overfitting to validation set noise during hyper-parameter tuning, or heightened sensitivity to train/test label distribution shift that occurs when test labels are corrected. An intuitive hypothesis is that high-capacity models more closely fit all statistical patterns present in the data, including those patterns related to systematic label errors that models with more limited capacity are less capable of closely approximating. A rigorous analysis to disambiguate and understand the contribution of each of these causes and their effects on benchmarking stability is a natural next step, which we leave for future work. How to best allocate a given human relabeling budget between training and test data also remains an open question.

Acknowledgements

This work was supported in part by funding from the MIT-IBM Watson AI Lab.

REFERENCES

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pp. 233–242. PMLR, 2017.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning (ICML)*, 2019.

F. R. Cordeiro and G. Carneiro. A survey on deep learning with noisy labels: How to train your model when you cannot trust on the annotations? In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 9–16, 2020.

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.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014. ISSN 21622388. doi: 10.1109/TNNLS.2013.2292894.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, New Orleans, LA, 2017.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations (ICLR)*, 2017.

G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. URL http://authors.library.caltech.edu/7694.

Patrick J Grother. Nist special database 19 handprinted forms and characters database. *National Institute of Standards and Technology*, 1995.

David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

Hrayr Harutyunyan, Kyle Reing, Greg Ver Steeg, and Aram Galstyan. Improving generalization by controlling label-noise information in neural network weights. In *International Conference on Machine Learning*, pp. 4071–4081. PMLR, 2020.

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*, pp. 770–778, 2016.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

Sara Hooker, Aaron Courville, Yann Dauphin, and Andrea Frome. Selective brain damage: Measuring the disparate impact of model pruning. *arXiv preprint arXiv:1911.05248*, 2019.

W Ronny Huang, Zeyad Emam, Micah Goldblum, Liam Fowl, Justin K Terry, Furong Huang, and Tom Goldstein. Understanding generalization through visualizations. *arXiv preprint arXiv:1906.03291*, 2019.

Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In Hal Daumé III and Aarti Singh (eds.), *ICML*, volume 119 of *PMLR*, pp. 4804–4815. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/jiang20c.html.

Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *International Conference on Data Mining (ICDM)*, 2016.

Jan Kremer, Fei Sha, and Christian Igel. Robust active label correction. In Amos Storkey and Fernando Perez-Cruz (eds.), *Proceedings of Machine Learning Research (PMLR)*, volume 84 of *Proceedings of Machine Learning Research*, pp. 308–316, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR. URL http://proceedings.mlr.press/v84/kremer18a.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html.

Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv:1708.02862*, 2017.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning (ICML)*, 2018.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Annual Conference of the Association for Computational Linguistics (ACL)*, pp. 142–150, Portland, Oregon, USA, June 2011. Annual Conference of the Association for Computational Linguistics (ACL). URL http://www.aclweb.org/anthology/P11-1015.

D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the Limits of Weakly Supervised Pretraining. *ArXiv*, May 2018.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Special Interest Group on Information Retrieval (SIGIR)*, pp. 43–52, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767755. URL http://doi.acm.org/10.1145/2766462.2767755.

Tom Mitchell. Twenty newsgroups dataset, 1999. URL https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2013.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research (JAIR)*, 2021.

Giorgio Patrini, Frank Nielsen, Richard Nock, and Marcello Carioni. Loss factorization, weakly supervised learning and label noise robustness. In *International Conference on Machine Learning (ICML)*, pp. 708–717, 2016.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, pp. 5389–5400, 2019.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv:1705.10694*, 2017.

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. "Everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *Human Factors in Computing Systems (CHI)*, 2021.

Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on ImageNet. In Hal Daumé III and Aarti Singh (eds.), *International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8634–8644. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/shankar20c.html.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. In *International Conference on Learning Representations (ICLR)*, 2015.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era, 2017. URL https://arxiv.org/pdf/1707.02968.pdf.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From imagenet to image classification: Contextualizing progress on benchmarks. In *International Conference on Machine Learning*, pp. 9625–9635. PMLR, 2020.

Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. On the margin theory of feedforward neural networks. *arXiv:1810.05369*, 2018.

Wikipedia contributors. List of datasets for machine learning research — wikipedia, the free encyclopedia, 2020. URL https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research. Online; accessed 22-October-2018.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Conference on Neural Information Processing Systems (NeurIPS)*, pp. 6225–6236. Curran Associates, Inc., 2019.

Jing Zhang, Victor S Sheng, Tao Li, and Xindong Wu. Improving crowdsourced label quality using noise correction. *IEEE transactions on neural networks and learning systems*, 29(5):1675–1688, 2017.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, 2018.

APPENDIX:PERVASIVE LABEL ERRORS IN TEST SETS DESTABILIZE MACHINE LEARNING BENCHMARKS

A DATASETS

We select 10 of the most-cited, open-source datasets created in the last 20 years from the Wikipedia List of ML Research Datasets (Wikipedia contributors, 2020), with preference for diversity across computer vision, NLP, sentiment analysis, and audio modalities. Citation counts were obtained via the Microsoft Cognitive API. In total, we evaluate six visual datasets: MNIST, CIFAR-10, CIFAR-100, Caltech-256, ImageNet, and QuickDraw; three text datasets: 20news, IMDB, and Amazon Reviews; and one audio dataset: AudioSet.

A.1 DATASET DETAILS

For each of the datasets we investigate, we summarize the original data collection and labeling procedure as they pertain to potential label errors.

MNIST (Lecun et al., 1998). MNIST is a database of binary images of handwritten digits. The dataset was constructed from Handwriting Sample Forms distributed to Census Bureau employees and high school students; the ground-truth labels were determined by matching digits to the instructions of the task in order to copy a particular set of digits (Grother, 1995). Label errors may arise from failure to follow instructions or from handwriting ambiguities.

CIFAR-10 / CIFAR-100 (Krizhevsky, 2009). The CIFAR-10 and CIFAR-100 datasets are collections of small 32×32 images and labels from a set of 10 or 100 classes, respectively. The images were collected by searching the internet for the class label. Human labelers were instructed to select images that matched their class label (query term) by filtering out mislabeled images. Images were intended to only have one prominent instance of the object, but could be partially occluded as long as it was identifiable to the labeler.

Caltech-256 (Griffin et al., 2007). Caltech-256 is a database of images and classes. Images were scraped from image search engines. Four human labelers were instructed to rate the images into "good," "bad," and "not applicable," eliminating the images that were confusing, occluded, cluttered, artistic, or not an example of the object category from the dataset.

ImageNet (Deng et al., 2009). ImageNet is a database of images and classes. Images were scraped by querying words from WordNet "synonym sets" (synsets) on several image search engines. The images were labeled by Amazon Mechanical Turk workers who were asked whether each image contains objects of a particular given synset. Workers were instructed to select images that contain objects of a given subset regardless of occlusions, number of objects, and clutter to "ensure diversity" in the dataset's images.

QuickDraw (Ha & Eck, 2017). The Quick, Draw! dataset contains more than 1 billion doodles collected from users of an experimental game to benchmark image classification models. Users were instructed to draw pictures corresponding to a given label, but the drawings may be "incomplete or may not match the label." Because no explicit test set is provided, we study label errors in the entire dataset to ensure coverage of any test set split used by practitioners.

20news (Mitchell, 1999). The 20 Newsgroups dataset is a collection of articles posted to Usenet newsgroups used to benchmark text classification and clustering models. The label for each example is the newsgroup it was originally posted in (e.g. "misc.forsale"), so it is obtained during the overall data collection procedure.

IMDB (Maas et al., 2011). The IMDB Large Movie Review Dataset is a collection of movie reviews to benchmark binary sentiment classification. The labels were determined by the user's review: a score ≤ 4 out of 10 is considered negative; ≥ 7 out of 10 is considered positive.

Amazon Reviews (McAuley et al., 2015). The Amazon Reviews dataset is a collection of textual reviews and 5-star ratings from Amazon customers used to benchmark sentiment analysis models. We use the 5-core (9.9 GB) variant of the dataset. **Modifications**: In our study, 2-star and 4-star reviews are removed due to ambiguity with 1-star and 5-star reviews, respectively. If these reviews were left in the dataset, they could inflate error counts. Because no explicit test set is provided, we study label errors in the entire dataset to ensure coverage of any test set split used by practitioners.

AudioSet (Gemmeke et al., 2017). AudioSet is a collection of 10-second sound clips drawn from YouTube videos and multiple labels describing the sounds that are present in the clip. Three human labelers independently rated the presence of one or more labels (as "present," "not present," and "unsure"), and majority agreement was required to assign a label. The authors note that spot checking revealed some label errors due to "confusing labels, human error, and difference in detection of faint/non-salient audio events."

B Details of Confident Learning (CL) for finding label errors

Here we summarize CL joint estimation and how it is used to algorithmically flag candidates with likely label errors for subsequent human review. An unnormalized representation of the joint distribution between observed and true label, called the *confident joint* and denoted $\boldsymbol{C}_{\tilde{y},y^*}$, is estimated by counting all the examples with noisy label $\tilde{y}=i$, with high probability of actually belonging to label $y^*=j$. This binning can be expressed as:

$$\boldsymbol{C}_{\tilde{y},y^*} = |\{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}:\ \hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta}) \geq t_j\}|$$

where $\boldsymbol{x}$ is a data example (e.g. an image), $\boldsymbol{X}_{\tilde{y}=i}$ is the set of examples with noisy label $\tilde{y}=i$, $\hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta})$ is the out-of-sample predicted probability that example $\boldsymbol{x}$ actually belongs to noisy class $\tilde{y}=j$ (even though its given label $\tilde{y}=i$) for a given model $\boldsymbol{\theta}$. Finally, t_j is a per-class threshold that, in comparison to other confusion matrix approaches, provides robustness to heterogeneity in class distributions and class distributions, defined as:

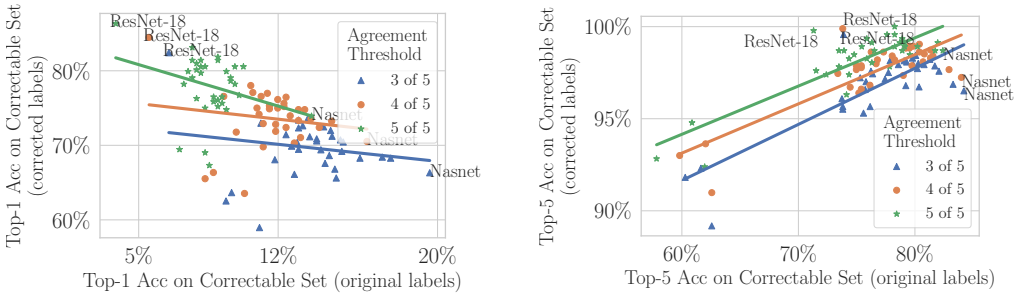
$$t_j = \frac{1}{|\boldsymbol{X}_{\tilde{y}=j}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta}) \tag{1}$$

A caveat occurs when an example is confidently counted into more than one bin. When this occurs, the example is only counted in the $\arg\max_{l \in [m]} \hat{p}(\tilde{y}=l;\boldsymbol{x},\boldsymbol{\theta})$ bin.

$\boldsymbol{Q}_{\tilde{y},y^*}$ is estimated by normalizing $\boldsymbol{C}_{\tilde{y},y^*}$, as follows:

$$\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} = \frac{\frac{\boldsymbol{C}_{\tilde{y}=i,y^*=j}}{\sum_{j \in [m]} \boldsymbol{C}_{\tilde{y}=i,y^*=j}} \cdot |\boldsymbol{X}_{\tilde{y}=i}|}{\sum_{i \in [m], j \in [m]} \left(\frac{\boldsymbol{C}_{\tilde{y}=i,y^*=j}}{\sum_{j \in [m]} \boldsymbol{C}_{\tilde{y}=i,y^*=j}} \cdot |\boldsymbol{X}_{\tilde{y}=i}| \right)} \tag{2}$$

The numerator calibrates $\sum_j \hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} = |\boldsymbol{X}_i| / \sum_{i \in [m]} |\boldsymbol{X}_i|, \forall i \in [m]$ so that row-sums match the observed prior over noisy labels. The denominator makes the distribution sum to 1.



(a) Top-1 Accuracy. (b) Top-5 Accuracy.

Figure S1: Benchmark ranking comparison of 34 pre-trained models on the ImageNet val set (used as test data here) for various settings of the agreement threshold. Top-5 benchmarks are unchanged by removing label errors (a), but change drastically on the correctable subset with original (erroneous) labels versus corrected labels. Corrected test set sizes: 1428 (▲), 960 (●), 468 (⋆).

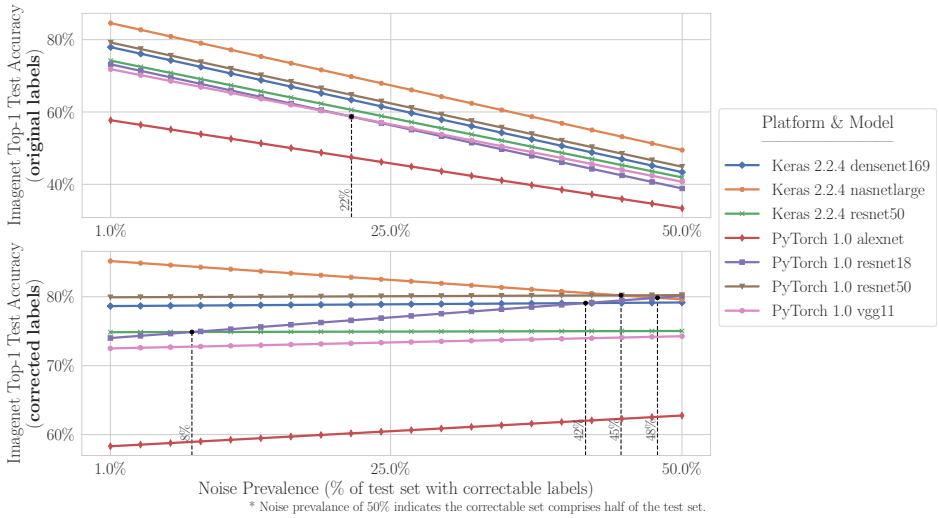


Figure S2: ImageNet top-1 original accuracy (top panel) and top-1 corrected accuracy (bottom panel) vs Noise Prevalence with agreement threshold = 5 (instead of threshold = 3, c.f., Fig. 4).

C CASE STUDY BENCHMARKING DETAILS

Figure 3 depicts how the benchmarking rankings on the correctable subset of ImageNet examples change significantly for an *agreement threshold* $= 5$, meaning 5 of 5 human raters need to independently select the same alternative label for that data point and a new label to be included in the accuracy evaluation. To ascertain that the results of this figure are not due to the setting of the agreement threshold, the results for all three settings of the agreement threshold are shown in Sub-figure S1b. Observe the negative correlation (for top-1 accuracy) occurs in all three settings. Furthermore, observe that this negative correlation no longer holds when top-5 accuracy is used (shown in S1a), likely because many of these models use a loss which maximizes (and overfits to noise) based on top-1 accuracy, not top-5 accuracy. Regardless of whether top-1 or top-5 accuracy is used, model benchmark rankings change significantly on the correctable set in comparison to the original test set (see Table S1).

The dramatic changes in ranking shown in Table S1 may be explained by overfitting to the validation when these models are trained, which can occur inadvertently during hyper-parameter tuning, or by overfitting to the noise in the training set. The benchmarking experiment was replicated on CIFAR-10 in addition to ImageNet. The individual accuracies for CIFAR-10 are reported in Table S2.

Table S1: Individual accuracy scores for Sub-figure 3b with *agreement threshold = 3 of 5*. Acc@1 stands for the (top-1 validation) original accuracy on the correctable set, in terms of original ImageNet examples and labels. *cAcc@1* stands for the (top-1 validation) corrected accuracy on the correctable set of ImageNet examples with correct labels. To be corrected, at least 3 of 5 Mechanical Turk raters had to independently agree on a new label, proposed by us using the class with the $\arg\max$ probability for the example.

Platform	Model	Acc@1	cAcc@1	Acc@5	cAcc@5	Rank@1	cRank@1	Rank@5	cRank@5
PyTorch 1.0	resnet18	6.51	82.42	73.81	99.58	34	1	30	1
PyTorch 1.0	resnet50	13.52	73.74	79.97	98.46	20	2	11	2
PyTorch 1.0	vgg19_bn	13.03	73.39	79.97	97.97	23	3	10	9
PyTorch 1.0	vgg11_bn	11.13	72.97	76.26	97.55	30	4	22	15
PyTorch 1.0	resnet34	13.24	72.62	77.80	98.11	21	5	18	6
PyTorch 1.0	densenet169	14.15	72.55	79.62	98.32	16	6	12	3
PyTorch 1.0	densenet121	14.29	72.48	78.64	97.97	14	7	16	11
PyTorch 1.0	vgg19	13.03	72.34	79.34	98.04	22	8	13	8
PyTorch 1.0	resnet101	14.64	71.99	81.16	98.25	11	9	5	4
PyTorch 1.0	vgg16	12.39	71.43	77.52	97.20	28	10	19	19
PyTorch 1.0	densenet201	14.71	71.22	80.81	97.97	10	11	6	10
PyTorch 1.0	vgg16_bn	13.59	71.15	77.87	97.41	19	12	17	17
Keras 2.2.4	densenet169	13.94	70.87	78.85	98.18	17	13	15	5
PyTorch 1.0	densenet161	15.13	70.73	80.11	98.04	7	14	8	7
Keras 2.2.4	densenet121	13.94	70.59	76.40	97.48	18	15	20	16
PyTorch 1.0	resnet152	15.27	70.45	81.79	97.83	5	16	4	12
PyTorch 1.0	vgg11	12.96	70.38	75.49	97.27	25	17	27	18
PyTorch 1.0	vgg13_bn	12.68	69.89	75.84	96.99	27	18	25	20
PyTorch 1.0	vgg13	13.03	69.47	76.40	96.78	24	19	21	24
Keras 2.2.4	nasnetmobile	14.15	69.40	79.27	96.85	15	20	14	21
Keras 2.2.4	densenet201	15.20	69.19	80.11	97.76	6	21	9	13
Keras 2.2.4	mobilenetV2	14.57	68.63	75.84	96.57	12	22	24	26
Keras 2.2.4	inceptionresnetv2	17.23	68.42	83.40	96.85	3	23	2	22
Keras 2.2.4	xception	17.65	68.28	82.07	97.62	2	24	3	14
Keras 2.2.4	inceptionv3	16.11	68.28	80.25	96.78	4	25	7	23
Keras 2.2.4	vgg19	11.83	68.07	73.95	95.52	29	26	29	30
Keras 2.2.4	mobilenet	14.36	67.58	73.60	96.08	13	27	31	27
Keras 2.2.4	resnet50	14.85	66.81	76.12	95.73	9	28	23	28
Keras 2.2.4	nasnetlarge	19.61	66.32	84.24	96.57	1	29	1	25
Keras 2.2.4	vgg16	12.82	66.11	74.09	95.66	26	30	28	29
PyTorch 1.0	inception_v3	14.92	65.62	75.56	95.38	8	31	26	31
PyTorch 1.0	squeezenet1_0	9.66	63.66	60.50	91.88	32	32	34	33
PyTorch 1.0	squeezenet1_1	9.38	62.54	61.97	92.30	33	33	33	32
PyTorch 1.0	alexnet	11.06	58.96	62.61	89.29	31	34	32	34

Table S2: Individual CIFAR-10 accuracy scores for Sub-figure 3c with *agreement threshold = 3 of 5*. Acc@1 stands for the top-1 validation accuracy on the correctable set ($n = 18$) of original CIFAR-10 examples and labels. See Table S1 caption for more details. Discretization of accuracies occurs due to the limited number of corrected examples on the CIFAR-10 test set.

Platform	Model	Acc@1	cAcc@1	Acc@5	cAcc@5	Rank@1	cRank@1	Rank@5	cRank@5
PyTorch 1.0	googlenet	55.56	38.89	94.44	94.44	1	10	13	13
PyTorch 1.0	vgg19_bn	50.00	38.89	100.00	100.00	2	11	7	7
PyTorch 1.0	densenet169	44.44	50.00	100.00	100.00	5	4	2	2
PyTorch 1.0	vgg16_bn	44.44	44.44	100.00	100.00	3	8	5	5
PyTorch 1.0	inception_v3	44.44	33.33	100.00	100.00	6	12	8	8
PyTorch 1.0	resnet18	44.44	55.56	94.44	100.00	4	2	10	10
PyTorch 1.0	densenet121	38.89	50.00	100.00	100.00	8	5	3	3
PyTorch 1.0	densenet161	38.89	50.00	100.00	100.00	9	6	4	4
PyTorch 1.0	resnet50	38.89	44.44	100.00	100.00	7	9	6	6
PyTorch 1.0	mobilenet_v2	38.89	27.78	100.00	100.00	10	13	9	9
PyTorch 1.0	vgg11_bn	27.78	66.67	100.00	100.00	11	1	1	1
PyTorch 1.0	resnet34	27.78	55.56	94.44	100.00	13	3	11	11
PyTorch 1.0	vgg13_bn	27.78	50.00	94.44	100.00	12	7	12	12