Pervasive Label Errors in Test Sets
Destabilize Machine Learning Benchmarks

Curtis G. Northcutt*  Anish Athalye  Jonas Mueller
ChipBrain, MIT  MIT  Amazon

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 3.3% errors across the 10 datasets, 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 crowd-sourcing, techniques which are inherently error-prone [37]. Even with controls for error correction [20, 46], 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 [30, 33], and also because deep learning is believed to be naturally robust to label noise [17, 27, 36, 41].

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 [16, 31, 38], 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

*Correspondence to: curtis@chipbrain.com or cgn@mit.edu.

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

2All label errors can be reproduced at https://github.com/cgnorthcutt/label-errors.

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. 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 [4], 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 [31], we algorithmically identify putative label errors in test sets at scale \(^3\), and we validate these label errors through human evaluation, estimating an average of 3.3% 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.

---

\(^3\)To find all label errors, we use the cleanlab implementation of confident learning open-sourced at: https://github.com/cgnorthcutt/cleanlab

---

**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.
with the prevalence of mislabeled test data. This is not traditional overfitting. Larger models are 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. We discover label errors are pervasive in test sets of popular benchmarks used in nearly all machine learning research.
2. We open-source the resources\textsuperscript{4} to clean and correct each test set, 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 on benchmarks, finding 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 corrected test data we provide).
4. 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 benchmarks.

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 \cite{19, 30, 32, 40, 43} 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 \cite{18}), or a naturally noisy dataset is used and accuracy is measured on a noisy test set. In the noisy WebVision dataset \cite{23}, accuracy on the ImageNet validation data is often reported as a “clean” test set, but several studies \cite{16, 31, 35, 42} have 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 prior work deals with label errors in the training set \cite{4, 7}, much less work has been done to understand the implications of label errors in the test set.

\textsuperscript{4}A corrected version of each test set is provided at https://github.com/cgnorthcutt/label-errors.
theoretical support for realistic settings with asymmetric, non-uniform noise. For robustness to class imbalance and theoretical support for exact uncertainty quantification, we use a model-agnostic framework, confident learning (CL) [31], to estimate which labels are erroneous across diverse datasets. [31] have demonstrated that CL more accurately identifies label errors than other label-error identification methods. Unlike prior work that only models symmetric label noise [43], 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).

3 Identifying label errors in benchmark datasets

Here we summarize our algorithmic label error identification performed prior to crowd-sourced human verification. An overview of each dataset and any modifications is detailed in Appendix A. Step-by-step instructions to obtain each dataset and reproduce the label errors for each dataset are provided at https://github.com/cgnorthcutt/label-errors. The primary contribution of this section is not in the methodology, which is covered extensively in [31], 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 [31] to estimate $Q_{\tilde{y}, y}^\ast$, the $m \times m$ discrete joint distribution of observed, noisy labels, $\tilde{y}$, and unknown, true labels, $y^\ast$. Inherent in $Q_{\tilde{y}, y}^\ast$ is the assumption that noise is class-conditional [1], depending only on the latent true class, not the data. This assumption is commonly used [9, 40] because it is reasonable. For example, in ImageNet, a tiger is more likely to be mislabeled cheetah than flute.

The diagonal entry $\hat{P}(\tilde{y}=i, y^\ast=i)$ of matrix $Q_{\tilde{y}, y}^\ast$ is the probability that examples in class i are correctly labeled. If the dataset is error-free, then $\sum_{i \in [m]} \hat{P}(\tilde{y}=i, y^\ast=i) = 1$. The fraction of label errors is $\rho = 1 - \sum_{i \in [m]} \hat{P}(\tilde{y}=i, y^\ast=i)$ 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: $\hat{P}(\tilde{y}=i; x) - \max_{j \neq i} \hat{P}(\tilde{y}=j; x)$ [44]. Table 1 shows the number of CL guessed label issues for each test set across ten popular ML benchmark datasets. CL estimation of $Q_{\tilde{y}, y}^\ast$ is summarized in the appendices in (Sec. C).

Computing out-of-sample predicted probabilities Estimating $Q_{\tilde{y}, y}^\ast$ for CL noise characterization requires two inputs for each dataset: (1) out-of-sample predicted probabilities $\hat{P}_{k,i}$ ($n \times m$ matrix) and (2) the test set labels $\tilde{y}_k$. We observe the best results computing $\hat{P}_{k,i}$ by pre-training on the train set, then fine-tuning (all layers) on the test set using cross-validation to ensure $\hat{P}_{k,i}$ 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 $\hat{P}_{k,i}$ 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.

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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Modality</th>
<th>Size</th>
<th>Model</th>
<th>CL guessed</th>
<th>MTurk checked</th>
<th>Test Set Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>image</td>
<td>10,000</td>
<td>2-conv CNN</td>
<td>100</td>
<td>100 (100%)</td>
<td>15</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>image</td>
<td>10,000</td>
<td>VGG</td>
<td>275</td>
<td>275 (100%)</td>
<td>54</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>image</td>
<td>10,000</td>
<td>VGG</td>
<td>2,335</td>
<td>2,335 (100%)</td>
<td>585</td>
</tr>
<tr>
<td>Caltech-256</td>
<td>image</td>
<td>29,780</td>
<td>Wide ResNet-50-2</td>
<td>2,360</td>
<td>2,360 (100%)</td>
<td>458</td>
</tr>
<tr>
<td>ImageNet</td>
<td>image</td>
<td>50,000</td>
<td>ResNet-50</td>
<td>5,440</td>
<td>5,440 (100%)</td>
<td>2,916</td>
</tr>
<tr>
<td>QuickDraw</td>
<td>image</td>
<td>50,426,266</td>
<td>VGG</td>
<td>6,825,383</td>
<td>6,825,383 (100%)</td>
<td>1870</td>
</tr>
<tr>
<td>20news</td>
<td>text</td>
<td>7,532</td>
<td>THDF + SGD</td>
<td>93</td>
<td>93 (100%)</td>
<td>82</td>
</tr>
<tr>
<td>IMDB</td>
<td>text</td>
<td>25,000</td>
<td>FastText</td>
<td>1,131</td>
<td>1,131 (100%)</td>
<td>725</td>
</tr>
<tr>
<td>Amazon</td>
<td>text</td>
<td>9,996,437</td>
<td>FastText</td>
<td>533,249</td>
<td>533,249 (100%)</td>
<td>732</td>
</tr>
<tr>
<td>AudioSet</td>
<td>audio</td>
<td>20,371</td>
<td>VGG</td>
<td>307</td>
<td>307 (100%)</td>
<td>275</td>
</tr>
</tbody>
</table>

* 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Test Set Errors Categorization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>non-errors</td>
</tr>
<tr>
<td>MNIST</td>
<td>85</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>221</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>1650</td>
</tr>
<tr>
<td>Caltech-256</td>
<td>1902</td>
</tr>
<tr>
<td>ImageNet</td>
<td>2524</td>
</tr>
<tr>
<td>QuickDraw</td>
<td>630</td>
</tr>
<tr>
<td>20news</td>
<td>11</td>
</tr>
<tr>
<td>IMDB</td>
<td>585</td>
</tr>
<tr>
<td>Amazon Reviews</td>
<td>268</td>
</tr>
<tr>
<td>AudioSet</td>
<td>32</td>
</tr>
</tbody>
</table>

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

4 Validating label errors

We validated the algorithmically identified label errors with a Mechanical Turk study. For two datasets with a large number of errors (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 S1 in the appendices 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.

4.1 Failure Modes of confident learning

![Figure 2: Difficult examples from various datasets in https://labelerrors.com where confident learning potentially finds a label error incorrectly prior to human validation. Example (a) is a cropped image of part of an antiquated sewing machine; (b) is a viewpoint from inside an airplane, looking out at the runway and grass with a partial view of the nose of the plane; (c) is an ambiguous shape which could be a potato; (d) is a digit which is impossible to distinguish; (e) is a male whose exact age cannot be determined; and (f) is a straw used as a pole within a miniature replica of a village.](https://example.com/image)

By visually inspecting putative label errors, we identified certain previously unexamined failure modes of confident learning [31]. Appendix D provides a mathematical description of the conditions under...
which these failure modes occur. Figure 2 shows uniquely challenging examples, with excessively
erroneous \( \hat{y}(\tilde{y} = j; x) \), where failure mode cases potentially occur. The sewing machine in Subfigure
2(a), for example, exhibits a “part versus whole” issue where the image has been cropped to reveal
only a small portion of the object. The airplane in Subfigure 2(b) is from the perspective of the pilot,
looking out of the front cockpit window.

Figure 2 clarifies that our corrected test set labels are not 100% perfect. Even with a stringent 5
of 5 agreement threshold where all human reviewers agreed on a label correction, the “corrected”
label is not always actually correct. Fortunately, these failure mode cases are rare. Inspection of
https://labelerrors.com shows that the majority of the labels we corrected are reasonable. Our
corrected test sets, while imperfect in these rare cases, are vastly improved from the original test sets.

5 Implications of label errors in test data

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

Definition 1 (original accuracy, \( A^\circ \)). 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 \( A^\circ \) for evaluating models (because \( A^* \) better reflects performance
in real-world applications).

In the following definitions, “\( \backslash \)” denotes a set difference, \( D \) denotes the full test dataset, and \( B \subset D \)
denotes the subset of benign test examples that CL did not flag as likely label errors.

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

Definition 4 (pruned set, \( P \)). The remaining test data after removing \( U \) from \( D \) (\( P = D \backslash U \)).

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

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 \( N = \frac{|C|}{|P|} \).

These definitions imply \( B, C, U \) are disjoint with \( D = B \cup C \cup U \), and also \( P = B \cup C \). In subsequent
experiments, we report corrected test accuracy over \( P \) after correcting all of the labels in \( C \subset P \).
We ignore the unknown-label set \( U \) (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
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. [35]; 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” \( C \). When evaluating models only on the
subset of test examples in \( C \), models which perform best on the original (incorrect) labels perform the
worst on corrected labels. For example, ResNet-18 [14] significantly outperforms NasNet [47] in
terms of corrected accuracy over \( C \), despite exhibiting far worse original test accuracy. The change
in ranking can be dramatic: Nasnet-large drops from ranking 1/34 → 29/34, Xception drops from ranking 2/34 → 24/34, ResNet-18 increases from ranking 34/34 → 1/34, and ResNet-50 increases from ranking 20/24 → 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 [39] 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.

Arpit et al. [2], Harutyunyan et al. [13] 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 \( N = \frac{|\mathcal{C}|}{|\mathcal{P}|} \cdot \frac{|\mathcal{B}|}{|\mathcal{B}|} \). By varying \( x \) 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 \( A(x; \mathcal{M}) = \frac{A_C(\mathcal{M}) \cdot |\mathcal{C}| + (1-x) \cdot A_B(\mathcal{M}) \cdot |\mathcal{B}|}{C + (1-x) \cdot B} \), where \( A_C(\mathcal{M}) \) and \( A_B(\mathcal{M}) \) denote the (original or corrected) accuracy over the correctable set and benign set, respectively (accuracy before removing any examples). Here \( A_B = A_B = \tilde{A}_B \) 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, \( \bar{A}(x; \mathcal{M}) \), is depicted on the y-axis of Figures 4-5.
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 $|C|/|P|$, i.e. the (rounded) estimated noise prevalence of the pruned set, $P$. 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.

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 ($U$) 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 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 Discussion

This paper demonstrates that label errors are ubiquitous in the test sets of many popular benchmarks
used to gauge progress in machine learning. We hypothesize that this has not been previously
discovered and publicized at such scale due to various challenges. Firstly, human verification of all
labels can be quite costly, which we circumvented here by using CL algorithms to filter for likely label
errors prior to human verification in an automated manner. Secondly, working with 10 differently
formatted datasets was nontrivial, with some exhibiting peculiar issues upon close inspection (despite
being standard benchmarks). For example, IMDB, QuickDraw, and Caltech-256 lack a global index
making it difficult to map model outputs to corrected test examples on different systems. We provide
index files in our open-source repository to address this. Furthermore, Caltech-256 contains several
duplicate images, which we found no previous mention of. Lastly, ImageNet contains duplicate class
labels, e.g. maillot and crane [31, 42].

Traditionally, ML practitioners choose which model to deploy based on test accuracy — our findings
advise caution here. Instead, judging models over correctly labeled test sets may be important, espe-
cially for real-world datasets that are likely noisier than these academic benchmarks. 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 label verification 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


Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We discuss failure modes of confident learning (Appendix D), show examples that human annotators still failed to correct (Fig. 2), and state additional limitations in the Discussion (Section 6).
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discuss the potential flaws of choosing models based on original test set accuracy with noisy labels and the implications for real-world deployment of ML models (see Section 5).
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report all of our results under both 5 of 5 human annotator agreement threshold and 3 of 5 threshold to define corrected labels. This provides some idea of the uncertainty in the label correction process. For reporting accuracy of models, we did not compute error bars because we are working with established benchmarks, and thus ran our experiments using the provided train/test split in the standard fashion.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section D.1.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See https://github.com/cgnorthcutt/label-errors#license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See https://github.com/cgnorthcutt/label-errors and https://github.com/cgnorthcutt/label-errors/releases.
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [Yes] We are not providing any new data in this work, merely providing new corrected labels for previously published data that is widely used in ML research.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We saw none of these issues in our inspection of the data, although such issues have previously plagued even the most popular benchmarks (presumably because they were never closely audited in the first place).

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] See Section B.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] See Section B.1.